5700, 5800, 5900, 7000 Series Routers

Command Line Interface Guide

June 2001

Copyright

Trademarks

What's New in This Release?

This version of the Command Line Interface (CLI) manual has been updated to document features available with this release of the kernel software. The following list directs you to the new CLI documentation:

Release 5.3:

Built-in firewall filters — page 129

- Options available for maximum, medium, minimum, and no security.

Support for Tollbridge voice gateways

- Additional voice troubleshooting commands available — page 191

New voice router selections.

- alaw or µlaw encoding — page 24
- CAS refresh signaling mode — page 24

Support for G.shdsl routers

- Commands to manage G.shdsl link — page 346

Support for Copper Mountain Plug & Play configuration

- Available when using the Copper Mountain CopperEdge™ 200 DSLAM, version 3.0, and router models 5871 IDSL, 5851 SDSL, and 7851 SDSL IAD — page 48

New SNTP commands — page 221

- Commands to manage the SNTP server list and issue an SNTP time request.

New date and time commands — page 212 and page 224

- Commands to display and change the current date and time.

New command to turn on the PPP retry timer — page 318

New IKE command — page 376

- Determines setting of IKE commit bit.

New debugging commands

- For ATM debugging (page 204); for IKE debugging (page 208); and for the ADSL DMT modem timeout (page 206).

Release 5.1.0:

New IP Filter types — page 129

- The IP filter type **Forward** has been replaced by the filter types **Receive** and **Transmit**.

H.323 support with Network Address Translation — page 100

- Configuration needed to receive incoming calls from NetMeeting or other H.323 applications.

Domain names allowed on ping and traceroute commands — page 217 and page 224

- Command requests IP address of domain from domain name server.

Rapid Secure Encryption hardware— page 125

- Hardware option available to accelerate encryption.

Over Subscription support for VoDSL routers— page 22

- Upstream over subscription is supported for CopperCom and Jetstream gateways.

Release 5.0:

VRRP Backup — page 116

- Implements the Virtual Router Redundancy Protocol (RFC 2338).
- Allows other routers in the LAN to serve as backups for a static default gateway.

Dial Backup — see page 109

- Uses a V.90 modem connected to the console port as an automatic backup when the DSL link fails.

SDSL Autobaud Pre-Activation Procedure — see page 343

- Shortens the time required to find the correct line speed by checking the line quality before activation.

New IKE Commands

- Perfect Forward Secrecy option to increase the security of the IKE key exchange — see page 151.
- Option to restrict policy use to a specific interface — see page 379.
- Option to have network address translation performed before IPSec encryption— see page 382.

Interface Stop, Start and Restart

- Commands to stop, start, and restart a logical Ethernet interface — see page 79.
- Commands to stop, start, and restart an active session for a remote — see page 35.

Voice router support

- Command to change voice profile available if voice gateway is ATM standards-based — see page 23.
- Command to adjust jitter buffer size — see page 193.

Multiple BootP relays — see page 167

- Use **dhcp addrelay** and **dhcp delrelay** commands to change the BootP server list.

Changes to Syslog server list — see page 168

- The command syntax has changed for **system addSyslogServer** and **system delSyslogServer**.

New “all protocols” option on AddServer commands

- **system addServer** — see page 233; **remote addServer** — see page 293; **eth ip addServer** — see page 265.

New DHCP command — see page 354

- Command to clear all DHCP information.

IP Filter changes — see **eth ip filter,** page 270 or **remote ipfilter**, page 300

- New **-tcp rst** parameter allows a filter to match the TCP RESET flag.
- Watch messages are also sent to Unix Syslog servers.

New IP remote LANCONFIG option — see page 311

- PPP remote can receive IPCP information for dyamically reconfiguring the Ethernet interface.

Upgradable Bridge Support — see page 20

About This Manual

This manual contains information on the syntax and use of the Command Line Interface for this family of DSL routers. Configuration of network connections, bridging, routing, and security features are essentially the same for all DSL routers, unless otherwise noted.

This manual is intended for small and home office users, remote office users, and other networking professionals who are installing and maintaining bridged and routed networks.

It assumes that you have read the *User Reference Guide* that came with the router and have installed the router as described in that guide.

As described in the *User Reference Guide*, a graphical interface is also available for configuring the router. It provides many, but not all, of the capabilities of the Command Line Interface. Look for the *User Reference Guide* in the box in which your router was shipped or find it on the Technical Support web site (www.efficient.com).

How This Manual is Organized

This manual is organized into these parts:

How to Access the Command Line. Describes how to access the router command line from a PC so you can enter router commands.

Router Concepts. Contains information on topics such as routing and bridging operations, voice routing, PAP/CHAP security negotiation, bandwidth management, interoperability, protocol conformance, and the file system.

Planning for Router Configuration. Discusses the information required for basic configuration of the router.

Configuring the Router. Outlines the commands required for basic configuration of the router.

Configuring Special Features. Describes how to configure advanced features, such as Bridge Filtering, RIP, DHCP, NAT, Dial Backup, and VRRP.

Configuring Software Options. Describes how to install and configure features available via software option keys, including Encryption, IP Filtering, L2TP Tunneling, and IKE/IPSec.

Managing the Router. Describes router management capabilities, including SNMP, Telnet, TFTP client and server, BootP, Syslog, boot code options, software upgrades, backup and recovery procedures, and batch file command execution.

Troubleshooting. Describes diagnostic tools used for identifying and correcting hardware and software problems.

Command Reference. Provides a description and syntax for each command.

Appendix A provides blank Network Information Sheets.

Appendix B describes IPX configuration.

Two indexes are provided at the end of the manual. The **Command Index** directs you to the desired command description. The **Topic Index** directs you to specific feature discussions.

Typographic Conventions

The following typeface conventions are used in this guide:

Typeface	Item	Examples
Italics	Book titles, command reference parameters, cross-references, text emphasis.	Refer to the *User Reference Guide*. system name <*name*>
Bold	Keywords in command reference instructions	**save**
Mono-spaced font	Examples.	`remote listIpRoute hq`
Uppercase	File names	Copy file CFGMGR.EXE

Table of Contents

How to Access the Command Line

This manual describes the Command Line Interface for your router. The Command Line Interface gives you access to all capabilities of your router.

A GUI (graphic user interface) is also available for configuring the router. It provides many, but not all, of the capabilities of the Command Line Interface. To learn how to access the GUI, refer to the *User Reference Guide* that came in the box in which your router was shipped or find the guide on the Technical Support web site (www.efficient.com).

To use the Command Line Interface, you must first access the router command line. To do this, you:

1. Connect a PC (or ASCII) terminal to a port of the router.
   (The required cable and adapter are provided with the router. The connection procedure is described in detail in the *User Reference Guide* that came with the router.)

2. Restart the PC and power on the router.

3. Open a terminal window or start a terminal session on the PC.

4. The router displays the **Login:** prompt.

   ```
   Login:
   ```

5. Enter the login password. (The default is **admin**. To change the login password, use the **system admin** command, page 236). You may then begin entering router commands.

The router supports both local access and remote access. In step 3 above, the terminal session could be:

- The terminal window from within the Quick Start or Configuration Manager application (for local access)
- A terminal session (for local access)
- A Telnet session (for remote access)

Terminal Window

To access the terminal window from within the Quick Start or Configuration Manager application:

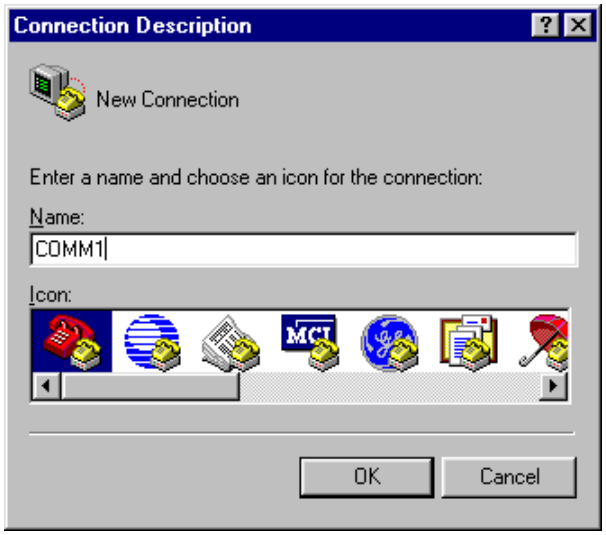
- Select **Tools** and **Terminal Window** from the main menu.

The menu selection **Commands** provides shortcuts to most of the commands described in this manual.

Terminal Session under Windows (HyperTerminal)

To open the HyperTerminal emulator available under the Windows operating system:

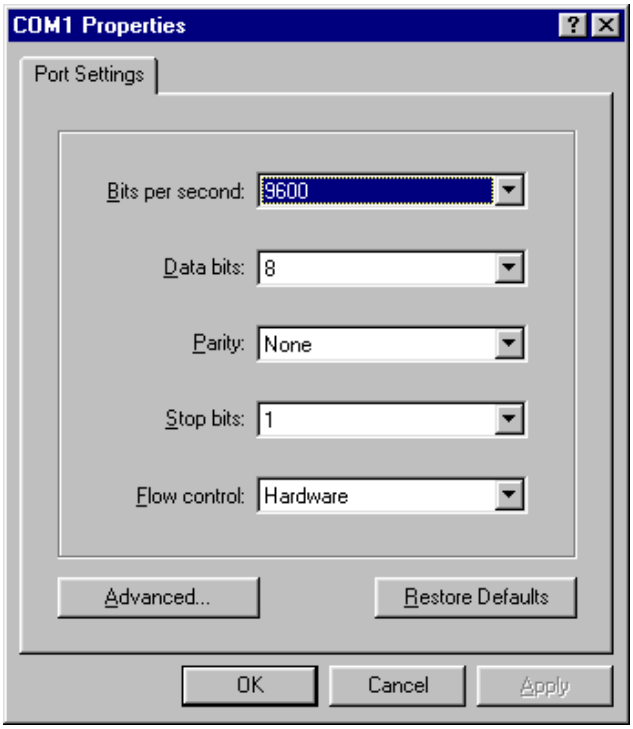
1. Click **Start** on your desktop and then select **Programs > Accessories > Communications > Hyperterminal**.
2. Double-click **Hypertrm.exe.**
3. In the **Connection Description** window, enter a name for the connection and select **OK**.



3. In the **Phone Number** window, under **Connect using**, select **Choose Direct to Com 1** (or **2**).
4. In the **Com 1** (or **2**) **Properties page**, enter the following port settings and select **OK**:

Bits per second:	9600
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	Hardware

(To use a baud rate other than 9600, see page 172.)



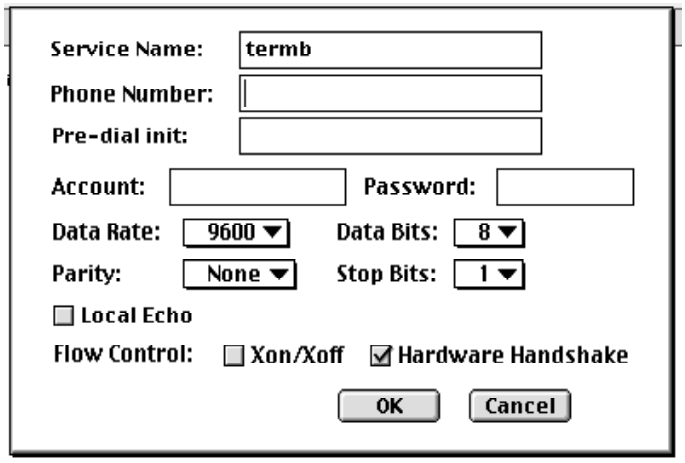
Terminal Session for Macintosh or UNIX

To open a terminal window emulation in a Macintosh or UNIX environment, you need a VT100 terminal emulation program.

1. Start your VT100 terminal emulator.
2. Configure the emulator with the following port settings:

Baud rate (Data rate):	9600
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	Hardware

(To use a baud rate other than 9600, see page 172.)

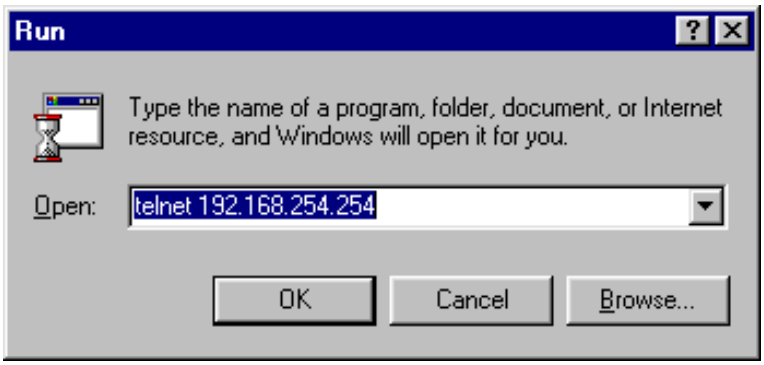


Telnet Session for Remote Access

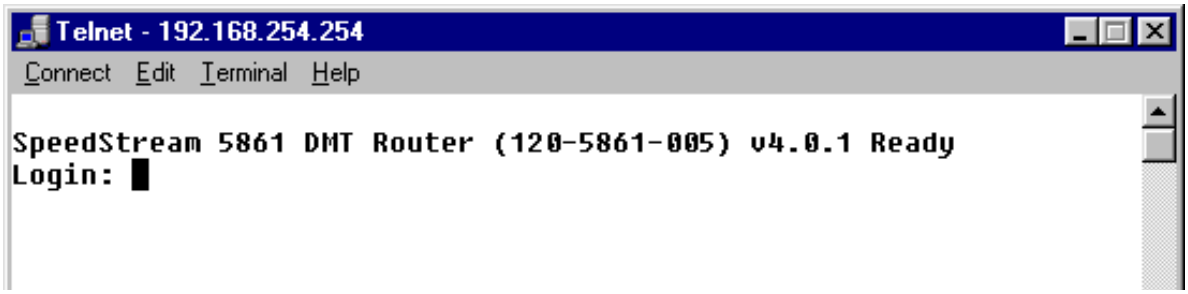
The router supports Telnet access. (For more information, see *Telnet Remote Access*, page 166.) However, remote access to the router configuration can be disabled or restricted (see *Controlling Remote Management*, page 107).

To set up a Telnet session under Windows for remote access to the router:

1. Make sure that your PC and router addresses are in the same subnetwork. For example, the router address could be 192.168.254.254 and the PC address could be 192.168.254.253.
2. Click the **Start** button on your PC desktop and select **Run**.
3. In the window, enter **telnet 192.168.254.254** and select **OK**.



4. The router displays a line identifying itself and then displays the **Login:** prompt.



Chapter 1. Router Concepts

This chapter provides background information applicable to the router on topics useful to network administrators. These topics include:

- Routing and bridging
  - Routing and bridging controls
  - Bridge filtering
- xDSL WAN interfaces
- Voice routing

- PAP/CHAP security authentication
- Security passwords and levels

- Interoperability between the router and other systems
- Protocol conformance
- Encapsulation options

- System and configuration files

Routing and Bridging

The router can operate as a bridge, as a router, or as both. The following sections describe routing and bridging and how the two functions operate together.

Routing

Routing is the process that determines where data is sent. A router can route user data from source to destination over different LAN and WAN links. Routing relies on routing address tables to determine the best path for each packet to take.

The routes within a routing address table are established in two ways:

- You can enter specific static routes. For each route, you enter the address for a remote destination with path details and a value for the perceived cost of that route (path latency).
- The routing tables can also be built dynamically; i.e., the location of remote stations, hosts, and networks are updated from broadcast packet information.

Routing offers advantages over bridging because:

- It limits broadcasts to the local LAN segment.
- It limits the protocols that are routed beyond the LAN segment.
- Routed protocols allow networks to grow as large as needed.
- Filters and firewalls can provide screens for improved security and managed traffic flow.

Numerous network protocols have evolved, and within certain protocol suites are associated protocols for routing, error handling, network management, etc. The following chart lists networking protocols and associated protocols supported by the router.

Network Protocol	Associated Protocols	Description
IP (Internet Protocol)	**RIP** (Routing Information Protocol)	Maintains a map of the network
	ARP (Address-Resolution Protocol)	Maps IP addresses to data-link addresses
	RARP (Reverse Address Resolution Protocol)[a]	Maps data-link addresses to IP addresses
	ICMP (Internetwork Control Message Protocol)	Diagnostic and error reporting/ recovery
	SNMP (Simple Network Management Protocol)	Network management
IPX (Internet Packet Exchange)	**RIP** (Routing Information Protocol)[b]	Maintains a map of the network
	SAP (Service Advertising Protocol)	Distributes information about service names and addresses

a Used only during a network boot.
b IPX-RIP is a different protocol from IP-RIP and it includes time delays.

Bridging

Bridging connects two or more LANs so that all devices share the same logical LAN segment and network numbers. Transparent bridging allows locally connected devices to send frames to all devices as if they were local.

The MAC layer header contains source and destination addresses used to transfer frames. An address table is dynamically built and updated with the logical port a device is connected to as frames are received. (To see the contents of the bridging table, use the command **bi list**, page 212.)

Bridging has these capabilities:

- Allows protocols that cannot be routed (such as NETBIOS) to be forwarded.
- Allows optimizing internetwork capacity by localizing traffic on LAN segments.
- Extends the physical reach of networks beyond the limits of each LAN segment.
- Bridge filtering may increase network security.

Our bridging support includes the IEEE 802.1D standard for LAN-to-LAN bridging and the Spanning Tree Protocol for interoperability with other vendors' bridge/routers. Bridging is provided over PPP as well as adjacent LAN ports.

Bridge Filtering

You can control the flow of packets through the router using bridge filters. The filters can “deny” or “allow” packets to cross the network based on the content of the packets. This feature lets you restrict or forward messages with a specified address, protocol, or data content. Common uses are to prevent access to remote networks, control unauthorized access to the local network, and limit unnecessary traffic.

For example, to restrict remote access for specific users, you could define bridge filters using the local MAC address of each user to be restricted. Each bridge filter is specified as a “deny” filter based on the MAC address and position of the address within the packet. Deny filtering mode is then enabled to initiate bridge filtering. While in deny mode, all packets containing one of the filtered MAC addresses are denied bridging across the router.

Similarly, protocol filtering can be used to prevent a specific protocol from being bridged. In this case, the protocol ID field in a packet is used to deny or allow a packet. You can also restrict the bridging of specific broadcast packets.

For a further discussion of bridge filtering, see page 81.

Bridge-Only Units

A series of bridge-only units is available, both upgradable and non-upgradable. An upgradable bridge can be upgraded to a router; a non-upgradable bridge cannot.

These bridge-only units are pre-configured; no further configuration is required. The unit comes up in bridge mode automatically.

Upgrading an upgradable bridge to become a router requires the addition of a software option key. The software option key turns on the IP Routing feature. To read about software option keys, see page 124.

When to Use Routing or Bridging or Both

The following charts describe the operational characteristics of the router when you enable routing, bridging, or both routing and bridging.

IP/IPX Routing On	Bridging to/from Remote Router Off
Data packets carried	IP (TCP, UDP), IPX
Operational characteristics	Basic IP, IPX connectivity
Typical usage	When only IP/IPX traffic is to be routed and all other traffic is to be ignored. For IP, used for Internet access. **Note:** This is the most easily controlled configuration.

IP/IPX Routing On	Bridging to/from Remote Router On
Data packets carried	IP/IPX routed; all other packets bridged.

Operational characteristics	IP/IPX routing; allows other protocols, such as NetBEUI (that can't be routed), to be bridged.
Typical usage	When only IP/IPX traffic is to be routed but some non-routed protocol is required. Used for client/server configurations.

IP/IPX Routing Off	Bridging to/from Remote Router On
Data packets carried	All packets bridged.
Operational characteristics	Allows use of protocols that can't be routed (such as NetBEUI).
Typical usage	Peer-to-peer bridging and when the remote end supports only bridging.

How Routing and Bridging Work Together

The router follows these rules when operating as both a router and a bridge:

- The router operates as a router for network protocols that are enabled for routing (IP or IPX).
- The router operates as a bridge for protocols that are not supported for routing.
- Routing takes precedence over bridging; i.e., when routing is active, the router uses the packet's protocol address information to route the packet.
- If the protocol is not supported, then bridging uses the MAC address information to forward the packet.

Routing and Bridging Controls

The router can be configured to perform general routing and bridging while allowing you to set specific controls.

- One remote router can be designated as the outbound default bridging destination. All outbound bridging traffic with an unknown destination is sent to the default bridging destination.
- Bridging can be enabled or disabled for specific remote routers.
- Routing can be enabled or disabled for the entire router and for individual remotes.

Operation of the router is influenced by routing and bridging controls and filters set during router configuration as well as automatic spoofing and filtering performed by the router. For example, general IP or IPX routing, and routing or bridging from specific remote routers are controls set during the configuration process.

Spoofing and filtering, which minimize the number of packets that flow across the WAN, are performed automatically by the router. For example, RIP routing packets and certain NetBEUI packets are spoofed even if only bridging is enabled.

xDSL WAN Interfaces

Routers are available whose WAN interfaces conform to various DSL standards. Separate sets of commands are provided for each type of DSL. The following lists each supported DSL standard with a link to its set of commands:

ADSL *ADSL Commands, page 326*

HDSL *HDSL Commands, page 336*

IDSL *IDSL Commands, page 339*

SDSL *SDSL Commands, page 342*

SHDSL *SHDSL Commands, page 346*

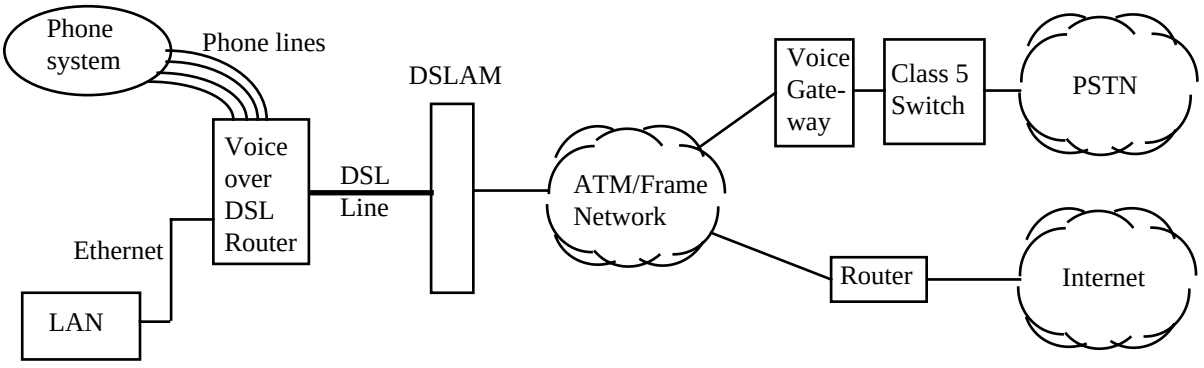
The WAN interface of the router is displayed each time the router reboots, as in the following IDSL example:

```
Efficient 5871 IDSL Router (120-5871-001/2) v5.0.0 Ready
Login:
```

Voice Routing

A Voice over DSL (VoDSL) router allows the delivery of both telephony (voice) and data services over a single DSL line. It acts as an Integrated Access Device (IAD), residing on the customer premises and connecting to a DSL circuit. As such, it serves as a circuit/packet gateway and provides standard telephone service as well as Internet service via an Ethernet connection. Thus, the user has access to toll-quality telephone lines and continuous, high-speed Internet and remote LAN services over a single copper loop.

This diagram illustrates how a Voice over DSL router connects both a phone system to the PSTN and a LAN to the Internet over the same DSL line.



Features provided in your VoDSL router may include:

- Support for these voice gateways:
  - Jetstream (proprietary)
  - CopperCom (proprietary)

    - Tollbridge (proprietary)
    - ATM standards-based (ATM Forum document VMOA-0145.00), also known as BLES (broadband loop emulation service)

- Upstream traffic shaping (bandwidth management) of data when the telephony interface is active
- ADPCM or PCM voice encoding
- Local echo canceling (G.168)

Upstream over subscription is supported for most gateways. If the bandwidth is insufficient to support a new outgoing call, the user does not receive a dial tone when going off hook. Downstream over subscription should be managed by the gateway. Currently, if the bandwidth is occupied by calls and a new call is received, all users may hear clicking.

Configuring Your Telephony Services

Router models are available to support telephony services over both ATM and Frame Relay networks.

- For telephony over ATM, the VPI/VCI is automatically set. (For most routers, it is set to 0*39).
- For telephony over Frame Relay, the DLCI is automatically set to 22. The value must match your service provider's value.

Use the Web GUI to verify the VPI/VCI or DLCI numbers for the data and voice connections.You can change the value if necessary (see *Voice Routing (VoDSL) Troubleshooting, page 191*).

The phone number for each port is set by your voice service provider.

The phone dial tone is provided by the Class 5 switch via the voice gateway at your regional switching center (RSC). All voice features of the switch are passed through to the phone set. The router supports the calling services that you subscribe to from your service provider, such as call forwarding, caller ID, messaging, etc.

The bandwidth required to initiate an upstream call is always about 80 Kbps (64 Kb plus overhead), whether PCM or ADPCM encoding is used. Although for ADPCM only about 40 Kb is used (32 Kb plus overhead), 80Kbps is always reserved because the ADPCM call can switch to a PCM call on the fly. This switching is done for all V.90 calls and some fax calls by the voice gateway.

Configuration for voice and data routing can be performed using the Web-based Easy Setup configuration program. For ATM standards-based gateways, the voice profile must match the configuration of the voice gateway (see *Changing Your ATM Standard Voice Profile, page 23*).

You can use the Port Monitor GUI program to see the voice PVC and the last event message. To see LMI statistics for a frame relay router, use the **frame stats** command (page 334). To see AAL2 statistics for the voice gateway, use the **voice l2stats** command (Jetstream gateway only). For other commands to monitor telephony services, see Trouble-Shooting Telephony Services (page 191).

Changing Your ATM Standard Voice Profile

If your voice gateway is an *ATM standards-based* gateway, the voice profile *must* match the configuration of the voice gateway. **Note:** You do not set a voice profile for the other supported gateways.

The voice profile determines the following attributes:

- Voice compression: ADPCM32 or PCM *or* PCM only?

- Silence suppression supported: yes or no?
- Voice cell payload size: 44 bytes or 40 bytes?

You can display and change your active voice profile. The default voice profile is 9. To display the current voice profile, enter this command:

voice profile

To change your active voice profile, specify the profile number on the **voice profile** command.

voice profile <*profile*>

Note: Currently, only profiles **9** and **10** are supported.

profile	**ADPCM32?**	**Silence Suppression?**	**Payload Size?**
7	Yes	Yes	44 bytes
8	No	Yes	44 bytes
9	No	No	44 bytes
10	Yes	No	44 bytes
11	Yes	No	40 bytes
12	Yes	Yes	40 bytes

For example, the following command selects voice profile 10:

```
# voice profile 10
The active profile has been changed
Profile 10 active, pcm or adpcm32, 44 byte packets
```

Selecting µlaw or Alaw Encoding

The router may allow you to select either µlaw or alaw digital audio encoding (Jetstream and Coppercom gateways only). Your selection should match the encoding used by the audio you are receiving. The default is µlaw encoding. To see the current selection, use the command:

dsp vr or **dsp vpinfo**.

To change the encoding selection, use the following command:

dsp ecode < alaw | ulaw >

CAS Refresh During Idle State

Certain ATM standards-based gateways require that the router send CAS (channel associated signaling) refresh packets periodically so that the gateway knows that the voice port is still available. Other gateways do not require CAS refresh signaling during an idle state (when no voice is present). The router can operate in either mode.

The two modes are called active and always; the default is active mode, in which CAS refresh is *not* performed during an idle state. To see the current setting, enter this command:

voice refreshcas

To have CAS refresh signals sent *both* when voice is present *and* during an idle state, enter this command:

voice refreshcas always

To have CAS refresh signals sent *only* when voice is present, enter this command:

voice refreshcas active

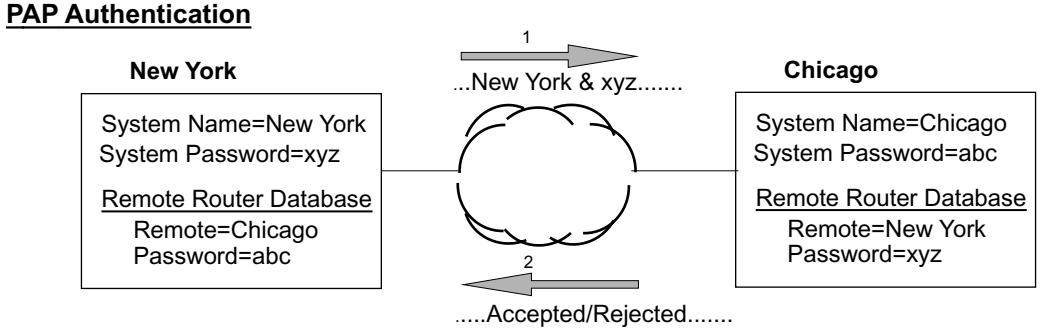
Note: A mode change is effective immediately. However, you must **save** the change if it is to persist across reboots.

PAP/CHAP Security Authentication

The router supports PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol) under PPP.

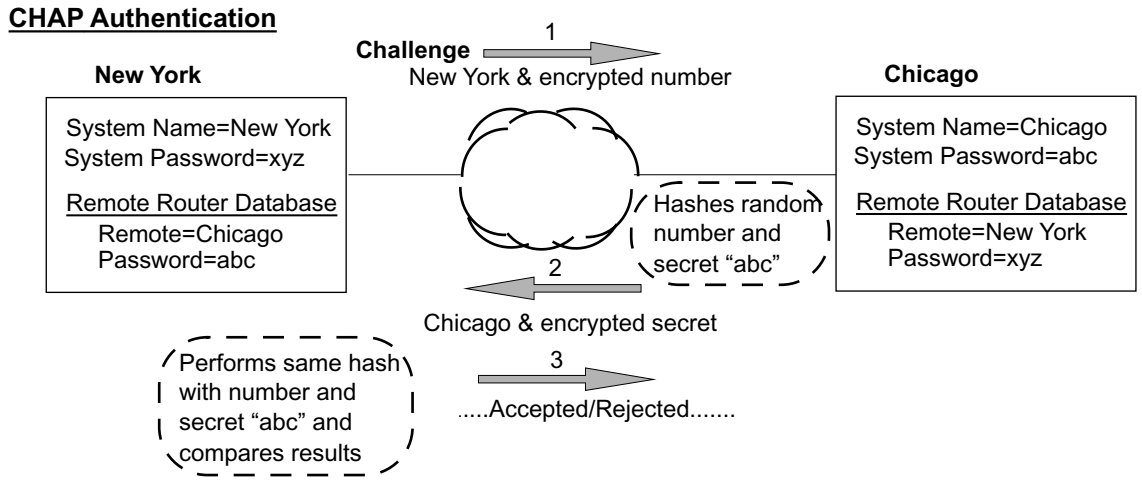
Security authentication may not be required due to the nature of the connection in a DSL environment (traffic occurs on a dedicated line/virtual circuit. However, authentication may be specifically required by the remote end, the ISP, or the NSP. When authentication is not required, security can be disabled with the command **remote disauthen** (page 299).

PAP provides verification of passwords between routers using a two-way handshake. One router (peer) sends the system name and password to the other router. Then the other router (known as the authenticator) checks the peer's password against the configured remote router's password and returns acknowledgment.



CHAP is more secure than PAP because unencrypted passwords are not sent across the network. CHAP uses a three-way handshake. One router (known as the authenticator) challenges the other router (known as the peer) by generating a random number and sending it along with the system name. The peer then applies a one-way hash algorithm to the random number and returns this encrypted information along with the system name.

The authenticator then runs the same algorithm and compares the result with the expected value. This authentication method depends upon a password or secret known only to both ends.



Authentication Process

The authentication process occurs regardless of whether a remote router connects to the local router or vice versa, and even if the remote end does not request authentication. It is a *bi-directional process*, where each end can authenticate the other using the protocol of its choice (provided the other end supports it).

During link negotiation (LCP), each side of the link negotiates which protocol to use for authentication during the connection.

Note: If desired, you can override the negotiation of an authentication protocol and force the local router to use the designated protocol. To designate PAP or CHAP, use the command **system authen** (page 236).

If both routers have PAP authentication, then they negotiate PAP authentication. Otherwise, the local router *always* requests CHAP authentication first; if CHAP is refused, PAP is requested. If the remote does not accept either PAP or CHAP, the link is dropped; i.e., the router does not communicate without a minimum security level. On the other hand, the local router does accept any authentication scheme required by the remote, including no authentication at all.

CHAP Authentication

For CHAP, the router issues a CHAP challenge request to the remote side. The challenge includes the system name and random number. The remote end, using a hash algorithm, transforms the name and number into a response value. When the remote end returns the challenge response, the router can validate the response challenge value using the entry in the remote router database. If the response is invalid, the call is disconnected.

If the other end negotiated CHAP, the remote end can, similarly, request authentication from the local router. The router uses its system name and password to respond to the CHAP challenge.

PAP Authentication

For PAP, when a PAP login request is received from the remote end, the router checks the remote router PAP security using the remote router database. If the remote router is not in the remote router database or the remote router password is invalid, the call is disconnected. If the remote router and password are valid, the local router acknowledges the PAP login request.

If PAP was negotiated by the remote end for the remote-side authentication, the router issues PAP login requests *only* if it knows the identity of the remote end. The identity is known if the call was initiated from the router, or if the remote end returned a successful CHAP challenge response. For security reasons, the router *never* identifies itself using PAP without first knowing the identity of the remote router.

If PAP was negotiated by the remote end for the local side of the authentication process and the minimum security level is CHAP, as configured in the remote router database, the link is dropped as a security violation.

Authentication Passwords

Access to the router is controlled by an administration password set by the command **system admin** (page 236). As part of the router configuration, you may set the following authentication passwords:

- **System authentication password** — the default system password used to access any remote router. Remote sites use this password to authenticate the local site.

  This default authentication password is set by the command **system passwd** (page 252).

- **System override password** — optional password used only to connect to a specific remote router for authentication by that remote site.

  To specify a unique system override password for a remote router, use the command **remote SetOurPasswd** (page 315). This password is used instead of the general system password *only* for connecting to a specific remote router. This allows you to set a unique CHAP or PAP authentication password for authentication of the local site by the remote site *only* when the router connects to that remote site.

  A common use for the system override password is to set the password assigned to you by your Internet Service Provider (ISP). Similarly, the system name of the local router (set by the command **system name**) can be overridden for connecting to a specific remote with the command **remote setOurSysName** (page 316).

- **Remote authentication password** — password used by the router to authenticate the remote site. Each remote router entered in the remote router database has a password used when the remote site attempts to gain access to the local router.

  To set the remote authentication password, use the command **remote setpasswd** (page 316).

Authentication Levels

The router also uses security levels, as follows:

- Remote authentication protocol — Each remote router entered in the remote router database has a minimum security level that must be negotiated before the remote router gains access to the local router.
- System authentication protocol — A system-wide control is available for overriding the minimum security level in the entire remote router database.

Interoperability Between the Router and Other Equipment

The router uses industry-wide standards to ensure compatibility with routers and equipment from other vendors. To interoperate, the router supports standard protocols on the physical level, data link level, and network level. For two systems to communicate directly, they must use the same protocol at each level.

Level	Interoperability	Determined by
Physical media	Hardware and electrical signaling	Router Ethernet and modem hardware interfaces for copper wire or fiber cable
Data link	Packet transmission method (frame type or encapsulation method)	Router hardware and software kernel. Can be Ethernet, ATM, or Frame Relay
Network layer	Network protocol	Router configuration. Can be IP or IPX

The data-link protocol level defines the transmission of data packets between two systems over the LAN or WAN physical link. The frame type or encapsulation method defines a way to run multiple network-level protocols over a single LAN or WAN link. Most protocols do not support negotiable options, except for PPP.

The router supports both ATM (Asynchronous Transfer Mode) and Frame Relay transmission. ATM transport uses fixed-length cells; Frame Relay transport uses variable-length packets.

The router supports the following WAN encapsulations:

- PPP (VC multiplexing)
- PPP (LLC multiplexing)
- PPPoE (PPP over Ethernet)
- RFC 1483 (for ATM)
- RFC 1483 with MAC encapsulated routing (for ATM)
- FRF8 (for ATM)
- RFC 1490 (for Frame Relay)
- RFC 1490 with MAC encapsulated routing (for Frame Relay)

The packet formats for these encapsulation methods are given in *Encapsulation Options, page 29*.

Protocol Conformance

The router conforms to RFCs designed to address performance, authentication, and multi-protocol encapsulation. The following RFCs are supported:

- RFC 1058 Routing Information Protocol (RIP)
- RFC 1144 Compressing TCP/IP headers (Van Jacobson)
- RFC 1220 Bridging Control Protocol (BNCP)
- RFC 1332 IP Control Protocol (IPCP)

- RFC 1334 Password Authentication Protocol and Challenge Handshake Authentication Protocol (PAP/CHAP)
- RFC 1389 RIP2
- RFC 1483 Multiprotocol Encapsulation over ATM Adaptation Layer 5
- RFC 1490 Multiprotocol Interconnect over Frame Relay
- RFC 1542 DHCP Relay Agent
- RFC 1552 Novell IPX Control Protocol (IPXCP)
- RFC 1577 Classical IP and ARP over ATM
- RFC 1631 Network Renumbering
- RFC 1661 Point-to-Point Protocol (PPP)
- RFC 1723 RIP Version 2
- RFC 1769 Simple Network Time Protocol (SNTP)
- RFC 1877 Automatic IP / DNS
- RFC 1962 PPP Compression Control Protocol (CCP)
- RFC 1969 PPP DES Encryption Protocol (ECP)
- RFC 1973 PPP in Frame Relay
- RFC 1974 PPP Stac LZS Compression Protocol
- RFC 1990 Multi-Link Protocol (MLP)
- RFC 1994 User Authentication PAP / CHAP
- RFC 2104 HMAC: Keyed-Hashing for Message Authentication
- RFC 2131 Dynamic Host Configuration Protocol (DHCP)
- RFC 2132 DHCP Client
- RFC 2364 PPP over AAL5
- RFC 2401 Security Architecture for the Internet Protocol
- RFC 2402 IP Authentication Header
- RFC 2403 The Use of HMAC-MD5-96 within ESP and AH
- RFC 2404 The Use of HMAC-SHA-1-96 within ESP and AH
- RFC 2405 The ESP DES-CBC Cipher Algorithm With Explicit IV
- RFC 2406 IP Encapsulating Security Payload (ESP)
- RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP
- RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 2409 The Internet Key Exchange (IKE)
- RFC 2410 The NULL Encryption Algorithm and Its Use with IPSec
- RFC 2412 The OAKLEY Key Determination Protocol
- RFC 2419 PPP DES Encryption v2
- RFC 2451 The ESP CBC-Mode Cipher Algorithms

IP Routing

IP routing support, in conformance with RFC 791, provides the ability to process TCP/IP frames at the network layer for routing. IP routing support includes the Routing Interface Protocol (RIP), in conformance with RFC 1058 (RIP v.1) and RFC 1723 (RIP v.2).

IPX Routing

IPX routing conforms to the Novell® NetWare™ IPX Router Development Guide, Version 1.10.

Encapsulation Options

This section describes the packet format for each encapsulation option supported by the router.

The encapsulation method used by a remote is defined by the **remote setProtocol** command (page 320).

Note: The same encapsulation method must be used by both ends of the connection (the router and the DSLAM).

PPP

This protocol uses VC multiplexing, as defined in RFC 2364; it dedicates a virtual circuit to PPP traffic only. (The other encapsulation method defined in RFC 2364, LLC multiplexing, is described in the next section, *PPPLLC*.)

Each packet begins with a one- or two-byte protocol ID. Typical IDs are:

0xc021	LCP
0x8021	IPCP
0x0021	IP
0x002d	Van Jacobson compressed TCP/IP
0x002f	Van Jacobson uncompressed TCP/IP
0x8031	Bridge NCP
0x0031	Bridge Frame

The command for this encapsulation option is: **remote setProtocol PPP** <*remoteName*> (page 320).

Note: With PPP over ATM, the address and control fields (i.e., FF03) are never present; this also is the case for LCP packets.

PPPLLC

This protocol (LLC-multiplexed) allows PPP traffic to be carried simultaneously with other traffic on a single virtual circuit (as opposed to the PPP method of encapsulation—VC multiplexing—which dedicates a virtual circuit to PPP traffic only).

Each PPP packet is prepended with the sequence 0xFEFE03CF. Thus, an LLC packet has the format: 0xFEFE03CF 0xC021.

The command for this encapsulation option is: **remote setProtocol PPPLLC** <*remoteName*> (page 320).

RFC 1483 or RFC 1490

Bridging

User data packets are prepended by the sequence 0xAAAA0300 0x80c20007 0x0000 followed by the Ethernet frame containing the packet.

802.1D Spanning Tree packets are prepended with the header 0xAAAA0300 0x80C2000E.

Routing

IP packets are prepended with the header 0xAAAA0300 0x00000800.

IPX packets are prepended with the header 0xAAAA0300 0x00008137.

For this encapsulation option, the commands, as described on page 320, are:

remote setProtocol RFC1483 <*remoteName*> (for ATM)

remote setProtocol FR <*remoteName*> (for Frame Relay - RFC 1490)

MAC Encapsulated Routing: RFC 1483MER (ATM) or RFC 1490MER (Frame Relay)

MER encapsulation allows IP packets to be carried as bridged frames, but does not prevent bridged frames from being sent as well, in their normal encapsulation format: RFC 1483 (ATM) or RFC 1490 (Frame Relay).

If IP routing is enabled, then IP packets are prepended with the sequence 0xAAAA0300 0x80c20007 0x0000 and sent as bridged frames. If IP routing is not enabled, then the packets appear as bridged frames.

The commands for this encapsulation option are:

remote setProtocol RFC1483MER <*remoteName*> (for ATM)

remote setProtocol MER <*remoteName*> (for Frame Relay)

FRF8

IP packets have prepended to them the following sequence: 0x03CC.

The command for this encapsulation option is: **remote setprotocol FRF8** <*remoteName*>

Note: This protocol allows sending ATM over Frame Relay.

rawIP

IP packets do not have any protocol headers prepended to them; they appear as IP packets on the wire. Only IP packets can be transported since there is no possible method to distinguish other types of packets (bridged frames or IPX).

The command for this encapsulation option is: **remote setProtocol rawIP** <*remoteName*>

Router System and Configuration Files

The system software and configuration information for the router are in its DOS-compatible file system. The file system commands, similar to DOS commands, are described in *File System Commands, page 226*.

It is wise to keep a backup copy of the system and configuration files. For more information on the backup and restoration of configuration files, see page 179.

Any file contained within the system may be retrieved or replaced using the TFTP protocol. Specifically, configuration files and the operating system upgrades can be updated. Only one copy of the router software is allowed in the router's FLASH memory. For more information on these topics, see *Managing the Router, page 165*.

Note: Users should not delete any of these files, unless advised to do so by Technical Support.

The router software files are as follows:

KERNEL.F2K Router system software (KERNEL.FP1 for IDSL routers).

ASIC.AIC FPGA (Field Programmable Gate Array) file that provides the logic that customizes the router hardware (not present in the 5950 or 5871 models).

The router configuration files are as follows:

SYSTEM.CNF System configuration information, including:

DOD	Remote router database
SYS	System settings: name, message, authentication method, and passwords
ETH	Ethernet LAN configuration settings

DHCP.DAT DHCP data.

FILTER.DAT Bridge filters.

ETH.DEF File used by the manufacturer to set a default Ethernet configuration.

ATM.DAT ATM configuration.

ATOM.DAT ATM configuration.

SDSL.DAT SDSL configuration.

SHDSL.DAT SHDSL configuration.

DMT.DAT DMT configuration.

IPSEC.DAT IPSec configuration.

IKE.DAT IKE configuration.

The following files are for automatic execution of command scripts. For more information, see page 183.

AUTOEXEC.BAT Autoexec file of commands to run on next reboot.

AUTOEXEC.OLD Autoexec file that has run already

The following script files are used for creating firewall filters via the web GUI.

maxsec.txt Maximum security

medsec.txt Medium security

minsec.txt Minimum security

nosec.txt No security

The keys for software options that have been purchased are kept in the file **KEYFILE.DAT**. Do not copy the key file from one router to another router, even if the two routers are the same model with the same kernel. The software keys are isomorphic to one and only one router. For more information on software option keys, see page 124.

Chapter 2. Planning for Router Configuration

This chapter describes the basic information you need before you can begin configuring your router. The basic configuration tasks can be performed using the Command Line Interface described in this manual or the graphic interface described in the *User Reference Guide*. (A copy of the *Guide* came with your router; it is also availabe on the web site www.efficient.com.) The basic configuration information is the same for either interface.

The basic configuration tasks include the following:

- Setting names, passwords, PVC numbers, and link and network parameters
- Configuring specific protocol requirements, such as IP or IPX addresses and IP protocol controls
- Activating bridging and routing protocols
- Enabling the Internet firewall filter with IP routing

An alternate configuration method, called Plug & Play, is available with the Copper Mountain CopperEdge™ 200 DSLAM. This method is described at the end of the chapter.

Remote Routers

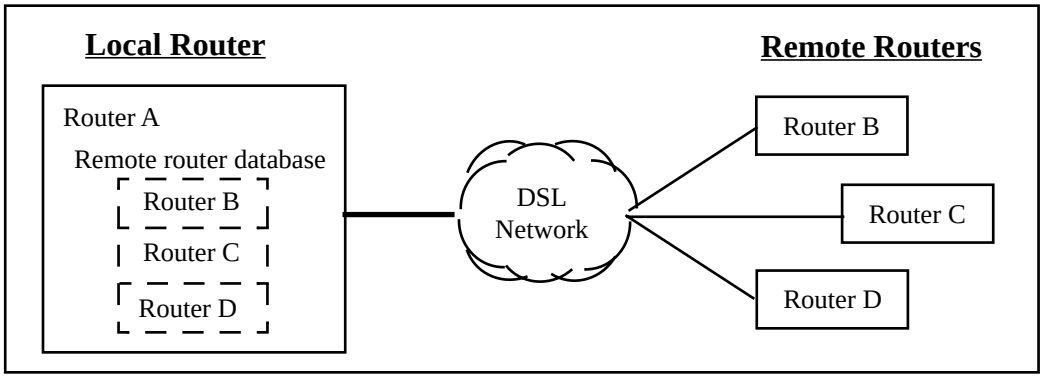
This manual frequently refers to the local router and remote routers, which are defined as follows.

Local router. Router that you are configuring. Also referred to as *target* router.

Remote routers. All the routers to which the local (target) router may connect.

Remote router database. Database which resides in the local router and contains information about the remote routers to which the local router can connect.

The following diagram illustrates these terms.



As shown in the illustration, the remote router database in the local router contains an entry for each remote router. A remote router entry defines:

- Connection parameters
- Security features

- Route addressing and bridging functions

The commands that define information for a remote router entry start with the word **remote** and end with the name of the remote entry. Most of these commands are described in the section REMOTE Commands, on page 289.

Managing the Remote Entries

You can control the use of a remote entry in the remote router database by enabling or disabling its use. To enable a remote, use the **remote enable** command (page 299). To disable a remote, use the **remote disable** command (page 298). The remote remains enabled or disabled across reboots of the router.

The active session for a remote can be stopped and started independently (see the commands **remote stop**, page 324, **remote start**, page 322, and **remote restart**, page 307). These commands allow you to activate configuration changes for a remote without rebooting the router. (Many configuration changes require a save and restart or reboot before the change becomes effective.)

Protocols to be Used

The information needed to configure the router depends on the link protocol and network protocols that are to be used. The link protocol and network protocols used are generally determined by your Network Service Provider.

This chapter is organized in sections that apply to specific protocols.

Note: Use the blank Network Information Worksheets in appendix A to collect your network information.

If you are using Link and Network Protocols:

PPP with:

- **IP Routing**, go to *page 36*
- **IPX Routing**, go to *page 38*
- **Bridging**, go to *page 40*

RFC 1483 or RFC 1490 with:

- **IP Routing**, go to *page 41*
- **IPX Routing**, go to *page 42*
- **Bridging**, go to *page 43*

RFC 1483 MER or RFC 1490 MER (MAC Encapsulated Routing) with:

- **IP Routing**, go to *page 44*

FRF8 with:

- **IP Routing**, go to *page 46*

To configure a Dual-Ethernet Router:

Select one of these two configurations:

Configuring the Dual-Ethernet Router as a Bridge, on page 47

Configuring the Dual-Ethernet Router for IP Routing, on page 47

PPP Link Protocol (over ATM or Frame Relay)

The PPP link protocol is an encapsulation method that can be used over ATM or over Frame Relay . For PPP over Ethernet (PPPoE), see page 103.

PPP over ATM and PPP over Frame Relay use different connection identifiers:

- ATM uses VPI/VCI numbers.
- Frame Relay uses a DLCI number.

IP Routing Network Protocol

To configure the IP network protocol and PPP link protocol, you need the following information.

- **System Names and Authentication Passwords for the Local Router and All Remote Routers**

  For the Local Router:

  You define a system name and authentication password for the local router. Remote routers check the system name and authentication password to authenticate the local router.

  For Remote Routers:

  Although the system names and authentication passwords for the remote routers are defined by the service provider, you *must* have this information because the local router uses it to authenticate the remote router. The name and password are used in both PAP and CHAP authentication. To see how this information is used, refer to PAP/CHAP Security Authentication, on page 25 .

  Note: If the service provider does not support the authentication of remotes by the local router, use the command **remote disauthen** <*remoteName*> to disable the authentication process.

  Note: A sample configuration containing names and passwords is provided in the section *Sample Configuration 1: PPP with IP and IPX, on page 65* .

- **For an ATM router, its VPI and VCI numbers**

  ATM uses two connection identifiers that describe the PVC (Permanent Virtual Circuit). These identifiers are the VPI (Virtual Path Identifier) and the VCI (Virtual Channel Identifier). Your router may have been preconfigured with VPI/VCI numbers. If not, you need to get these numbers from your service provider.

  If you are connecting to multiple remote sites, you need the unique VPI and VCI numbers that identify each remote destination.

- **For a Frame Relay router, its DLCI number**

The DLCI (Data Link Connection Identifier) number applies to Frame Relay routers only. Get your DLCI from your service provider.

- **DNS Internet Account Information (optional)**

  The Domain Name Service (DNS) maps host names to IP addresses. DNS is performed by Domain Name Servers. The router can get DNS information automatically. Or, you can choose to configure DNS manually. Consult with your Network Service Provider to determine if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

- **IP Routing Addresses**

  For the Ethernet interface:

  Ethernet IP Address (Local LAN)
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection. This information is defined by the user or your network administrator.

  Note: An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information. This feature is only used in special circumstances.

  For the WAN interface:

  The following information is defined by your network service provider.

  Source (Local) WAN Port Address
  If Network Address Translation (NAT) is enabled, you must specify a source WAN IP address for the WAN connection to the remote router if IP address negotiation under PPP does not provide one. Check with your network administrator for details on whether the router must communicate in numbered or unnumbered mode and which addresses are required.

  Remote WAN Address
  You may need to specify a remote WAN IP address for the WAN connection to the remote router depending on IP address negotiation under PPP. Check with your network administrator for details on whether the router must communicate in numbered or unnumbered mode and which addresses are required.

  TCP/IP Remote Routes
  An IP route includes an IP address, subnet mask, and metric (a number representing the perceived cost to reach the remote network or station).

  A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. Define the default route to a remote router or, in special circumstances, define an Ethernet gateway. There can be *only one* default route specified.

IPX Routing Network Protocol

To configure the IPX network protocol and PPP link protocol, you need the following information.

- **System Names and Authentication Passwords for the Local Router and All Remote Routers**

  For the Local Router:

  > You define a system name and authentication password for the local router. Remote routers check the system name and authentication password to authenticate the local router.

  For Remote Routers:

  > Although the system names and authentication passwords for the remote routers are defined by the service provider, you *must* have this information because the local router uses it to authenticate the remote router. The name and password are used in both PAP and CHAP authentication. To see how this information is used, refer to PAP/CHAP Security Authentication, on page 25 .

  Note: If the service provider does not support the authentication of remotes by the local router, use the command **remote disauthen** <*remoteName*> to disable the authentication process.

  Note: A sample configuration containing names and passwords is provided in the section *Sample Configuration 1: PPP with IP and IPX, on page 65* .

- **For an ATM router, its VPI and VCI numbers**

  ATM uses two connection identifiers that describe the PVC (Permanent Virtual Circuit). These identifiers are the VPI (Virtual Path Identifier) and the VCI (Virtual Channel Identifier). Your router may have been preconfigured with VPI/VCI numbers. If not, you need to get these numbers from your service provider.

  If you are connecting to multiple remote sites, you need the unique VPI and VCI numbers that identify each remote destination.

- **For a Frame Relay router, its DLCI number**

  The DLCI (Data Link Connection Identifier) number applies to Frame Relay routers only. Get your DLCI from your service provider.

- **IPX routing entries**

  IPX routes define the *paths* to specific destinations. Routers need them so servers and clients can exchange packets. A path to a file server is based on the Internal Network Number of the server. A path to a client is based on the External Network Number (Ethernet) of the client.

  You need the following information (most likely from your network administrator) for IPX routing.

  > **Internal Network Number**
  >
  > It is a logical network number that identifies an individual Novell server. It specifies a route to the services (i.e., file services, print services) that Novell offers. It must be a unique number.
  >
  > **External Network Number (IPX Network Number)**
  >
  > It refers to a physical LAN/wire network segment to which servers, routers, and PCs are connected (Ethernet cable-to-router segment). It must be a unique number.

WAN Network Number

Important: This number is part of the routing information. It only identifies the WAN segment between the two routers. Note that only those two routers need to have the WAN Network Number configured.

Service Advertisement Protocol (SAP)

SAP entries should reflect primary logon servers for the clients on the local LAN. Only the servers on the remote side of the link have to be entered. Local servers do not need to be entered.

Frame Type

With local servers on your LAN, make sure to select the proper frame type for the IPX network number. To determine this, consult with your network administrator. When you have only NetWare clients on your LAN, keep the default (802.2) selected as most clients can support any type. The frame type choices are:

802.2 Default recommended by Novell

802.3 Other most common type

DIX For DEC, Intel, Xerox; this setting is also referred to as “Ethernet II”, and it is becoming obsolete.

Note: For step-by-step information on how to configure IPX routing, see Configuring IPX Routing, on page 408.

Bridging Network Protocol

To configure bridging as the network protocol and PPP as the link protocol, you need the following information:

- **System Names and Authentication Passwords for the Local Router and All Remote Routers**

  For the Local Router:

  > You define a system name and authentication password for the local router. Remote routers check the system name and authentication password to authenticate the local router.

  For Remote Routers:

  > Although the system names and authentication passwords for the remote routers are defined by the service provider, you *must* have this information because the local router uses it to authenticate the remote router. The name and password are used in both PAP and CHAP authentication. To see how this information is used, refer to PAP/CHAP Security Authentication, on page 25 .

  Note: If the service provider does not support the authentication of remotes by the local router, use the command **remote disauthen** <*remoteName*> to disable the authentication process.

  Note: A sample configuration containing names and passwords is provided in the section *Sample Configuration 1: PPP with IP and IPX, on page 65* .

- **For an ATM router, its VPI and VCI numbers**

  ATM uses two connection identifiers that describe the PVC (Permanent Virtual Circuit). These identifiers are the VPI (Virtual Path Identifier) and the VCI (Virtual Channel Identifier). Your router may have been preconfigured with VPI/VCI numbers. If not, you need to get these numbers from your service provider.

  If you are connecting to multiple remote sites, you need the unique VPI and VCI numbers that identify each remote destination.

- **For a Frame Relay router, its DLCI number**

  The DLCI (Data Link Connection Identifier) number applies to Frame Relay routers only. Get your DLCI from your service provider.

- **DNS Internet Account Information (optional)**

  The Domain Name Service (DNS) maps host names to IP addresses. DNS is performed by Domain Name Servers. The router can get DNS information automatically. Or, you can choose to configure DNS manually. Consult with your Network Service Provider to determine if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

RFC 1483/RFC 1490 Link Protocols

The link protocols RFC 1483 and RFC 1490 are multiprotocol encapsulation methods. RFC 1483 is used over ATM; RFC 1490 is used over Frame Relay.

RFC 1483 and RFC 1490 combined with the IP, IPX, or Bridging network protocols share the same configuration characteristics, except for the connection identifiers: VPI/VCI numbers are used for RFC 1483 and a DLCI number is used for RFC 1490.

Obtain the information as described in the appropriate section. This data will be used later to configure your router using the Command Line Interface (see Configuration Tables, on page 52).

IP Routing Network Protocol

To configure IP as the network protocol and RFC 1483 or RFC 1490 as the link protocol, you need the following information:

- **VPI and VCI Numbers (for RFC 1483)**

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, get these numbers from your Network Service Provider.

  If you are connecting to multiple remote sites, get additional VPI and VCI numbers from your Network Service Provider. These numbers identify the remote destination and must be unique for each remote.

- **DLCI (for RFC 1490)**

  The DLCI number applies to Frame-Relay routers only; it identifies your connection. Get you DLCI from your Network Service Provider.

- **DNS Internet Account Information (optional)**

  Consult with your Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

- **IP Routing Entries**

  For the Ethernet Interface:

  This information is defined by the user or the Network Administrator.

  Ethernet IP Address (Local LAN)
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

  TCP/IP Ethernet Routes
  You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information.

For the WAN Interface:

This information is obtained from the Network Administrator.

Source (Target/Local) WAN Port Address
If Network Address Translation (NAT) is enabled, you *must* specify a source WAN IP address for the WAN connection to the remote router.

If NAT is *not* enabled, you may need to specify a source WAN IP address for the WAN connection to the remote router.

TCP/IP Remote Routes
An IP route includes an IP address, subnet mask, and metric (a number representing the perceived cost to reach the remote network or station).

A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. Define the default route to a remote router or, in special circumstances, define an Ethernet gateway. There can be only one default route specified.

IPX Routing Network Protocol

To configure IPX as the network protocol and RFC 1483 or RFC 1490 as the link protocol, you need the following information:

- **VPI and VCI Numbers (for RFC 1483)**

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, get these numbers from your Network Service Provider.

  If you are connecting to multiple remote sites, get additional VPI and VCI numbers from your Network Service Provider. These numbers identify the remote destination and must be unique for each remote.

- **DLCI (for RFC 1490)**

  The DLCI number applies to Frame-Relay routers only; it identifies your connection. Get you DLCI from your Network Service Provider.

- **IPX Routing Entries**

  IPX routes define the *paths* to specific destinations. Routers need them so servers and clients can exchange packets. A path to a file server is based on the Internal Network Number of the server. A path to a client is based on the External Network Number (Ethernet) of the client.

  You need the following information (most likely from your network administrator) for IPX routing.

  Internal Network Number

  It is a logical network number that identifies an individual Novell server. It specifies a route to the services (i.e., file services, print services) that Novell offers. It must be a unique number.

  External Network Number (IPX Network Number)

  It refers to a physical LAN/wire network segment to which servers, routers, and PCs are connected (Ethernet cable-to-router segment). It must be a unique number.

WAN Network Number

Important: This number is part of the routing information. It is only used to identify the WAN segment between the two routers. Note that only those two routers need to have the WAN Network Number configured.

Service Advertisement Protocol (SAP)

SAP entries should reflect primary logon servers for the clients on the local LAN. Only the servers on the remote side of the link have to be entered. Local servers do not need to be entered.

Frame Type

With local servers on your LAN, make sure to select the proper frame type for the IPX network number. To determine this, consult with your network administrator. When you have only NetWare clients on your LAN, keep the default (802.2) selected as most clients can support any type. The frame type choices are:

802.2 Default recommended by Novell

802.3 Other most common type

DIX For DEC, Intel, Xerox; this setting is also referred to as “Ethernet II”, and it is becoming obsolete.

Bridging Network Protocol

To configure bridging as the network protocol and RFC 1483 or RFC 1490 as the link protocol, you need the following information:

- **VPI and VCI Numbers (with RFC 1483)**

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, get these numbers from your Network Service Provider.

  If you are connecting to multiple remote sites, get additional VPI and VCI numbers from your Network Service Provider. These numbers identify the remote destination and must be unique for each remote.

- **DLCI (with RFC 1490)**

  The DLCI number applies to Frame-Relay routers only; it identifies your connection. Get you DLCI from your Network Service Provider.

- **DNS Internet Account Information (optional)**

  Consult with your Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

MAC Encapsulated Routing

MAC Encapsulated Routing (MER) allows IP packets to be carried as bridged frames (bridged format). The link protocol RFC 1483 with MER (referred to as RFC 1483MER) is a multiprotocol encapsulation method over ATM

used by ATM routers. RFC 1490 with MER (referred to as RFC 1490MER) is a multiprotocol encapsulation method over Frame Relay used by Frame-Relay routers.

RFC 1483MER and RFC 1490MER combined with the IP, IPX, or Bridging network protocols share the same configuration characteristics, except for the connection identifiers: VPI/VCI numbers are used for RFC 1483MER and a DLCI number is used for RFC 1490.

Obtain the information as described in the appropriate section. This data will be later used to configure your router using the Command Line Interface (see Configuring MAC Encapsulated Routing: RFC 1483MER / RFC 1490MER with IP Routing, on page 59).

IP Routing Network Protocol

- **VPI and VCI Numbers (for RFC 1483MER)**

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, get these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, get additional VPI and VCI numbers from your Network Service Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- **DLCI (for RFC 1490MER)**

  The DLCI number applies to Frame Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a DLCI (Data Link Connection Identifier). The DLCI is an address identifying your connection.

- **DNS Internet Account Information (optional)**

  This information is obtained from the Network Service Provider. Consult with your Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

  Note: If you intend to only connect to the Internet, enter this information using the Internet Quick Start configurator.

- **IP Routing Entries**

  ### For the Ethernet Interface

  This information is defined by the user or the Network Administrator.

  Ethernet IP Address (Local LAN)
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

  TCP/IP Ethernet Routes
  You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information between them.

For the ATM WAN Interface

This information is obtained from the Network Administrator or the Network Service Provider.

Source (Target/Local) WAN Port Address and Mask
You must specify a Source WAN IP address for the WAN connection to the remote router (whether or not Network Address Translation is enabled). The Source WAN Address is the address of the local router on the remote network. The mask is the mask used on the remote network. Check with your system administrator for details.

TCP/IP Remote Routes
If you are using RFC 1483MER or RFC 1490MER, the IP route includes an IP address, subnet mask, metric (a number representing the perceived cost in reaching the remote network or station), and a *gateway*. The gateway address that you enter is the address of a router on the remote LAN. Check with your system administrator for details.

A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. You will need to define the default route to a remote router or, in DLCI (special circumstances, define an Ethernet gateway. There can be only one default route specified.

FRF8 Link Protocol

The FRF8 link protocol is an encapsulation method that allows an ATM router to interoperate with a Frame- Relay network.

FRF8 is only used in conjunction with the IP network protocol. Obtain the information described below. This data will be used later to configure your router using the Command Line Interface (see Configuration Tables, on page 52).

IP Routing Network Protocol

- **VPI and VCI Numbers**

  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, get additional VPI and VCI numbers from your Network Service Provider. These numbers identify the remote destination and must be unique for each remote.

- **DNS Internet Account Information (optional)**

  Consult your Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

  Note: If you intend to connect only to the Internet, enter this information using the Internet Quick Start configurator.

- **IP Routing Entries**

  ### For the Ethernet Interface

  This information is defined by the user or the Network Administrator.

  Ethernet IP Address (Local LAN)
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

  TCP/IP Ethernet Routes
  You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information.

For the ATM WAN Interface

This information is obtained from the Network Administrator or the Network Service Provider.

Source (Target/Local) WAN Port Address and Mask
You *must* specify a Source WAN IP address for the WAN connection to the remote router (whether or not Network Address Translation is enabled). The Source WAN address is the address of the local router on the remote network. The mask is the mask used on the remote network.

TCP/IP Remote Routes
If you are using FRF8, the IP route includes an IP address, subnet mask, metric (a number representing the perceived cost to reach the remote network or station).

A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. Define the default route to a remote router or, in special circumstances, define it to an Ethernet gateway. There can be only one default route specified.

Dual-Ethernet Router Configuration

The Dual-Ethernet router has two interfaces:

ETH/0	The router's hub with four 10Base-T connectors
ETH/1	The single 10Base-T connector (or the second hub on the new hardware model)

Bridging is enabled by default when the router boots up. IP and IPX routing are disabled.

The router's default IP address is 192.168.254.254.

DHCP is enabled by default and the router's DHCP server issues IP addresses to any PC request. The DHCP default IP pool is 192.168.254. 2 through 192.168.254.20.

To connect to the router, use the router's default IP address using a Telnet session, for example, and any 10Base-T port on the router.

Configuring the Dual-Ethernet Router as a Bridge

This router is configured by default as a bridge and no configuration steps are needed. The user needs only establish a connection to the remote location (to the Internet Service Provider, for example).

Bridging is enabled by default when the router boots up. IP and IPX routing are disabled.

Configuring the Dual-Ethernet Router for IP Routing

The **eth** commands are used to configure the Dual-Ethernet router for IP routing. Refer to the section *Dual-Ethernet Router (ETH) Commands, on page 332*, for usage and syntax information.

The last parameter on an **eth** command determines which interface is being configured (0 for ETH/0, 1 for ETH/1).

Each interface (ETH/0 and ETH/1) must be configured. A minimum of one route must be defined to have a working configuration. This is generally a default route on the ETH/1 interface where all traffic otherwise specified is automatically forwarded. This default route is: 0.0.0.0 255.255.255.255 1.

The gateway address is the IP address supplied by your Internet Service Provider or Network Administrator.

You can customize your router by using the scripting feature, which loads batch files of preset configuration commands into the router (refer to the *Batch File Command Execution, on page 183* section).

A Dual-Ethernet router sample configuration with IP Routing is provided in the *Sample Configuration 3: Configuring a Dual-Ethernet Router for IP Routing, on page 77* section.

Copper Mountain Plug & Play

An alternate configuration method, called Plug & Play, is available when using the Copper Mountain CopperEdge™ 200 DSLAM, version 3.0, and the router models that support Copper Mountain, that is, 5871 IDSL, 5851 SDSL, and 7851 SDSL IAD.

Plug & Play eliminates the need for users or administrators to configure CPEs or IADs locally. It allows the Copper Mountain CE200 DSLAM and all routers connected to it to exchange information via both Copper Mountain Control Protocol (CMCP) messages and DHCP messages. The CE200 and its routers are able to exchange the following types of data: IP addresses, net masks (for both voice and data VCs), voice gateway type, voice gateway IP address, fragmentation capabilities, maximum number of voice channels, and CPE data function and data encapsulation.

Plug & Play can run with the following network models as specified by the DSLAM:

HDIA Network Model

HDIA stands for High Density IP Access. This network model applies to both the data and voice VCs configured for an IAD attached to a single DSL port to the CE200. It allows you to use limited IP addressing space with maximum efficiency. It allows IADs (or routers) and hosts to be on different premises but still on the same subnet. In this network model, an IAD can be configured as a bridge or as a router.

Cross-Connect Network Model

With the Cross-Connect network model, the CE200 can multiplex point-to-point connections over different WAN interfaces. It can also provide different encapsulation types and translations between different encapsulation types. For example, the Cross-Connect network model now functions as a vehicle for converting disparate frame formats between the DSL and WAN interfaces, as well as frame formats into ATM formats.

Copper VPN Network Model

With the Copper VPN™ network model, either in static forwarding mode or auto learning mode, subscriber links act as wide-area extensions of a DSL remote LAN. Unlike the VWAN network model, CopperVPN does not rely on the upstream device for integration of routing and bridging functions. The CE200 learns the IP and MAC address and port number for every host supported by the CPEs.

For full details on the network models, please see your Copper Mountain documentation.

Plug & Play Configuration Process

The Plug and Play configuration process proceeds as follows:

1. The DSLAM is configured to supply the router with its IP address, subnet mask, default gateway and primary DNS server.

2. When the router is placed on the Copper Mountain DSL line, it trains with the DSLAM and senses the encapsulation type and relevant layer 2 parameters such as the DLCI being used (DLCI 528).

3. Once it senses these parameters, by default, it enables IP routing and activates the built-in remote router database entry named **configuredforCMPPlay**.

4. The router then sends a DHCP request to the WAN for its IP address information.

5. IP filters are also automatically created for UDP port 500 to and from the router.

6. The Copper Mountain DSLAM, despite being a layer 2 pass-through device in most network models, intercepts this request and answers it with the values configured in step 1.

Bridge or Router?

If the router is to be used as a bridge (that is, as a pass-through device), you need to enter a command before you place the router on the Copper Mountain DSL line. The command is as follows:

frame cmPPlay bridge

Then, after it is placed on the DSL line, the router automatically activates in bridge mode. Otherwise, the router automatically activates in router mode.

To see the current mode of the router, you can list the remote entry **configuredForCMPPlay** as shown in the next section. In bridging mode, the **Bridging enabled** line is set to **yes**.

The Copper Mountain DSLAM supports the use of RFC 1483 IP encapsulation (routed) and RFC 1483 Ethernet (bridged) encapsulation. However, the **frame cmpplay** command setting *does not* correlate to the type of encapsulation used on the DSL link, but rather to the mode used when the DSL link activates.

- In *router* mode, the protocol type in the remote is set to RFC1483MER and IP translation is turned on. MER implies that, although bridged 1483 encapsulation is used on the WAN, the router is still an IP router.
- Conversely, in *bridge* mode, bridged 1483 encapsulation is used, but the router is not acting as an IP router, but just as a pass-through device.

Remote configuredForCMPPlay

When configured using Copper Mountain Plug & Play, the router creates a remote profile and automatically sets up IP filters as shown below.

```
# rem list
INFORMATION FOR <configuredForCMPPlay>
  Status.............................. enabled
  Interface in use.................... FR
  Protocol in use..................... RFC1483 (SNAP) - MAC Encapsulated Routing
  Data Link Connection Id (DLCI)...... 528
  IP address translation.............. on
  IP filters defined.................. yes
  Send/Receive Multicast.............. off
  Block NetBIOS Packets............... off
  Source IP address/subnet mask....... 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask....... 0.0.0.0/0.0.0.0
  Management IP address/subnet mask.... 0.0.0.0/0.0.0.0
```

```
  Send IP RIP to this dest.............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
          0.0.0.0/0.0.0.0/1
  IPX network number................... 00000000
  Use IPX RIP/SAP (negotiate with PPP): yes
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
    Bridge only PPPoE with dest........ no
  mtu.................................. 1500

Begin IPFilters for configuredForCMPPlay
# watching for dropped/rejected packets is OFF

# Begin rules for input list
remote ipfilter flush  input configuredForCMPPlay
remote ipfilter insert 2 input accept -c 0 -p udp -sp 500 -da 172.17.32.132
-dp 500 (IKE Global Filter) configuredForCMPPlay
# End rules for input list

# Begin rules for receive list
remote ipfilter flush  receive configuredForCMPPlay
# End rules for receive list

# Begin rules for transmit list
remote ipfilter flush  transmit configuredForCMPPlay
remote ipfilter insert 0 transmit accept -c 0 -p udp -sa 172.17.32.132 -sp
500 -dp 500 (IKE Global Filter) configuredForCMPPlay
# End rules for transmit list

# Begin rules for output list
remote ipfilter flush  output configuredForCMPPlay
# End rules for output list

End IPFilters for configuredForCMPPlay
```

Chapter 3. Configuring the Router

Having planned your configuration and acquired the necessary information as described in chapter 2, you are ready to configure your router.

This chapter assumes that you have:

- installed the router hardware,
- connected to the router with a terminal emulation session (or ASCII terminal), and
- powered the unit on.

These tasks are described in the *User Reference Guide* that came with your router. If you intend to use the Configuration Manager, it is assumed that you have installed the Configuration Manager software and can access the terminal window. For more information, see How to Access the Command Line, on page 14.

This chapter contains:

- configuration commands for each combination of link protocol and network protocol supported by the router. (Your Network Service Provider determines the link protocol that you use.) A configuration table for the Dual-Ethernet Router (with IP routing enabled) is also provided.
- a section on verifying the router configuration describes how to test IP, IPX, and bridging.
- two sample configurations with diagrams, commands, and list outputs.

For complete, individual descriptions of the commands mentioned in this chapter, refer to Chapter 8. Command Reference on page 209.

Worksheets are provided in appendix A so that you can enter details about your local router and remote routers. The worksheets list the commands used. Fill out one worksheet for the local router and one for *each* remote router.

Note: If you are setting up both ends of the network, use a *mirror image* of the information listed below for configuring the router on the other end of the link.

Important: If you change any the of the following settings, you must save the change and then either reboot the router or restart the interface for the change to take effect:

Ethernet LAN: Ethernet IP or IPX address, TCP/IP routing, IPX routing

Bridging: Bridging, filters

Remote Router: TCP/IP route addresses, IPX routes, IPX SAPs and bridging control, and enable, disable, or add remote routers

Configuration Tables

The following tables give you step-by-step instructions for standard configurations of the following network protocol/link protocol combinations, as well as a configuration table for a dual-Ethernet router:

Link Protocol	Network Protocol	Configuration Table
PPP	IP routing	page 53
PPP	IPX routing	page 54
PPP	Bridging	page 55
RFC 1483	IP routing	page 56
RFC 1490	IP routing	page 56
RFC 1483	IPX routing	page 57
RFC 1490	IPX routing	page 57
RFC 1483	Bridging	page 58
RFC 1490	Bridging	page 58
RFC 1483MER	IP routing	page 59
RFC 1490MER	IP routing	page 59
FRF8	IP routing	page 60
Mixed network protocols		page 61
Dual-Ethernet	IP routing	page 62

Appendix A contains blank configuration worksheets for these protocol combinations.

Configuring PPP with IP Routing

This table outlines configuration commands for the PPP link protocol with the IP Routing network protocol.

PPP with IP Routing		
Steps	**Settings**	**Commands**
System Settings		
System Name	Required	**system name** <*name*>
System Message	Optional	**system msg** <*message*>
Authentication Password	Required	**system passwd** <*password*>
Ethernet IP Address	As required	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC[a] (for ATM routers)	Select: PPP Enter: VPI/VCI numbers	**remote setProtocol PPP** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[b] (for Frame Relay routers)	Select: PPP Enter: DLCI number	**remote setProtocol PPP** <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Security [c] Remote's Password	Choose security level Enter: password	**remote setAuthen** <*protocol*> <*remoteName*> **remote setOurPasswd** <*password*> <*remoteName*>
Bridging On/Off	Must be off	**remote disBridge** <*remoteName*>
TCP/IP Route Address	Enter: Explicit or default route	**remote addIproute** <*ipnet*> <*ipnetmask*> <*hop*s> <*remoteName*>
If NAT is enabled:	To enable NAT, use:	**remote setIpTranslate** on <*remoteName*>
	You may need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>
If NAT is not enabled:	You may need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	Must be enabled (optional)	**eth ip enable** **eth ip firewall** <on | off>
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using PPP in an ATM environment.
b Enter this information if you are using PPP in a Frame Relay environment.
c If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** <*remoteName*> to disable the authentication.

Configuring PPP with IPX Routing

This table outlines configuration commands for the PPP link protocol with the IPX Routing network protocol.

Note: Appendix B provides step-by-step information on how to configure IPX routing.

PPP with IPX Routing		
Steps	**Settings**	**Commands**
System Settings		
System Name	Required	**system name** <*name*>
System Message	Optional	**system msg** <*message*>
Authentication Password	Required	**system passwd** <*password*>
Ethernet IP Address	As required	**eth ip addr** <*ipaddr*> <*ipnetmask*>[<*port#*>]
Settings DHCP	Already enabled; addit. settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** < *ipaddr* >
Change Login	Optional	**system admin** <*password*>
Ethernet IPX Network #	Enter: IPX network # Frame Type (default: 802.2)	**eth ipx addr** <*ipxnet*> [<*port#*>] **eth ipx frame** <*type*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC[a] (for ATM routers)	Select: PPP Enter: VPI/VCI numbers	**remote setProtocol PPP** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[b] (for Frame Relay routers)	Select: PPP Enter: DLCI number	**remote setProtocol** PPP <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Security[c] Remote's Password	Choose security level Enter: password	**remote setAuthen** <*protocol*> <*remoteName*> **remote setPasswd** <*password*> <*remoteName*>
Bridging On/Off	Must be off	**remote disBridge** <*remoteName*>
Add IPX Routes	Enter appropriate info	**remote addIpxroute** <*ipxNet*> <*metric*> <*ticks*> <*remoteName*>
Add IPX SAPs	Enter appropriate info	**remote addIpxsap** <*servicename*> <*ipxNet*> <*ipxNode*> <*socket*> <*type*> <*hops*> <*remoteName*>
WAN Network #	Enter appropriate info	**remote setIpxaddr** <*ipxNet*> <*remoteName*>
IP and IPX Routing		
TCP/IP Routing	Must be disabled	**eth ip disable**
IPX Routing	Must be enabled	**eth ipx enable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using PPP in an ATM environment.

b Enter this information if you are using PPP in a Frame- Relay environment.

c If the ISP does not support the authentication of the ISP system by the caller, use the command:
remote disauthen <*remoteName*> to disable the authentication.

Configuring PPP with Bridging

This table outlines configuration commands for the PPP link protocol with the Bridging network protocol.

PPP with Bridging		
Steps	**Settings**	**Commands**
System Settings		
System Name	Required	**system name** <*name*>
System Message	Optional	**system msg** <*message*>
Authorization Password	Required	**system passwd** <*password*>
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** < *ipaddr* >
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC[a] (for ATM routers)	Select: PPP Enter: VPI/VCI	**remote setProtocol PPP** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[b] (for Frame Relay routers)	Select: PPP Enter: DLCI number	**remote setProtocol PPP** <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Security[c] Remote's Password	Choose security level Enter: Password	**remote setAuthen** <*protocol*> <*remoteName*> **remote setOurPasswd** <*password*> <*remoteName*>
Bridging On/Off	Must be ON	**remote enaBridge** <*remoteName*>
IP and IPX Routing		
IP Routing	Must be disabled	**eth ip disable**
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using PPP in an ATM environment.

b Enter this information if you are using PPP in a Frame-Relay environment.

c If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** <*remoteName*> to disable the authentication.

Configuring RFC 1483 / RFC 1490 with IP Routing

This table outlines configuration commands for the RFC 1483 and the RFC 1490 link protocols with the IP Routing network protocol.

RFC 1483 / RFC 1490 with IP Routing		
Steps	**Settings**	**Commands**
System Settings		
System Message	Optional	**system msg** <*message*>
Ethernet IP Address	As required	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC[a] (for ATM routers)	Select: RFC 1483 Enter: VPI/VCI Numbers	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[b] (for Frame Relay routers)	Select: FR Enter: DLCI number	**remote setProtocol FR** <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Bridging On/Off	Must be OFF	**remote disBridge** <*remoteName*>
TCP/IP Route Address	Enter: Explicit or default route with remote gateway	**remote addiproute** <*ipnet*> <*ipnetmask*> <*hops*> <*remoteName*>
If Address Translation (NAT) is enabled:	To enable NAT, use:	**remote setIpTranslate on** <*remoteName*>
TCP/IP Route Addresses	Enter: Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>
If NAT is off: TCP/IP Route Addresses	You may still need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	Must be enabled (Optional)	**eth ip enable** **eth ip firewall <on | off >**
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using RFC 1483 in an ATM environment.

b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring RFC 1483 / RFC 1490 with IPX Routing

This table outlines configuration commands for the RFC 1483 and RFC 1490 link protocols with the IPX Routing network protocol.

Note: Appendix B provides step-by-step information on how to configure IPX routing.

RFC 1483 / RFC 1490 with IPX Routing		
Steps	**Settings**	**Commands**
System Settings		
System Message	Optional	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
Ethernet IP Address	As required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** < *ipaddr* >
DHCP Settings	Already enabled; additional settings may be required	**eth ipx addr** <*ipxnet*> [<*port#*>]
Ethernet IPX Network #	Enter: IPX Network # Frame Type (default is 802.2)	**eth ipx frame** <*type*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC (for ATM routers)	Select: RFC 1483 Enter: VPI/VCI Numbers	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <*vpi number*>***** <*vci number*> <*remoteName*>
Link Protocol/DLCI[a] (for Frame Relay routers)	Select: FR Enter: DLCI number	**remote setProtocol FR** <*remoteName*> **remote setDLCI** < *number*> <*remoteName*>
Bridging on/off	Must be off	**remote disBridge** <*remoteName*>
IPX Routes Add	Enter appropriate info	**remote addIpxroute** <*ipxNet*> <*metric*> <*ticks*> <*remoteName*>
IPX SAPs Add	Enter appropriate info	**remote addIpxsap** <*servicename*> <*ipxNet*> < *ipxNode*> <*socket*> <*type*> <*hop*s> <*remoteName*>
WAN Network Number	Enter appropriate info	**remote setIpxaddr** <*ipxNet*> <*remoteName*>
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	Must be disabled (optional)	**eth ip disable** **eth ip firewall** <on | off >
IPX Routing	Must be enabled	**eth ipx enable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using RFC 1490 in a Frame Relay environment.

Configuring RFC 1483 / RFC 1490 with Bridging

This table outlines configuration commands for the RFC 1483 and RFC 1490 link protocols with the Bridging network protocol.

RFC 1483 / RFC 1490 with Bridging		
Steps	**Settings**	**Commands**
System Settings		
System Message	Optional	**system msg** <*message*>
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC (for ATM routers)	Select: RFC 1483 Enter: VPI/VCI Numbers	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[a] (for Frame Relay routers)	Select: FR Enter: DLCI number	**remote setProtocol FR** <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Bridging On/Off	Must be on	**remote enaBridge** <*remoteName*>
IP and IPX Routing		
IP Routing	Must be disabled	**eth ip disable**
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring MAC Encapsulated Routing: RFC 1483MER / RFC 1490MER with IP Routing

This table outlines configuration commands for the RFC 1483MER and RFC 1490MER link protocols with the IP Routing network protocol.

RFC 1483MER / RFC 1490 MER with IP Routing		
Steps	**Settings**	**Commands**
System Settings		
System Message	Optional	**system msg** <*message*>
Ethernet IP Address	As required	**eth ip addr** <*ipnet*> <*ipnetmask*> [<*port#*>]
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC[a] (for ATM routers)	Select: RFC 1483MER Enter: VPI/VCI Numbers	**remote setProtocol** RFC1483MER <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Link Protocol/DLCI[b] (for Frame Relay routers)	Select: MER Enter: DLCI number	**remote setProtocol MER** <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>
Bridging On/Off	Must be off	**remote disBridge** <*remoteName*>
TCP/IP Route Address	Enter: Explicit or default route with remote gateway	**remote addiproute** <*ipnet*> <*ipnetmask*><*ipGateway*> <*ipGateway*> <*remoteName*>
If NAT is enabled:	To enable NAT, use:	**remote setIpTranslate on** <*remoteName*>
If NAT is OFF:	Enter: Source WAN Port Address + mask of the remote network	**remote setSrcIpAddr** <*ipaddr*> <*mask*><*remoteName*>
TCP/IP Route Addresses	Enter a Source WAN Port Address + mask of the remote network's mask	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	Must be enabled (optional)	**eth ip enable** **eth ip firewall** <on | off >
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

a Enter this information if you are using RFC 1483 in an ATM environment.

b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring FRF8 with IP Routing

This table outlines configuration commands for the FRF8 link protocol with the IP Routing network protocol.

FRF8 with IP Routing		
Steps	**Settings**	**Commands**
System Settings		
System Message	Optional	**system msg** <*message*>
Ethernet IP Address	As required	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
DHCP Settings	Already enabled; additional settings may be required	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>
Change Login	Optional	**system admin** <*password*>
Remote Routers		
New Entry	Enter: Remote Name	**remote add** <*remoteName*>
Link Protocol/PVC	Select: FRF8 Enter: VPI/VCI Numbers	**remote setProtocol FRF8** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>
Bridging On/Off	Must be off	**remote disBridge** <*remoteName*>
TCP/IP Route Address	Enter: explicit or default route	**remote addIproute** <*ipnet*> <*ipnetmask*> <*hops*> <*remoteName*>
If Address Translation (NAT) is enabled:	To enable NAT, use:	**remote setIpTranslate on** <*remoteName*>
If NAT is OFF:	Enter: Source WAN Port Address + mask of the remote network	**remote setSrcIpAddr** <*ipaddr*> <*mask*><*remoteName*>
TCP/IP Route Addresses	Enter a Source WAN Port Address + mask of the remote network	**remote setSrcIpAddr** <*ipaddr*> <*mask*><*remoteName*>
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	Must be enabled (Optional)	**eth ip enable** **eth ip firewall** <on | off >
IPX Routing	Must be disabled	**eth ipx disable**
Store **Reboot**		**save** **reboot**

Configuring Mixed Network Protocols

Certain combinations of network protocols can be configured concurrently in the same router. The possible combinations are:

- Bridging and IP routing
- Bridging and IPX routing
- Bridging and IP routing and IPX routing
- IP routing and IPX routing

General configuration rules:

- IP (and IPX) routing takes precedence over bridging.
- Each network protocol in the combination is individually configured as described in the preceding tables.
- When configuring multiple network protocols, *make sure that they are all enabled* (even though the configuration tables show them to be mutually exclusive).

  For example, to configure bridging and IP routing (both with link protocol RFC 1483), you would refer to the tables for *RFC 1483 with Bridging* and *RFC 1483 with IP Routing*. However, you must enable *both* bridging and IP routing. IP routing will take precedence over bridging.

Configuring a Dual-Ethernet Router for IP Routing

This table outlines commands used to configure a Dual-Ethernet router for IP routing.

Dual-Ethernet Router with IP Routing		
Steps	**Settings**	**Commands**
System Settings		
System Name **System Settings**	Optional	**system name** <*name*>
Message	Optional	**system msg <***message***>**
Ethernet Settings		
Routing/ bridging controls	Enable IP routing Disable bridging	**eth ip enable** **eth br disable**
ETH/0 IP address	Define ETH/0 IP address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
ETH/1 IP address	Define ETH/1 IP address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]
TCP/IP static routes	Set routes that do not change	**eth ip addroute** <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*port#*>]
DHCP Settings Already enabled; additional settings may be required		
DHCP settings	Define DHCP network	**dhcp add** | **<***net*> <*mask*> | <*ipaddr*> | <*code*> <*min*> <*max*> <*type*>
	Create an address pool	**dhcp set addresses** <*first ipaddr*> <*last ipaddr*>
	DNS Domain Name	**dhcp set valueoption domainname <***domainname***>**
	DNS Server	**dhcp set valueoption domainnameserver** <*ipaddr*>
	WINS Server Address	**dhcp set valueoption winsserver** <*ipaddr*>

Verify the Router Configuration

Test IP Routing

Test IP Routing over the Local Ethernet LAN (from PC)

- Use the TCP/IP **ping** command or a similar method to contact the configured local router specifying the Ethernet LAN IP address. The LEDs on the router should flash for each ping received.
- If you cannot contact the router, verify that the Ethernet IP address and subnet mask are correct and check the cable connections.
- Make sure that you have saved and rebooted after setting the IP address.
- Check Network TCP/IP properties under Windows 95. If you are running Windows 3.1, check that you have a TCP/IP driver installed.

Test IP Routing to a Remote Destination

- Using the TCP/IP **ping** command, contact a remote router from a local LAN-connected PC. When you enter the **ping** command, the router will connect to the remote router using the DSL line.
- If remote or local WAN IP addresses are required, verify that they are valid.
- Use the **iproutes** command to check, first, the contents of the IP routing table and, second, that you have specified a default route as well.

Test Routing from a Remote Destination

- Have a remote router contact the local router using a similar method.

Test TCP/IP Routes

- Contact a station, subnetwork, or host located on the network beyond a remote router to verify the TCP/IP route addresses entered in the remote router database.
- Verify that you configured the correct static IP routes.
- Use the **iproutes** command to check the contents of the IP routing table.

Test Bridging to a Remote Destination

Use any application from a local LAN-attached station that accesses a server or disk using a protocol that is being bridged on the remote network beyond the remote router. If you cannot access the server:

- Verify that you have specified a default destination remote router.
- Make sure that you have enabled bridging to the remote router.
- Check that bridge filtering does not restrict access from the local station.

Test IPX Routing

One way to test IPX routing is to check for access to servers on the remote LAN. Under Windows, use the **NetWare Connections** selection provided with NetWare User Tools. Under DOS, use the command **pconsole** or type **login** on the login drive (usually F:). Select the printer server and verify that the server you have defined is listed. When you attempt to access the server, the router will connect to the remote router using the DSL line.

If you cannot access the remote server:

- Check that the local Ethernet LAN IPX network number is correct.
- Verify that the WAN link network number is the same as the remote WAN link network number.
- Check cable connections and pinouts.
- Verify that the IPX routes and IPX SAPs you have specified are correct.
- List the contents of the routing and services tables using the **ipxroutes** and **ipxsaps** commands, respectively.
- Make sure that the security authentication method and password that you configured match the remote router.

Sample Configurations

Sample Configuration 1: PPP with IP and IPX

In this configuration example of a hypothetical network, a small office/home office (SOHO) accesses:

- The Internet through an Internet Service Provider (ISP); it uses PPP as the link protocol with IP routing as the network protocol. Network Address Translation (NAT) is enabled to the ISP because the ISP assigned the SOHO only one IP address.
- A central site (HQ) through a Network Service Provider. (The NSP provides access to the DSL/ATM Wide Area Network.) It uses PPP as the link protocol with IP and IPX as its network protocols.

IP addresses are issued by the DHCP server. DHCP is set up to issue DNS information to the SOHO LAN.

Names and Passwords for Sample Configuration 1

In this configuration example, the PPP link protocol requires using system names and passwords.

- **System Passwords**

  SOHO has a system password “SOHOpasswd,” which is used when SOHO communicates with HQ for authentication by that site and at any time when HQ challenges SOHO.

  HQ has a system password “HQpasswd,” which is, likewise, used when HQ communicates with site SOHO for authentication by SOHO and at any time SOHO challenges HQ.
  ISP has a system password “ISPpasswd” used for the same purpose.

- **Remote Passwords**

  Each router has a remote router’s password for each remote router defined in its Remote Router Database. The router will use the remote password to authenticate the remote router when the remote router communicates with or is challenged by the local site.

  For example, SOHO has remote router entries for HQ and ISP; defined in each table entry is the respective remote router’s password.

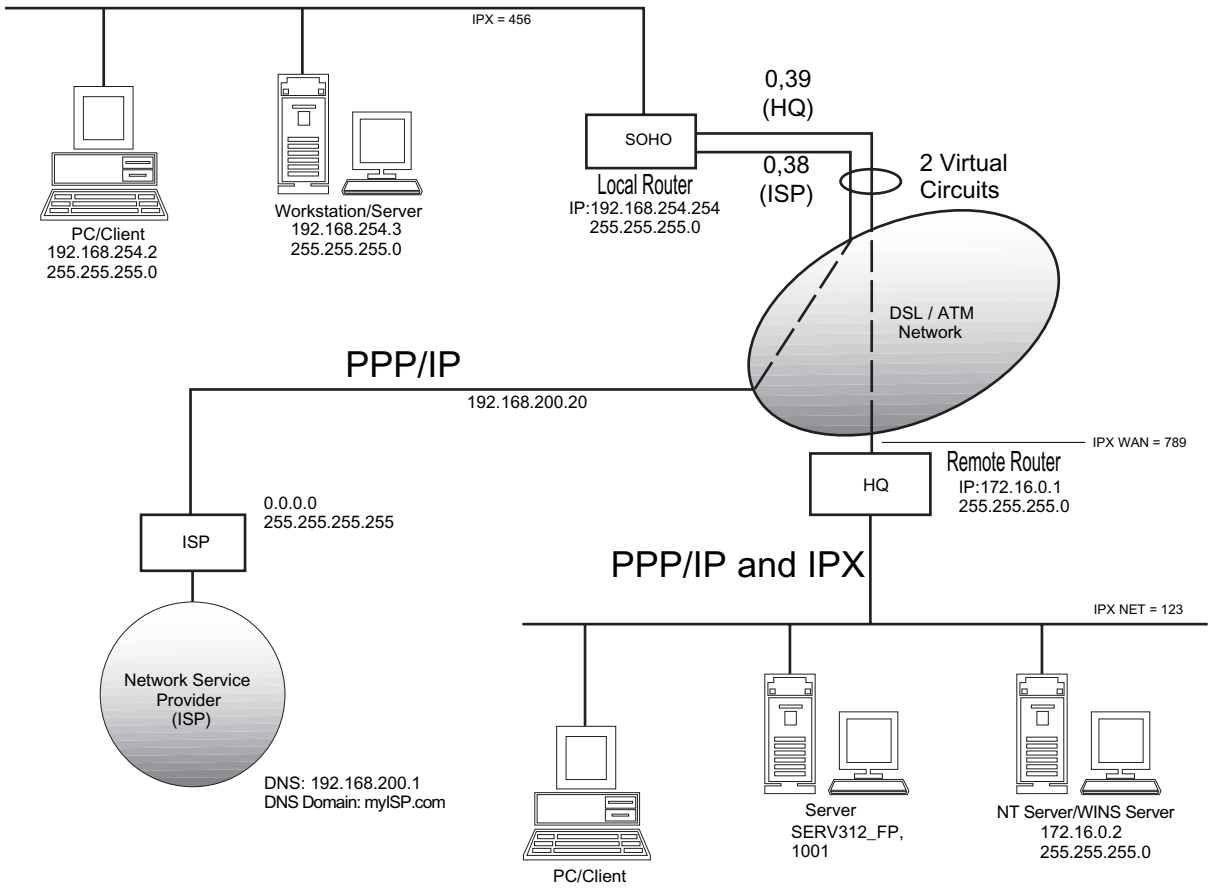
The following table shows the names and passwords for each router that must be defined for authentication to be performed correctly. (This assumes that all three systems use an authentication protocol.)

Note: If you have trouble with passwords, you can **disable authentication** to simplify the process.

	Configured in SOHO	**Configured in HQ**	**Confiigured in ISP**
System Name	SOHO	HQ	ISP
System Password	SOHOpasswd	HQpasswd	ISPpasswd
Remote Entries	HQpasswd ISPpasswd	SOHOpasswd	SOHOpasswd

Sample Configuration 1: Diagram for Local Router (SOHO)

Small Home Office SOHO (Local Router)



Network Service Provider
(HQ)

Sample Configuration 1: Tables for Local Router (SOHO)

SOHO System Settings		
Configuration Section	**Item**	**Commands**
System Settings		
System name	SOHO	**system name SOHO**
Message (optional)	Configured_Dec_1998	**system msg Configured_Dec_1998**
Authentication password	SOHOpasswd	**system password SOHOpasswd**
Ethernet IP address and subnet mask (default IP address)	192.168.254.254 255.255.255.0	**eth ip addr 192.168.254.254 255.255.255.0**
Ethernet IPX network number	456	**eth ipx addr 456**
DHCP Settings		
DNS Domain Name	myISP.com	**dhcp set valueoption domainname myISP.com**
DNS Server	192.168.200.1	**dhcp set valueoption domainnameserver 192.168.200.1**
WINS Server Address	172.16.0.2	**dhcp set valueoption winsserver 172.16.0.2**

Note: Fill in one worksheet for each remote router in the remote router database.

SOHO Remote Router Database Entry: HQ		
Configuration Section	**Item**	**Commands**
Remote router name	HQ	**remote add HQ**
Link protocol	PPP	**remote setProtocol PPP HQ**
PVC (VPI/VCI)	0*39	**remote setPVC 0*39 HQ**
Authentication	PAP (PAP is the default)	**remote setauthen PAP HQ**
Remote router's password	HQpasswd	**remote setpasswd HQpasswd HQ**
Disable bridging	(Bridging is off by default)	**remote disbridge HQ**
TCP/IP route	IP address 172.16.0.0 Subnet mask 255.255.255.0 Metric 1	**remote addiproute 172.16.0.0 255.255.255.0 1 HQ**
IPX Address	Network number 1001 Hop count 1 Ticks 4	**remote addipxroute 1001 1 4 HQ**
IPX SAPs	Server name, server type, network #, node #, sockets, type, hops	**remote addipxsap SERV312_FP 4 1001 00-00-00-00-00-01 451 3 1 HQ**
WAN Network No.	789	**remote setipxaddr 789 HQ**

SOHO Remote Router Database Entry: ISP		
Configuration Section	**Item**	**Commands**
Remote router name	ISP	**remote add ISP**
Link protocol	PPP	**remote setProtocol PPP ISP**
PVC (VPI/VCI)	0*38	**remote setPVC 0*39 ISP**
Authentication	PAP (PAP is the default)	**remote setauthen PAP ISP**
Remote router's password	ISPpasswd	**remote setpasswd ISPpasswd ISP**
Disable bridging	(Bridging is off by default)	**remote disbridge ISP**
TCP/IP route	Default route	**remote addiproute 0.0.0.0 0.0.0.0 1 ISP**
Enable Network Address Translation		**remote setiptranslate on ISP**
In Advanced: Source WAN IP Address and Subnet Mask	(Needed only if the ISP does not assign an IP address automatically.)	**remote setsrcipaddr 192.168.200.20 255.255.255 255 ISP**

SOHO Routing controls		
Configuration Section	**Item**	**Commands**
IP and IPX Routing		
Enable TCP/IP routing		**eth ip enable**
Enable IPX routing		**eth ipx enable**
Enable Internet firewall	(Firewall is on by default)	**eth ip firewall on**

Sample Configuration 1: Check the Configuration with List Commands

Type the following **commands** to list your configuration.

system list

```
GENERAL INFORMATION FOR <SOHO>
  System started on..................... 12/1/1998 at 17:41
  Authentication override.............. none
  WAN to WAN Forwarding.................yes
  Block NetBIOS Default................ no
  BOOTP/DHCP Server address............ none
  Telnet Port.......................... default (23)
  Telnet Clients....................... all
  SNMP Port............................ default (161)
  SNMP Clients......................... all
  Syslog Port.......................... default (514)
  Allowed Syslog Servers............... all
  Default Syslog Servers............... none
  System message: configured Dec-1998
  Security timer....................... 30 minutes
  One WAN Dial Up...................... no
  Backup............................... no (no valid remote profile is enabled)
    Retry Interval in Minutes.......... 30
    Stability Interval In Minutes...... 3

MODEM STRINGS:
  Reset:    ATZ
  Escape:   +++
  Init:     ATS0=0Q0V1&C1&D2&K1X4&H1&I0
  Off-Hook: ATH1
  Dial:     ATDT
  Answer:   ATA
  Hangup:   ATH0
```

remote list

```
INFORMATION FOR <HQ>
  Status............................... enabled
  Protocol in use...................... PPP
  Authentication....................... enabled
  Authentication level required........ PAP
  Connection Identifier (VPI*VCI)...... 0*39
  IP address translation............... off
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           172.16.0.0/255.255.255.0/1
  IPX network number................... 00000789
  Total IPX remote routes.............. 1
           00001001/1/4
  Total IPX SAPs....................... 1
```

```
    SERV312_FP 00001001 00:00:00:00:00:01 0451 0003 1
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes

INFORMATION FOR <ISP>
  Status.............................. enabled
  Protocol in use..................... PPP
  Authentication...................... enabled
  Authentication level required........ PAP
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation.............. on
  Compression Negotiation............. off
  Source IP address/subnet mask........ 192.168.200.20/255.255.255.255
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............ no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes.............. 1
           0.0.0.0/255.255.255.255/1
  IPX network number.................. 00000000
  Total IPX remote routes............. 0
  Total IPX SAPs...................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
```

dhcp list

```
       bootp server ................. none
       bootp file ................... n/a

       DOMAINNAMESERVER (6) ......... 192.168.200.1
       DOMAINNAME (15) .............. myISP.com
       WINSSERVER (44) .............. 172.16.0.2

Subnet 192.168.254.0, disabled - other DHCP servers detected
       When DHCP servers are active . stop
       Mask ......................... 255.255.255.0
       first ip address ............. 192.168.254.2
       last ip address .............. 192.168.254.20
       lease ........................ default
       bootp ........................ not allowed
       bootp server ................. none
       bootp file ................... n/a
```

eth list

```
ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:A1:BF
  Bridging enabled..................... no
  IP Routing enabled................... yes
    Firewall filter enabled ........... yes
    Send IP RIP to the LAN............. rip-1 compatible
      Advertise me as default router... yes
    Process IP RIP packets received.... rip-1 compatible
      Receive default route by RIP..... yes
  RIP Multicast address................ default
  IP address/subnet mask............... 192.168.254.254/255.255.255.0
  IP static default gateway............ none
  IPX Routing enabled.................. yes
```

```
External network number..... 00000456
Frame type..................... 802.2
```

Sample Configuration 2: RFC 1483 with IP and Bridging

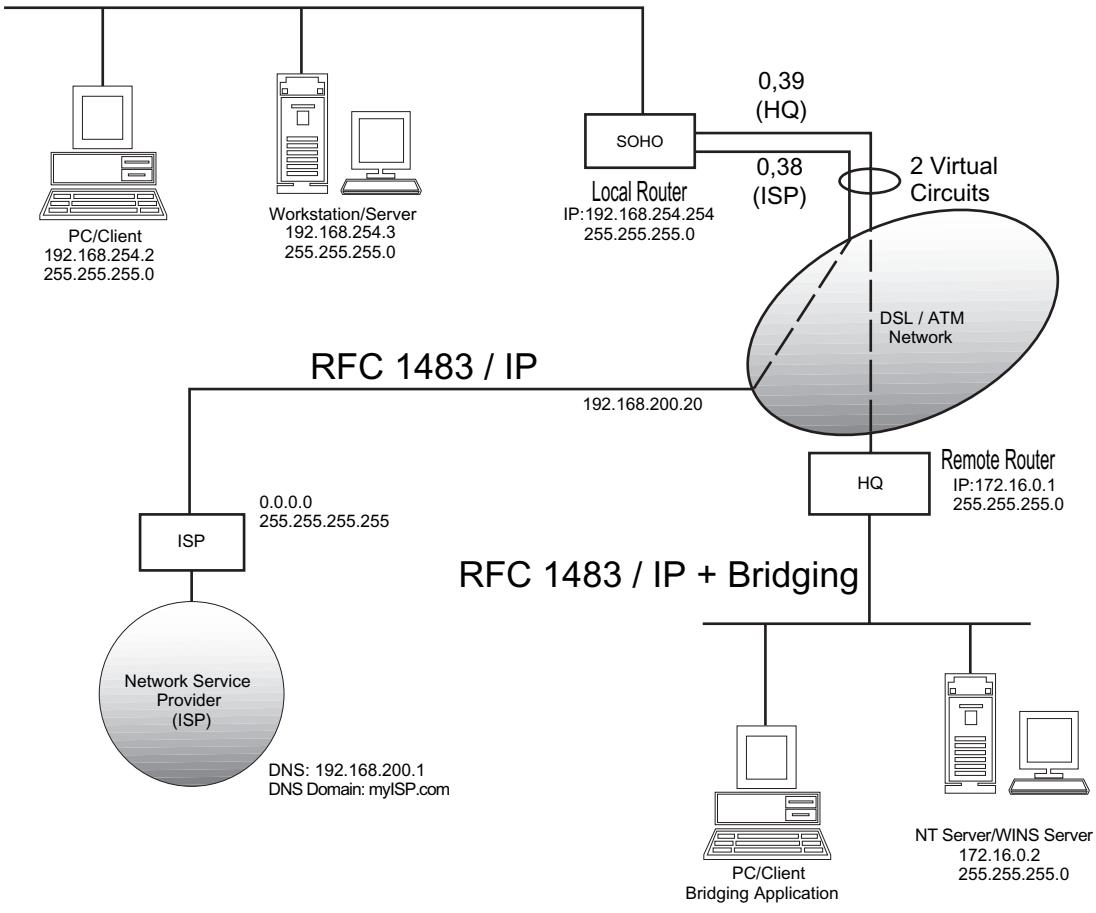
In this configuration example of a hypothetical network, a small office/home office (SOHO) will access:

- The Internet through an Internet Service Provider (ISP); it uses RFC 1483 as the link protocol with IP routing as the network protocol. Network Address Translation (NAT) is enabled to the ISP because the ISP assigned SOHO only one IP address.
- A central site (HQ) through a Network Service Provider. (The NSP provides access to the DSL/ATM Wide Area Network.) It uses RFC 1483 as the link protocol with bridging and IP routing as its network protocols. **Note:** Names and passwords are *not* required with the RFC 1483 link protocol.

IP addresses are issued by the DHCP server. DHCP will be set up to issue DNS information to the SOHO LAN.

Sample Configuration 2: Diagram for Local Router SOHO

Small Home Office SOHO (Local Router)



Network Service Provider (HQ)

Sample Configuration 2: Tables for Local Router (SOHO)

SOHO System Settings		
Configuration Section	**Item**	**Commands**
Message (optional)	RFC1483_dec98	**system msg RFC1483_dec98**
Ethernet IP address and subnet mask	(Default IP address)	**eth ip addr 192.168.254.254 255.255.255.0**
DHCP settings		
DNS domain name	myISP.com	**dhcp set valueoption domainname myISP.com**
DNS server address	192.168.200.1	**dhcp set valueoption domainnameserver 192.168.200.1**
WINS server address	172.16.0.2	**dhcp set valueoption winsserver 172.16.0.2**

SOHO Remote Router Entry: HQ		
Configuration Section	**Item**	**Commands**
Remote router name	HQ	**remote add HQ**
Link protocol	RFC1483l	**remote setProtocol RFC1483 HQ**
PVC (VPI/VCI)	0*39	**remote setPVC 0*39 HQ**
Enable bridging		**remote enabridge HQ**
TCP/IP route	IP address 172.16.0.0 Subnet 255.255.255.0 Metric 1	**remote addiproute 172.16.0.0 255.255.255.0 1 HQ**

SOHO Remote Router Entry: ISP		
Configuration Section	**Item**	**Commands**
Remote router name	ISP	**remote add ISP**
Link protocol	RFC1483	**remote setProtocol RFC1483 ISP**
PVC (VPI/VCI)	0*38	**remote setPVC 0*38 ISP**
Disable bridging	(Bridging is off by default)	**remote disbridge ISP**
TCP/IP route	Remote network's IP address, subnet mask, and metric	**remote addiproute 0.0.0.0 255.255.255.255 1 ISP**
Enable Network Address Translation		**remote setiptranslate on ISP**
	In Advanced:	
	Source WAN IP address and subnet mask	**remote setsrcipaddr 192.168.200.20 255.255.255.255 ISP**

SOHO Routing Controls		
Configuration Section	**Item**	**Commands**
IP and IPX Routing		
Enable TCP/IP routing		**eth ip enable**
Disable IPX routing	(IPX routing is off by default)	**eth ipx disable**
Enable Internet firewall	(Firewall is on by default)	**eth ip firewall on**

Sample Configuration 2: Check the Configuration with List Commands

system list

```
GENERAL INFORMATION FOR <SOHO>
  System started on..................... 12/1/1998 at 17:48
  Authentication override.............. NONE
WAN to WAN Forwarding.................. yes
  BOOTP/DHCP Server address............ none
  Telnet Port.......................... default (23)
SNMP Port.............................. default (161)
  System message: ADSL RFC1483 sample
```

eth list

```
ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:A1:BF
  Bridging enabled..................... yes
  IP Routing enabled................... yes
    Firewall filter enabled ........... yes
    Send IP RIP to the LAN............. rip-1 compatible
      Advertise me as default router... yes
    Process IP RIP packets received.... rip-1 compatible
      Receive default route by RIP..... yes
  RIP Multicast address................ default
  IP address/subnet mask............... 192.168.254.254/255.255.255.0
    IP static default gateway.......... none
  IPX Routing enabled.................. no
    External network number............ 00000000
    Frame type.......................... 802.2
```

remote list

```
INFORMATION FOR <HQ>
  Status............................... enabled
  Protocol in use...................... RFC1483 (SNAP)
  Connection Identifier (VPI*VCI)...... 0*39
  IP address translation............... off
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
          172.16.0.0/255.255.255.0/1
  IPX network number................... 00000000
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... yes
    Exchange spanning tree with dest... yes

INFORMATION FOR <ISP>
  Status............................... enabled
  Protocol in use...................... RFC1483 (SNAP)
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation............... on
```

```
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 192.168.200.20/255.255.255.255
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no

Receive IP RIP from this dest......... .no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           0.0.0.0/255.255.255.255/1
  IPX network number................... 00000000
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest.... yes
```

dhcp list

```
       bootp server ................. none
       bootp file ................... n/a

       DOMAINNAMESERVER (6) ......... 192.168.200.1
       DOMAINNAME (15) .............. myISP.com
       WINSSERVER (44) .............. 172.16.0.2

Subnet 192.168.254.0, disabled - other DHCP servers detected
       When DHCP servers are active . stop
       Mask ......................... 255.255.255.0
       first ip address ............. 192.168.254.2
       last ip address .............. 192.168.254.20
       lease ........................ default
       bootp ........................ not allowed
       bootp server ................. none
       bootp file ..................... n/a
```

Sample Configuration 3: Configuring a Dual-Ethernet Router for IP Routing

The following example provides a simple sample configuration for a Dual-Ethernet router (eth_router) with IP routing enabled.

The router's hub (ETH/0) belongs to the 192.168.254.0 subnet. The router's ETH/1 belongs to the 192.168.253.0 subnet.

ETH/0 will route packets to ETH/1 at the address 192.168.253.254. DHCP is enabled for both subnets.

eth_router Configuration		
Configuration Section	**Item**	**Commands**
System Settings		
System Name (optional)	eth_router	**system name eth_router**
Message (optional)	Configured_Jan_1999	**system msg Configured_Jan_1999**
Ethernet Settings		
Enable IP routing Disable bridging		**eth ip enable** **eth br disable**
Define ETH/0 IP address	192.168.254.254 255.255.255.0	**eth ip addr 192.168.254.254 255.255.255.0 0**
Define ETH/1 IP address	192.168.253.254 255.255.255.0	**eth ip addr 192.168.253.254 255.255.255.0 1**
TCP/IP default route	ETH/0 sends all traffic to ETH/1	**eth ip addroute 0.0.0.0 0.0.0.0 192.168.253.254 1 1**
DHCP Settings		
Define DHCP network for ETH/1	192.168.253.0 255.255.255.0	**dhcp add 192.168.253.0 255.255.255.0**
Create an address pool for ETH/1	192.168.253.2 thru 192.168.253.20	**dhcp set addresses 192.168.253.2 192.168.253.20**
DNS domain name	myISP.com	**dhcp set valueoption domainname myISP.com**
DNS server	192.168.200.1	**dhcp set valueoption domainnameserver 192.168.200.1**
WINS server address	172.16.0.2	**dhcp set valueoption winsserver 172.16.0.2**

Chapter 4. Configuring Special Features

The features described in this chapter are advanced topics. They are primarily intended for experienced users and network administrators to perform network management and more complex configurations.

- IP subnets
- Virtual routing tables
- Bridge filtering and IP firewall
- RIP (IP protocol controls)
- DHCP (Dynamic Host Configuration Protocol)
- NAT (Network Address Translation)
- PPPoE (PPP over Ethernet)
- Management security
- Dial backup to a asynchronous modem
- VRRP backup for a static default gateway

Additional features can be purchased as software option keys. These features are described in Configuring Software Options, page 124. To determine which software options are installed on your router, use the **vers** command. (If a feature has not been enabled, it is listed with a ~ prefix.)

IP Subnets

You may configure the router to provide access to multiple IP subnets on the Ethernet network. (This feature does not apply to IPX or bridged traffic.)

Each IP subnet is referenced as a logical (or virtual) Ethernet interface. You may define multiple logical interfaces for each physical Ethernet interface (that is, port) in the router. Each logical interface is referenced by its port number and logical interface number (*port #:logical#).*

Logical Interface Commands

To define a logical interface, first use the **eth add** command (page 262); it specifies the port number and the new logical interface number. You then enter an **eth ip addr** command (page 264) to define the IP address and subnet mask of the IP subnet.

The default logical interface for each port is interface 0; this logical interface 0 always exists and cannot be deleted. (Other logical interfaces may be deleted using the **eth delete** command [page 263].)

Stopping and Starting an Interface

You can stop and start a logical interface without rebooting the entire router. To do so, use these commands:

- **eth stop** Stops a logical Ethernet interface (page 282).
- **eth start** Starts a logical Ethernet interface (page 282).
- **eth restart** Stops and restarts a logical Ethernet interface (page 281).

Note: When you stop or restart an interface, interface changes are discarded if they have not been saved.

Interface Routing and Filtering

After the **eth add** and **eth ip addr** commands define the Ethernet logical interface, other **eth** commands can reference it, including:

- **eth ip addRoute** Adds an Ethernet IP route that uses the logical Ethernet interface. The route is added to the default routing table (page 264).
- **eth ip bindRoute** Adds an Ethernet IP route that uses the logical Ethernet interface. The route is added to a virtual routing table (page 266).
- **eth ip filter** Manages IP filters for the logical Ethernet interface. Lists of input, output, and forward filters may be defined for the interface (page 270).
- **eth ip options** Sets RIP options for the logical interface; these options set IP routing information protocol controls (page 276).

Note: In general, logical interface commands are not effective until you **save** the change and either **restart** the logical interface or **reboot** the router. However, the **eth ip bindRoute** and **eth ip filter** commands are effective immediately if the logical Ethernet interface is already active.

Virtual Routing Tables

The virtual routing feature allows you to define multiple routing tables. This is also known as IP virtual router support.

To define a new routing table, you must specify a name for the routing table and a range of IP source addresses that use that table. The router determines which routing table to use based on the source address in the packet. For example, if the router receives a packet whose source address is 192.168.254.10, it checks if that address is within the address range defined for a virtual routing table. If it is, the virtual routing table is used to route the packet. If it is not, the default routing table is used instead.

The address ranges assigned to the virtual routing tables may not overlap. All source IP addresses not assigned to a virtual routing table are routed using the default routing table. You can add routes to the default routing table using **eth ip addroute** (page 264) and **remote addiproute** (page 291).

The following commands define the names and address ranges of the virtual routing tables:

system addIPRoutingTable Adds a range of IP addresses to a virtual routing table. The virtual routing table is defined if it does not already exist (page 232).

system delIPRoutingTable Deletes a range of IP addresses from the range defined for a virtual routing table or deletes the entire table (page 243).

system moveIPRoutingTable Moves a range of IP addresses from their current assignment to the specified virtual routing table. The virtual routing table is defined if it does not already exist (page 250).

To add and remove routes from a virtual routing table, use these commands:

eth ip bindRoute Adds an Ethernet route to a virtual routing table (page 266).

eth ip unbindRoute Removes an Ethernet route from a virtual routing table (page 277).

remote bindIPVirtualRoute Adds a remote route to a virtual routing table (page 293).

remote unbindIPVirtualRoute Removes a remote route from a virtual routing table (page 324).

Note: Unlike changes to the default routing table, changes to IP virtual routing tables take effect immediately. However, the changes are lost if they are not saved before the next **reboot**.

Bridge Filtering and IP Firewall

You can control the flow of packets across the router using bridge filtering. Bridge filtering lets you “deny” or “allow” packets to cross the network based on position and hexadecimal content within the packet. This enables you to restrict or forward messages with a specified address, protocol, or data content. Common uses are to prevent access to remote networks, control unauthorized access to the local network, and limit unnecessary traffic.

For example, it might be necessary to restrict remote access for specific users on the local network. In this case, bridging filters are defined using the local MAC address for each user to be restricted. Each bridging filter is specified as a “deny” filter based on the MAC address and position of the address within the packet. To initiate bridge filtering, “deny” filtering mode is then enabled. Every packet with one of the MAC addresses would not be bridged across the router until “deny” filtering mode was disabled.

Similarly, protocol filtering can be used to prevent a specific protocol from being bridged. In this case, the protocol id field in a packet is used to deny or allow a packet. You can also restrict, for example, the bridging of specific broadcast packets.

Configure Bridge Filtering

Bridge filtering allows you to control the packets transferred across the router. This feature can be used to enhance security or improve performance. The filtering is based on matched patterns within the packet at a specified offset. Two filtering modes are available:

- “Deny” mode will discard any packet matched to the “deny” filters in the filter database and let all other packets pass.
- “Allow” mode will only pass the packets that match the “allow” filters in the filter database and discard all others.

Up to 40 “allow” filters or 40 “deny” filters can be activated from the filter database.

Enter the filters, including the pattern, offset, and filter mode, into a filter database. If you intend to restrict specific stations or subnetworks from bridging, then add the filters with a “deny” designation and then enable “deny” filtering. If you wish to allow only specific stations or subnetworks to bridge, then add the filters with an “allow” designation and enable “allow” filtering. Add each filter with the following command:

filter br add [*pos*] [*data*] [deny | allow]

where [*pos*] is the byte offset within a packet (number from 0-127) to a [*data*] (a hex number up to 6 bytes). This data and offset number can be used to identify an address, a protocol id, or data content. After entering your filters, verify your entries with the following command:

filter br list

If you have entered an incorrect filter, delete the filter using the **filter br del** command. When you are satisfied with the filter list, save the filtering database with the **save filter** command. You must reboot the router to load the filtering database. Then enable bridging filtering with the following command:

filter br use [none | deny | allow]

To test the filtering configuration, access the remote destination identified in the filter.

Internet Firewall Filtering

The router supports IP Internet Firewall Filtering to prevent unauthorized access to your system and network resources from the Internet. This filter discards packets received from the WAN that have a source IP address recognized as a local LAN address.

Caution: This is a simple firewall check; it does not add much security. For more elaborate firewall features, see IP Filtering, page 129 .

Initially, the Internet Firewall defaults to **on**, but it is active *only* when Ethernet LAN IP routing is on.Thus, at initial configuration, to activate the Internet Firewall Filter, you need only enable IP routing.

Ethernet LAN IP routing is controlled by the commands:

eth ip enable
eth ip disable

To prevent IP Internet Firewall Filtering while IP routing is enabled, turn off the firewall filter. To turn off the firewall filter, use the command:

eth ip firewall off

To turn on the firewall filter again, use the command:

eth ip firewall on

Note: *Remember to save and reboot* if you alter the IP routing status.

IP Directed Broadcast Filtering

Initially, when you enable IP routing, a filter is also enabled that prevents the forwarding of broadcast packets directed to a specific network prefix. Using this filter, the router silently discards all packets broadcast to a subnet. This applies to all broadcast interfaces, including all Ethernet interfaces.

A network prefix-directed broadcast address is the broadcast address to a particular network. For example, if a network's IP address is 192.168.254.254 and its mask is 255.255.255.0, its network prefix-directed broadcast addresses are 192.168.254.0 and 192.168.254.255.

This feature is independent of the IP firewall and IP filtering features. However, it does require that IP routing be enabled (see **eth ip enable**, page 270). To see the current settings for IP routing and directed broadcasts, use the command **eth list**, page 280.

To *allow* the forwarding of network-prefix-directed broadcast packets, use the following command:

eth ip directedBcast on

To, once again, *prevent* the forwarding of network-prefix-directed broadcast packets, use the command:

eth ip directedBcast off

RIP Controls

The RIP control options allows you to decide what routing information you want to receive and what routing information you choose to share on the network.

For a remote interface, the default is to *not* send or receive IP RIP packets. If you choose to use this default, you *must* use the **remote addiproute** command (page 291) to configure static routes for this WAN link.

You can configure the router to send and receive RIP packet information, respectively, to and from the remote router. This means that the local site will "learn" all about the routes beyond the remote router and the remote router will "learn" all about the local site's routes. You may not want this to occur in some cases. For example, if you are connecting to a site outside your company, such as the Internet, you may want to keep knowledge about your local site's routes private.

To see the current settings for a remote interface, use the command **remote list** and check the output lines:

```
Send IP RIP to this dest.............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
```

For an Ethernet interface, the default is to:

- receive and process IP RIP-1 compatible and RIP-2 broadcast packets from the Ethernet LAN.
- receive and process RIP-2 packets that are multicast as defined by the **eth ip ripmulticast** command.
- transmit RIP-1 compatible broadcast packets and RIP-2 multicast packets over the Ethernet LAN.

To see the current settings for an Ethernet interface, use the command **eth list** and check the output lines:

```
Send IP RIP to the LAN............... rip-1 compatible
    Advertise me as default router..... yes
  Process IP RIP packets received...... rip-1 compatible
    Receive default route by RIP....... yes
```

To set or clear RIP options for a remote interface or an Ethernet interface, use these commands:

remote setipoptions <*option*> **on** | **off** <*remoteName*>

eth ip options <*option*> **on** | **off** <*interface*>

The available RIP options on these commands are:

rxrip	Receive IP RIP packets
txrip	Send IP RIP packets
rxrip1	Receive and process RIP-1 packets only
txrip1	Send RIP-1 packets only
rxrip2	Receive and process RIP-2 packets only
txrip2	Send RIP-2 packets only
rxdef	Receive the default route

txdef Advertise this router as the default router
avdfr

Advertising the Local Site

The default is to keep the local site's existence private. Unless specified otherwise, the remote does not advertise its route to other sites. This security mechanism is useful when the remote connects to a site outside your company (an Internet Service Provider, for example), or whenever you want to keep the identity of the site private.

To see the current setting, enter the command **remote list** and check the output line:

```
Keep this IP destination private..... yes
```

To turn off this security mechanism, use this command:

remote setipoptions private off *<remoteName>*

Changing the Multicast Address for RIP-2 Packets

The default multicast address for RIP-2 packets sent and/or received is 224.0.0.9. If necessary, you can change this address with the command **eth ip ripmulicast** (page 277).

To see the current setting, enter the command **eth list** and check the output line:

```
RIP Multicast address.............. default
```

Multicast Forwarding Controls

The forwarding of multicast packets by an interface depends on the setting of the multicast IP option for that interface. To turn on multicast forwarding for a remote interface, use the command:

remote setipoptions multicast on *<remoteName>*

If *any* remote interface has multicast forwarding enabled, then multicast forwarding is automatically enabled on *all* Ethernet interfaces. However, multicast forwarding can be turned off or turned on for an Ethernet interface using the command:

eth ip options multicast on | off *<interface>*

To see the current setting, use the command **eth list** and check the output line:

```
Multicast forwarding enabled....... no
```

DHCP (Dynamic Host Configuration Protocol)

The router supports DHCP and can act as the DHCP server. (The router's DHCP server disables itself if it locates other active DHCP servers on the network or if a DHCP server on the WAN has been explicitly specified.)

When configured, the router can provide DHCP functions as follows.

- As a *server*, IP addresses are assigned to workstations attached to the LAN that issue DHCP address requests.
- As a *client*, the router requests that an IP address be assigned to the WAN side port of the router.
- As a *relay*, the router passes through client requests from the LAN side onto the WAN asking for IP address assignment and relays responses back to the appropriate client.

This section describes how to configure DHCP using the Command Line Interface. Configuring DHCP can be a complex process; this section is therefore intended for network managers. For a complete list and explanation of the DHCP commands, see DHCP Commands, page 350.

Note: Some DHCP values can be set using the Windows Quick Start application, the Windows Configuration Manager, or the web-based EZ Setup application.

DHCP Address Allocation

DHCP is a service that allocates IP addresses *automatically* to any DHCP client requesting an IP address. (A DHCP client can be any device attached to your network, for example, a PC.) It can also provide option values (such as the subnet mask, DNS, and gateway values) automatically.

Using DHCP to automatically acquire initialization parameters translates into avoiding the more involved router/PC manual initialization process. (The manual initialization requires reconfiguration of router and/or PC addresses to be in the same network.)

To configure DHCP for a network, the network administrator defines a range of valid IP addresses to be used in the subnetwork as well as options and other parameters. This process is described in the next section, *DHCP Administration and Configuration*.

Note 1: DHCP is effective only if the TCP/IP stack is installed on the PCs.

Note 2: In Windows, DHCP is enabled by selecting it on your PC (under **Settings, Control Panel, Network,** and **TCP/IP** on the **Configuration** tab page).

DHCP Client Requests

Before becoming active, the router's DHCP server attempts to locate other active DHCP servers on the network, such as Windows NT servers. If one is detected, the router's DHCP server disables itself.

When the WAN link activates and the source IP address or mask is undefined (i.e. 0.0.0.0), the router places a DHCP client request over the WAN link. The router may learn the following parameters:

- DNS address
- Default gateway
- Syslog server IP address

- Time server IP address
- Source IP address to use

To see the gateway and source IP addresses that were returned, use the **iproutes** command.

The IP addresses and options assigned to a client are collectively called the “lease”. The lease is only valid for a certain period of time and is automatically renewed by the client.

DHCP Administration and Configuration

The DHCP administration and configuration process is divided into the following parts:

- Manipulating subnetworks and explicit client leases
- Setting option values
- Managing BootP
- Defining option types
- Configuring BootP/DHCP relays
- Other information

Note: To save the DHCP configuration or changes to flash memory in the router, remember to use the command **dhcp save**.

Manipulating Subnetworks and Explicit Client Leases

Enabling/Disabling a Subnetwork or a Client Lease

To enable/disable a subnetwork or a client lease, use the commands:

dhcp enable all | <*net*> <*ipaddr*>
dhcp disable all | <*net*> <*ipaddr*>

`Examples:`

To enable the subnetwork 192.168.254.0 if that subnetwork exists, enter:

```
dhcp enable 192.168.254.0
```

To enable the client lease 192.168.254.17 if that client lease exists, enter:

```
dhcp enable 192.168.254.17
```

To disable the client lease 192.168.254.18 if that client lease exists, enter:

```
dhcp disable 192.168.254.18
```

To check the results of these commands, use: **dhcp list**

If the client lease does not exist, it must be explicitly created.

Adding Subnetworks and Client Leases

• Adding a Subnetwork

The following commands are used to add/delete subnetworks. Only *one* subnetwork with *one* pool of IP addresses may be defined for a subnet.

To add a subnetwork, use:

dhcp add <*net*> <*mask*>

To remove a subnetwork, use:

dhcp del <*net*>

Note: All client leases associated with this subnetwork are automatically deleted.

Example 1:

The following command creates a subnetwork 192.168.254.0 with a subnet mask of 255.255.255.0:
`dhcp add 192.168.254.0 255.255.255.0`

Example 2:

The following command deletes the subnetwork 192.168.254.0 *and* deletes *all* client leases associated with that subnetwork:
`dhcp del 192.168.254.0`

• Adding Explicit or Dynamic Client Leases

Client leases may either be created dynamically or explicitly. Usually client leases are created dynamically when PCs boot and ask for IP addresses.

Explicit client leases

To add an explicit client lease, a subnetwork *must* already exist (use **dhcp add** <*net*> <*mask*> to add the subnetwork) before the client lease may be added. Use the command:
dhcp add <*ipaddr*>

To remove a client lease, use:
dhcp del <*ipaddr*>

Note: An administrator *may* create a client lease that is part of a subnet but does not fall within the pool of IP addresses.

Example 1:

To explicitly add the client lease 192.168.254.31, type:
`dhcp add 192.168.254.31`

Example 2:

To delete the client lease 192.168.254.31, type:
d**`hcp del 192.168.254.31`**

Dynamic Client Leases

Dynamic client leases are created from the pool of IP addresses associated with that subnetwork.

To set or change the pool, use:
dhcp set addresses <*first ip addr*> <*last ip addr*>

To clear the values from the pool, use:
dhcp clear addresses <*net*>

Note: Any client leases that currently exist will not be affected.

To remove a client lease that was dynamically created, use:
dhcp del <*ipaddr*>

Caution: If <*ipaddr*> is a subnet, you will delete the entire subnet.

Setting the Lease Time

• Concepts

The information given by the DHCP server (router) to your PC is leased for a specific amount of time. The client lease has already been selected. The DHCP server will select the lease time based on the option defined for the client lease as described by this algorithm:

1. If the client lease option is a specific number or is infinite, then the server uses the specified lease time associated with this client lease.

2. If the client lease option is “default”, then the server goes up one level (to the subnetwork) and uses the lease time explicitly specified for the subnetwork.

3. If the client *and* subnetwork lease options are both “default”, then the server goes up one level (global) and uses the lease time defined at the global level (server).

4. Lease time:
   The minimum lease time is 1 hour.
   The global default is 168 hours.

• Commands

The following commands are used by network administrators to control lease time.

To set the lease time explicitly for the client lease, use:
dhcp set lease <*ipaddr*> <*hours*>

To set the lease time explicitly for the subnetwork lease, use:
dhcp set lease <*net*> <*hours*>

To set the lease time explicitly for the global lease, use:
dhcp set lease <*hours*>

Example 1:

To set the lease time to “default” for the client 192.168.254.17, type:
```
dhcp set lease 192.168.254.17 default
```

Example 2:

To set the subnetwork lease time to infinite for the subnet 192.168.254.0, type:
```
dhcp set lease 192.168.254.0 infinite
```

Example 3:

To set the global lease time to 2 hours, type:
`dhcp set lease 2`

Manually Changing Client Leases

In general, administrators do not need to change client leases manually. However, if the need arises to do so, the following two commands are used.

Warning: The client will not be aware that the administrator has changed or released a client lease!

To change the client lease expiration time to a given value:
dhcp set expire <*ipaddr*> <*hours*>

Setting the expiration time to "default" will cause the server to compute the lease time using the algorithm as described in Setting the Lease Time, page 88.

To release the client lease so it becomes available for other assignments:
dhcp clear expire <*ipaddr*>

Setting Option Values

Administrators can set values for global options, for options specific to a subnetwork, or for options specific to a client lease.

Note: See RFC 2131/2132 for the description of various options.

Concepts

The server returns values for options explicitly requested in the client request. It selects the values to return based on the following algorithm:

1. If the value is defined for the client, then the server returns the requested value for an option.
2. If the value for the option has not been set for the client, then the server returns the value option if it has been defined for the subnetwork.
3. If the value option does not exist for the client *and* does not exist for the subnetwork, then the server returns the value option if it has been defined globally.
4. If the value option is not defined anywhere, the server does *not* return any value for that option in its reply to the client request.

Important: When the server replies to a client:

- *It does not* return any option values *not* requested by the client.
- *It does not* support the definition of a "class" of clients.
- *It does not* return any non-default option values *unless* the client requests the option value *and* the server has a value defined for that option.
- *It does not* return any non-default values on the clients subnet *unless* the client requests the value for that option.

Commands for Global Option Values

To set the value for a global option, use:
dhcp set valueoption <*code*> <*value*>**...**

The code can be a number between 1 and 61 or a keyword.

To see the list of predefined and user-defined options, use:
dhcp list definedoptions

To clear the value for a global option, use:
dhcp clear valueoption <*code*>

Example:

To set the global value for the domain name server option, enter:

```
dhcp set valueoption domainnameserver 192.168.254.2 192.168.254.3
```

Commands for Specific Option Values for a Subnetwork

To set the value for an option associated with a subnetwork, use:
dhcp set valueoption <*net*> <*code*> <*value*>**...**

To clear the value for an option associated with a subnetwork, use:
dhcp clear valueoption <*net*> <*code*>

Examples:

```
dhcp set valueoption 192.168.254.0 gateway 192.168.254.254
dhcp set valueoption 6 192.84.210.75 192.84.210.68
```

Commands for Specific Option Values for a Client Lease

To set the value for an option associated with a specific client, use:
dhcp set valueoption <*ipaddr*> <*code*> <*value*>**...**

To clear the value for an option associated with a specific client, use:
dhcp clear valueoption <*ipaddr*> <*code*>

Example:

```
dhcp set valueoption 192.168.254.251 winserver 192.168.254.7
```

Commands for Listing and Checking Option Values

To list the values for global options as well as subnet and client lease information, use:
dhcp list

To list options that are set for that subnet/client lease as well as subnet/client lease information, use:
dhcp list <*net*>|<*ipaddr*>

This command lists all available options (predefined and user-defined options):
dhcp list definedoptions

This command lists all available options starting with the string “name”.
dhcp list definedoptions name

To list the lease time use:
dhcp list lease

Example:

This command lists the subnet 192.168.254.0 including any options set specifically for that subnet:

```
dhcp list 192.168.254.0
```

Managing BootP

Administrators can enable and disable BootP and specify the BootP server. BootP can be enabled at the subnetwork and at the client lease level.

Note: By default, the DHCP server does *not* satisfy BootP requests unless the administrator has explicitly enabled BootP (at the subnetwork or lease level).

About BootP and DHCP

BootP and DHCP provide services that are very similar. However, as an older service, BootP offers only a subset of the services provided by DHCP.

The main difference between BootP and DHCP is that the client lease expiration for a BootP client is always *infinite*.

Note: Remember, when BootP is enabled, the client assumes that the lease is infinite.

Enable/Disable BootP

To allow BootP request processing for a particular client/subnet, use the command:
dhcp bootp allow <*net*>|<*ipaddr*>

To disallow BootP request processing for a particular client/subnet, type:
dhcp bootp disallow <*net*>|<*ipaddr*>

Specify the Boot (TFTP) Server

The following commands let the administrator specify the TFTP server (boot server) and boot file name. The administrator should first configure the IP address of the TFTP server and file name (kernel) from which to boot.

To set the IP address of the server and the file to boot from, use the commands:
dhcp bootp tftpserver [<*net*>|<*ipaddr*>] <*tftpserver ipaddr*>
dhcp bootp file [<*net*>|<*ipaddr*>] <*file name*>

To clear the IP address of the server and the file to boot from, use:
dhcp bootp tftpserver [<net>|<*ipaddr*>] 0.0.0.0

Example 1:

To set the global BootP server IP address to 192.168.254.7:

```
dhcp bootp tftpserver 192.168.254.7
```

Example 2:

To set the subnet 192.168.254.0 server IP address to 192.168.254.8:

```
dhcp bootp tftpserver 192.168.254.0 192.168.254.8
```

Example 3:

To set the client 192.168.254.21 server IP address to 192.168.254.9

```
dhcp bootp tftpserver 192.168.254.21 192.168.254.9
```

Example 4:

To set the subnet 192.168.254.0 boot file to “kernel.100”:
`dhcp bootp file 192.168.254.0 kernel.100`

Example 5:

To clear the global BootP server IP address and file name:
`dhcp bootp tftpserver 0.0.0.0`

Example 6:

To clear the subnet 192.168.254.0 server IP address and file name:
`dhcp bootp tftpserver 192.168.254.0 0.0.0.0`

Configuring BootP/DHCP Relays

BootP/DHCP relays are used by system administrators when the DHCP configuration parameters are acquired from a BootP/DHCP server other than the router’s DHCP server.

This feature allows configuration information to be centrally controlled. Enabling a BootP/DHCP relay disables DHCP on the router because, by definition, only one policy mechanism can be supported.

However, multiple relays may be specified. BootP/DHCP requests are forwarded to every relay on the list. It is assumed, in this case, that the multiple servers are configured to recognize the requests that they are to handle.

To add a BootP/DHCP Relay address to the list, use the command:

dhcp addrelay <*ipaddr*>

To remove a BootP/DHCP Relay address from the list, use the command:

dhcp delrelay <*ipaddr*>

Defining Option Types

Concepts

A DHCP option is a code, length, or value. An option also has a “type” (byte, word, long, longint, binary, IP address, string).

The subnet mask, router gateway, domain name, domain name servers, NetBios name servers are all DHCP options. Refer to RFC 1533 if you require more information.

Usually users will *not* need to define their own option types. The list of predefined option types based on RFC 1533 can be shown by typing **dhcp list definedoptions.**

Commands

The following commands are available for adding/deleting option types:
dhcp add <*code*> <*min*> <*max*> <*type*>

To list option types that are currently defined, use:
dhcp list definedoptions...

To list the definitions for all known options, use:
dhcp list definedoptions

To get help information, use:
dhcp list definedoptions?

To list the definition for option 1, if option 1 is defined, type:|
`dhcp list definedoptions 1`

To list the definition for all options that are well-known AND have a name starting with “h”, type:
`dhcp list definedoptions h`

Example:

To define a new option with a code of 128, a minimum number of IP addresses of 1, a maximum number of IP addresses of 4, of type “IP address”, type:
`dhcp add 128 1 4 ipAddress`

This information implies that:

- Some DHCP client will know about the option with code 128.
- Option 128 allows IP addresses.
- The server can have a minimum of 1 IP address.
- The server can have up to 4 IP addresses.
- The administrator will still need to set the option value either globally, specific to a subnetwork, or specific to a client for the option to have any meaning.

To delete the definition of the option with code 128, type:

`dhcp del 128`

The values for this option that have been set globally, specific to a subnetwork, or specific to a client will *not* be removed. The administrator must remove those values explicitly. Well-known type option codes *cannot* be changed or deleted.

DHCP Information File

DHCP information is kept in the file DHCP.DAT, a self-contained file.

This file contains *all* DHCP information including:

- the option definitions
- the subnetworks that have been added
- the client lease information
- the option values that have been set

This file can be uploaded/downloaded from one router to another.

Clearing All DHCP Information

If necessary, you can clear all DHCP information from memory, including all leases and all global DHCP information. To do so, enter this command:

dhcp clear all records

At this point, the DHCP information is cleared from memory, but the DHCP.DAT file remains unchanged. To clear the information from the DHCP.DAT file as well, enter:

save

Note: You cannot abbreviate the word **records** in the **dhcp clear all records** command.

Network Address Translation (NAT)

Network Address Translation (NAT) allows devices on the LAN to use private IP addresses that aren't recognized on the Internet. The router supports both of the following NAT techniques:

Classic NAT One NAT IP address is assigned to one PC IP address (see page 99).

Masquerading One NAT IP address is assigned to many PC IP addresses (see page 95).

Note: Some applications that use IP or UDP protocols may have problems with Network Address Translation. You may be able to avoid this problem by running in TCP mode or by disabling NAT and running as a subnetwork to your ISP.

Supported applications include AOL chat, CUSeeMe, Doom, FTP, L2TP, HTTP, Kali Netbios over IP, NetMeeting, PCanywhere, Quake, Quicktime Video, Real Audio, RTSP, SGI Media Base, SMTP, StreamWorks, Telnet, TFTP, Unix commands (finger, rcp, rshell, rlogin, whois) and VDO. To read more about H.323 with NAT, see NetMeeting (H.323) with NAT, page 100.

General NAT Rules

- IP routing must be enabled (see **eth ip enable**, page 270).
- NAT can be run globally or on a per-remote-router and per-Ethernet-interface basis.
- Any number of PCs on the LAN may be going to the same or different remote routers at the same time. In reality, the number of PCs on the LAN that can be supported is limited by how much memory the router consumes maintaining table information *and* by how many connections are currently active.
- Some operations will *not* work. Specifically, services that place IP address/port information in the data *may not work* until the router examines their packets and figures out what information in the data needs to be changed. Remember that the router is remapping both IP addresses and ports.
- When using NAT with a remote router, either the remote ISP *must* supply the IP address for NAT translation or the user *must* configure the IP address for NAT translation locally.
- Any number of PCs on the LAN may have a connection to the same or different remote routers at the same time. In reality, the number of PCs on the LAN that can be supported is limited by the amount of memory consumed by the router to maintain table information *and* by the number of connections the router "thinks" are currently active. Theoretically, up to 64,000 active connections per protocol type—TCP/UDP—can be concurrently running, if the table space is available.

Masquerading

With masquerading, multiple local (PC) IP addresses are mapped to a single global IP address. Many local (PCs) IP addresses are therefore hidden behind a single global IP address. The advantage of this type of NAT is that users only need one global IP address, but the entire local LAN can still access the Internet. This NAT technique requires not only remapping IP addresses but also TCP and UDP ports.

Each PC on the LAN side has an IP address and a mask. When the router connects to an ISP, the router appears to be a "host" with one IP address and mask. The IP address that the router uses to communicate with the ISP is obtained dynamically (with PPP/IPCP or DHCP) or is statically configured. When the PC connects to the ISP, the IP address and port used by the PC are remapped to the IP address assigned to the router. This remapping is done dynamically.

Client Configuration

- **Enable NAT**

  To enable NAT for a remote interface, use the commands:

  remote setIpTranslate on <*remoteName*>
  save

  To enable NAT for an Ethernet interface, use the commands:

  eth ip translate on <*interface*>
  save

  The **save** command makes the above changes persistent across reboots; these changes turn NAT on when the specified interface is used.

- **Obtain an IP Address for NAT**

  The IP address (the IP address "known" by the remote ISP) used for this type of NAT can be assigned in two ways.

  The ISP dynamically assigns the IP address. Use the commands:

  remote setSrcIpAddr 0.0.0.0 0.0.0.0 <*remoteName*>
  save

  The IP address is assigned locally. Use the commands:

  remote setSrcIpAddr *ww.xx.yy.zz* 255.255.255.255 <*remoteName*>
  save

  Note: *ww.xx.yy.zz* is the IP address that the user on the local LAN assigns.

Server Configuration

This section is intended for users and network administrators who wish to allow WAN access to a Web server, FTP server, SMTP server, etc., on their local LAN, while using NAT.

NAT needs a way to identify which local PC [local IP address(es)] should receive these server requests. The servers can be configured on a *per-remote-router* and *per-Ethernet-interface basis* as well as *globally.*

- **Interface-Specific Commands**

  You can specify servers for specific remote interfaces and for specific Ethernet interfaces. Servers can also be designated for specific protocols and ports. To enable and disable a local IP address (on your LAN) as the server for a specific remote interface, use these commands:

  remote addServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]] <*remoteName*>

  remote delServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]] <*remoteName*>

  See the command descriptions on page 293 and page 298. To see all of the remote entries, use the command **remote list** <*remoteName*>

  To enable and disable a local IP address (on your LAN) as the server for a specific Ethernet interface, use these commands:

eth ip addServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]] <*interface*>

eth ip delServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]] <*interface*>

See the command descriptions on page 265 and page 268.

Remember to type **save** to make the changes persistent across reboots.

Example 1:

Assume that the local LAN network is 192.168.1.0 255.255.255.0. The following commands enable a Telnet server on the local LAN with the IP address 192.168.1.3, and an FTP server with the IP address 192.168.1.2.

```
remote addServer 192.168.1.3 tcp telnet router1
remote addServer 192.168.1.2 tcp ftp router1
```

When the local router receives a request from *router1* to communicate with the local Telnet server, the local router sends the request to 192.168.1.3. If *router1* asks to talk to the local FTP server, the local router sends the request to 192.168.1.2.

Example 2:

Assume that the local LAN network is 192.168.1.0 255.255.255.0. When the port value of 0 (zero) is used, it directs all ports of the specified protocol to the IP address specified.

```
remote addServer 192.168.1.4 tcp 0 router1
```

Note: addserver commands using specific port numbers take priority over the port 0 setting. 192.168.1.4 will be asked to serve requests coming from *router1* to the local router. If the local router also has the same Telnet and FTP entries from the previous example, 192.168.1.3 will serve the Telnet request, 192.168.1.2 will serve the FTP request, and 192.168.1.4 will serve any other request, including HTTP, SMTP, etc.

Example 3:

```
remote addServer 192.168.1.10 tcp 9000 9000 telnet route-in
remote addServer 192.168.1.11 tcp 9001 9001 telnet route-in
```

In this example, an incoming request on TCP port 9000 will be sent to 192.168.1.10 with the port changed from 9000 to the telnet port (port 23).

An incoming request on TCP port 9001 will be sent to 192.168.1.11 with the port changed from 9001 to the telnet port.

Error Message: “Failed to add server”

The error message *Failed to add server* indicates that a server entry could not be created. This can occur either due to port overlap or due to not enough memory.

Port overlap

For example, you enter:

```
# remote addserver 192.168.1.10 tcp 9000 9000 telnet router1
# remote addserver 192.168.1.11 tcp 9000 9000 telnet router1
Failed to add server
```

The second command gets an error due to port overlap. If the second server entry was allowed and the remote end sends a server request to port 9000, the router wouldn't know whether to send the request to 192.168.1.10 or 192.168.1.11.

Not enough memory was available to create an entry.

This condition should not ordinarily occur because the amount of memory needed for a server entry is less than 30 bytes. Should this problem occur, it may cause many related problems or failures.

- **System Commands**

The following two commands are used to globally enable/disable a local IP address (on your LAN) as the server for that particular protocol and/or port.

system addServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]]

system delServer <*action*> <*protocol*> <*port*> [<*last port*>[<*first private port*>]]

For more information, see the command descriptions on page 233 and page 244.

Remember to type **save** to make the changes persistent across boots.

Examples:

```
system addserver 192.168.1.5 tcp smtp
system addserver 192.168.1.6 tcp 0
system addserver 192.168.1.6 udp 0
```

The router sends a server request for SMTP to 192.168.1.5 when such a request comes from any remote router running NAT. The router sends any other server request (tcp or udp) to 192.168.1.6.

Server Request Hierarchy

As shown above, multiple **system addserver**, **remote addserver**, and **eth ip addserver** commands can designate different servers for different protocols, ports, and interfaces. When handling a request from a remote router (to which the local router has NAT enabled), the local router searches the server list for the appropriate server. The following lists the order of search and the command that added the server to the list:

Search Order	Command
1. Protocol and port for a specific interface	**remote addserver** *or* **eth ip addserver**
2. Protocol and port for any interface	**system addserver**
3. Protocol and any port for a specific interface	**remote addserver** with *port* **0** *or* **eth ip addserver** with *port* **0**
4. Protocol and any port for any interface	**system addserver** with *port* **0**
5. Any protocol and any port for a specific interface	**remote addserver** with *protocol* **all** and *port* **0** **eth ip addserver** with *protocol* **all** and *port* **0**
6. Any protocol and any port for any interface	**system addserver** with *protocol* **all** and *port* **0**
7. Local LAN IP address mapped to the WAN interface IP address.	**system addhostmapping**

8. If none of the above, the local router selects itself (the local router) as the server.

Classic NAT

With classic NAT, one PC IP address is translated to one NAT IP address. This NAT technique is primarily used to make certain hosts on a private LAN globally visible and give them the ability to remap these IP addresses as well.

Client Configuration

Classic NAT requires that you first enable NAT Masquerading (as described in the previous section); thus, for the Classic and Masquerading forms of NAT, the clients are configured in the same way. Refer to the *Client Configuration, page 96* section.

Host Remapping

- #### Interface-Specific Commands

  You can enable and disable host remapping for specific remote interfaces and for specific Ethernet interfaces. To enable or disable host remapping on a per-remote basis, use these commands:

  remote addHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*remoteName*>

  remote delHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*remoteName*>

  Use the command **remote addHostMapping** whenever a host on the local LAN is known by different IP addresses to different remotes.

  To enable or disable host remapping on a per-Ethernet-interface basis, use these commands:

  eth ip addHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*interface*>

  eth ip delHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*interface*>

- #### System Commands

  Use these commands to enable or disable host remapping systemwide:

  system addHostMapping <*first private addr*> <*second private addr*> <*first public addr*>
  system delHostMapping <*first private addr*> <*second private addr*> <*first public addr*>

  Use the command **system addHostMapping** whenever a host on the local LAN is known by the same IP address on all remotes.

- #### IP Address Range

  The range of local LAN IP addresses to be remapped is defined by <*first private addr*> to <*second private addr*> inclusive. These addresses are mapped one-to-one to the public addresses.

The range of public IP addresses is defined by <*first public addr*> only. The rest of the range is computed automatically (from <*first public addr*> to <*first public addr*> + number of addresses remapped - 1) inclusive.

• Multiple-Host Remapping Entries

Users may enter as many host remapping entries as they wish.

Example:

```
remote addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 remote1
remote addHostMapping 192.168.207.93 192.168.207.99 10.0.20.4 remote1
remote addHostMapping 192.168.209.71 192.168.209.80 10.12.14.16 remote1
```

The above entries create three mappings:

192.168.207.40 through 192.168.207.49 are mapped to 10.0.20.11 through 10.0.20.20
192.168.207.93 through 192.168.207.99 are mapped to 10.0.20.4 through 10.0.20.10
192.168.209.71 through 192.168.209.80 are mapped to 10.12.14.16 through 10.12.14.25

• Range Overlap Rules

- The per-interface commands, **remote addHostMapping** and **eth ip addHostMapping** have these range overlap rules:

  Private IP address ranges cannot overlap for an interface.
  Public IP address ranges cannot overlap for an interface.

- The global command, **system addHostMapping**, has these range overlap rules:

  Private IP address ranges cannot overlap for a system.
  Public IP address ranges cannot overlap for a system.

- If a private IP address range for an interface and a private IP address range for the system overlap, the private IP address range for the interface has precedence.
- If a public IP address range for an interface and the public IP address range for the system overlap, the public IP address range for the interface has precedence.
- Private IP addresses and public IP addresses can be the same.

For example, to enable IP/port translation to a remote router and make the IP addresses 10.1.1.7 through 10.1.1.10 globally visible, it is permissible to use either one of the following commands:

```
remote addHostMapping 10.1.1.7 10.1.1.10 10.1.1.7 remoteName
system addHostMapping 10.1.1.7 10.1.1.10 10.1.1.7
```

If the remapped host's IP address (classic NAT, one-to-one IP address translation) and the masquerading IP address (many-to-one IP address translation) are the same, then NAT masquerading has precedence over classic NAT.

NetMeeting (H.323) with NAT

NetMeeting is an application that uses the TCP protocol H.323 (and, for certain options, T.120). If all NetMeeting connections are outgoing, NAT does not interfere and no additional configuration is needed. However, if

incoming NetMeeting calls from outside the local LAN are to be received, NAT needs additional directions from you.

NAT prevents requests coming from outside the LAN from connecting to private addresses on the LAN unless you specify the connections that are to be allowed.To receive NetMeeting audio and video connections from outside the local LAN while NAT is enabled, you must enter commands to direct the outside connections. To do this, you would enter commands to either:

- direct connections for TCP ports 1720 (**h323)** and 1503 (**t120**), or
- map a public IP address to a private IP address on the LAN.

Scenario 1: Global Server Connection

Let's suppose you want one private IP address on the local LAN to receive NetMeeting audio and video connections from outside the LAN while NAT is enabled. To allow this, you specify the IP address on the following command:

system addServer <*ipaddr*> **tcp h323**

The NetMeeting options, *Share Program*, *Chat*, *Whiteboard*, and *Transfer Files* use the TCP protocol T.120. To use these options, enter another command specifying the IP address, as follows:

system addServer <*ipaddr*> **tcp t120**

All IP addresses on the LAN can continue to connect to addresses outside the LAN, but only the specified IP address can receive the specified TCP connections from the outside.

Scenario 2: Interface-Specific Server Connection

Scenario 2 is the same as scenario 1, except that you want to limit the connections from outside to a specific interface. For a remote interface, you specify the IP address and the remote name on the following commands:

remote addServer <*ipaddr*> **tcp h323** <*remote*>

remote addServer <*ipaddr*> **tcp t120** <*remote*>

For a dual-Ethernet router where the connection to the WAN is through an Ethernet interface, you would use these commands that specify the IP address and the Ethernet interface that is connected to the WAN:

eth ip addServer <*ipaddr*> **tcp h323** <*interface*>

eth ip addServer <*ipaddr*> **tcp t120** <*interface*>

Scenario 3: Address Remapping

If the local LAN has more than one IP address visible from the WAN, you could map one of those visible IP addresses to a private IP address on the LAN. The router would then direct all connections for the "outside" IP address to the "inside" IP address. The command to do this is:

system addhostmapping <*private IPaddr*> <*privateIPaddr*> <*publicIPaddr*>

The first two parameters specify the first and last addresses in the address range. To remap just one address, you specify the same private address twice and then the public IP address.

Address remapping can also be done for a specific interface. For a remote interface, you would specify the addresses and the remote name on the following command:

remote addhostmapping *<private IPaddr> <privateIPaddr> <publicIPaddr> <remote>*

For an Ethernet interface, you would specify the addresses and the Ethernet interface on this command:

eth ip addhostmapping *<private IPaddr> <privateIPaddr> <publicIPaddr> <interface>*

PPPoE (PPP over Ethernet)

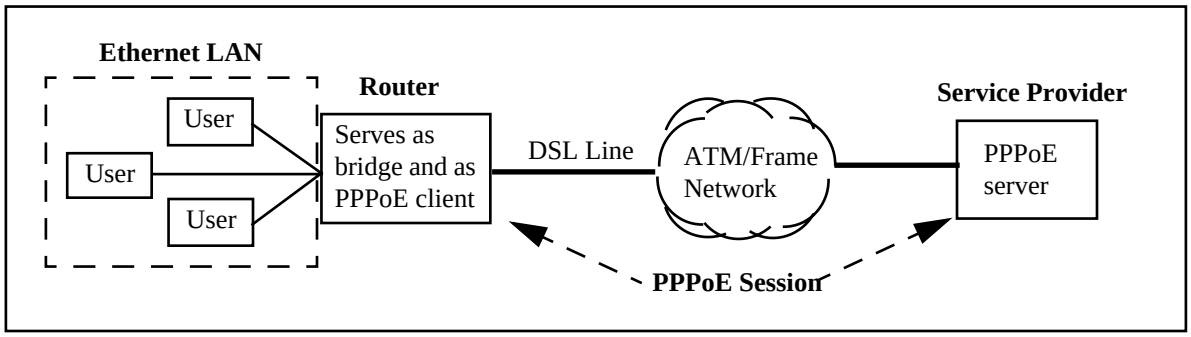
PPPoE is a method of delivering PPP sessions over an Ethernet LAN connected to a DSL line, as defined in the document RFC 2516. It was designed to maintain the established PPP interface for the end user and the service provider, while improving service through use of a DSL line.

- PPPoE allows the user to connect to a service provider using the same PPP interface as for a dialup connection, but the connection is through a DSL line, which provides greater speed and bandwidth.
- The service provider also perceives the connection as a standard PPP session, allowing for the same access control and billing per user as before.
- Multiple PPP users share the same DSL line to connect to an access concentrator.

Our router provides additional advantages to PPPoE users and service providers, as follows.

- Using our router, no software changes are required in the user PCs. Because the router acts as the PPPoE client, no PPPoE software is needed in the PC.
- Our router acts as both the PPPoE client and as the bridge connecting the Ethernet LAN to the DSL line. It does all IP address translation.
- The PPPoE client information (user name, password, and domain) are configured into the router. Once configured, the user does not need to enter them, ever.

The following diagram illustrates how our router connects an Ethernet LAN to a service provider by serving as both the bridge and the PPPoE client.



Configuring for PPPoE

Configuring the router for PPPoE requires that at least two remote router entries be defined in the router. One remote router entry serves as a bridge for PPPoE traffic. The other remote router entry serves as the PPPoE client.

PPPoE Bridge

PPPoE requires a remote router entry defined for bridging. All PPPoE traffic must be bridged through the PVC or DLCI of a remote router entry. The entry can use any protocol that supports bridging including PPP, RFC 1483, or RFC 1490.

The remote entry must be enabled for bridging using the **remote enabridge** command.

The PPPoE bridge does not require the Spanning Tree Protocol. Turn off the protocol with this command:

remote setBrOptions stp off <*remoteName*>

In addition, if the remote entry should be used only for PPPoE traffic, define it as “PPPoE only” using this command:

remote setBrOptions pppoeOnly on <*remoteName*>

For a Dual-Ethernet router, an Ethernet interface can be designated as “PPPoE only” using this command:

eth br options pppoeOnly on <*port#*>

PPPoE Client

PPPoE configuration requires creation of a new remote router entry to serve as the PPPoE client. The PPPoE client provides the user name, password, and domain name required for each PPPoE session. In our router, we refer to the PPPoE domain name as a “service name” as described later.

The user name and password can be the router name and password provided by the **system name** and **system passwd** commands. Or a name and password can be specified for the remote router entry using the **remote setOurSysName** and **remote setOurPasswd** commands.

To create the entry, begin by entering these two commands:

remote add <*remoteName*>

remote setPPPoEservice * <*remoteName*>

The preceding two commands create a remote router entry that can be used to connect to all PPPoE services. To create an entry for a specific PPPoE service, use the following two commands:

remote add <*remoteName*>

remote setPPPoEservice <*serviceName*> <*remoteName*>

The service name is the domain name defined by your service provider.

After defining the remote entry with the **remote add** and **remote setPPPoEservice** commands, enter commands to:

- Turn off authentication of the remote router by the target router (**remote disauthen**).
- Specify the user name and password for the service (**remote setoursysname** and **remote setourpasswd**).
- Define the IP route for the remote (**remote addiproute**). (IP routing must be enabled for the Ethernet interface with **eth ip enable**.)
- Turn on Network Address Translation (NAT) if needed (**remote setiptranslate**).
- Permanently allocate a channel or allocate a channel only when needed (**remote setminline**).

  If your service provider charges by the hour, you may want a PPPoE session to timeout after a period of no traffic. However, if you do use a timeout, bringing up a PPPoE session takes 2-3 seconds longer.

  To permanently allocate a channel, use:

  remote setminline 1 <*remotename*>

To set up a timeout, set the minline value to 0 and specify the timeout period in seconds, as follows:

remote setminline 0 <*remotename*>
remote settimer <*seconds*> <*remotename*>

Sample PPPoE Configuration Script

The following script is an example showing commands for a PPPoE configuration. The script assumes the following:

- The VPI/VCI for the connection is **0/35**.
- The domain name for the service is **DialUpPPP.net**.
- The CHAP user name is **JaneDoe** and the CHAP password is **Secret**.
- Network Address Translation is desired for the PPPoE session.
- Only PPPoE traffic should pass through the bridge interface.
- Default IP route is used for the PPPoE session.

```
# Sample PPPoE Configuration Script
# --------------------------------
# Enable IP routing for the Ethernet interface.
eth ip enable
#
# Define a remote router entry (named PPPoEbridge) to serve as
# the bridge for PPPoE traffic only.
remote add  PPPoEbridge
#
# Set the link protocol (PPP, RFC 1483, RFC 1490).
remote setprotocol  rfc1483mer  PPPoEbridge
#
# Specify the VPI/VCI for ATM. (For Frame Relay, you would set the DLCI).
remote setpvc  0*35  PPPoEbridge
#
# Enable bridging through the remote.
remote enabridge  PPPoEbridge
#
# Turn off the Spanning Tree Protocol.
remote setbroptions stp off PPPoEbridge
#
# Allow only PPPoE traffic through this remote.
remote setbroptions  pppoeonly on PPPoEbridge
#
# ------------------------------------------------------------
# Define a remote router entry (named PPPoEuser) to serve as
# the PPPoE client for connections to the service DialUpPPP.net.
remote add  PPPoEuser
remote setpppoeservice  DialUpPPP.net  PPPoEuser
#
# Turn off authentication of the remote router by the target router.
remote disauthen PPPoEuser
#
# Specify the CHAP user name and password required by the service.
remote setoursysname  JaneDoe  PPPoEuser
```

```
remote setourpasswd  Secret  PPPoEuser
#
# Define an IP route for the remote.
remote addiproute  0.0.0.0  0.0.0.0  1  PPPoEuser
#
# Turn on Network Address Translation for the remote.
remote setiptranslate  on  PPPoEuser
#
# Permanently allocate a channel for the connection.
remote setminline  1  PPPoEuser
# To have PPPoE sessions timeout after 10 min. (600 sec.) of no traffic,
# change the setminline value to 0 and add this command:
# remote settimer  600  PPPoEuser
#
# ------------------------------------------------
# Save the configuration changes and then reboot.
save
reboot
```

Managing PPPoE Sessions

Each PPPoE session is listed with the other interfaces in the output from an **ifs** command. In the following example, the PPPoE session is shown as the last line of the output.

```
# ifs
Interface   Speed       In %     Out % Protocol     State    Connection
ETHERNET/0 10.0.mb     0%/0%     0%/0% (Ethernet)   OPENED
DMT/0       8.0mb D    0%/0%           (ATM)        OPENED
            800kb U               0%/0% (ATM)        OPENED
ATM-VC/1    8.0mb D    0%/0%           (ATM)        OPENED   to PPPoEbridge
            800kb U               0%/0% (ATM)        OPENED   to PPPoEbridge
ATM-ECHO/2  8.0mb D    0%/0%           (ATM)        OPENED
            800kb U               0%/0% (ATM)        OPENED
CONSOLE/0  9600 b      0%/0%     0%/0% (TTY)        OPENED
PPPoE/1    10.0 mb     0%/0%     0%/0% (PPP)        OPENED   to PPPoEuser
```

You can list more information about the current PPPoE sessions using the **pppoe list** command. The following is an example:

```
# pppoe list
PPPoE Client Session ...... DialUpPPP.net
      PPPoE/Ifs number..... 1
      Access Concentrator.. 15021109931568-efficient
      Peer MAC Address .... 00:10:67:00:66:E2
      Session ID .......... 2
      State ............... 2
      Flags ............... 1
```

To close a PPPoE session before it terminates, use the **pppoe close** command. The session is specified by its number. (Use the `PPPoE/n` number from the **ifs** output or the `PPPoE/Ifs number` from the **pppoe list** output.)

Controlling Remote Management

With the following security control features, the user can control remote management of the router via Telnet, HTTP, Syslog, and/or SNMP. Disabling SNMP stops the Configuration Manager from accessing the router, which in some environments is desirable.

Router system event messages can be automatically sent to a Unix Syslog server. The **system syslogport** and **system addsyslogfilter** commands control the port number and valid IP addresses. For more information, see Syslog Client, page 168.

Disabling Remote Management

To completely disable remote management, enter the following commands from the command line:

system telnetport disabled
system snmpport disabled
system httpport disabled
system syslogport disabled
save
reboot

Re-enabling Remote Management

To reestablish the disabled remote management services, restore the default values with the commands:

system telnetport default
system snmpport default
system httpport default
system syslogport default

Validating Clients

The following commands are used to validate clients for Telnet, SNMP, HTTP, or Syslog. They define a range of IP addresses that are allowed to access the router via that interface. Only the IP addresses in the range specified for the interface can access the router via that interface. This validation feature is **off** by default.

Multiple address ranges can be specified for each filter. If no range is defined, then access to the router is through the LAN or WAN.

Note: These commands do *not* require a reboot and are effective immediately.

system addtelnetfilter <*first ip addr*> [<*last ip addr*>] | LAN
system addsnmpfilter <*first ip addr*> [<*last ip addr*>] | LAN
system addhttpfilter <*first ip addr*> [<*last ip addr*>] | LAN
system addsyslogfilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr	First IP address of the client range
last ip addr	Last IP address of the client range. May be omitted if the range contains only one IP address.
LAN	Local Ethernet LAN

Example:

```
system addsnmpfilter 192.168.1.5 192.168.1.12
```

To delete client ranges previously defined, use these commands:

system deltelnetfilter <*first ip addr*> [<*last ip addr*>] | LAN
system delsnmpfilter <*first ip addr*> [<*last ip addr*>] | LAN
system delhttpfilter <*first ip addr*> [<*last ip addr*>] | LAN
system delsyslogfilter <*first ip addr*> [<*last ip addr*>] | LAN

To list the range of allowed clients, use the command:

system list

Restricting Remote Access

To allow remote management while making it more difficult for non-authorized persons to access the router, you may redefine the ports to a less well-known value. When Network Address Translation (NAT) is used, this port redefinition feature also allows you to continue using the standard ports with another device on the LAN (provided the appropriate NAT server ports commands are issued), while simultaneously managing the router (with non-standard ports).

For example, the following commands redefine the Telnet, SNMP, HTTP, and Syslog ports:

```
system telnetport 4321
system snmpport 3214
system httpport 5678
system syslogport 6789
```

Changing the SNMP Community Name

Changing the SNMP community name from its default value of “public” to another string may further enhance SNMP security. This string then acts like a password, but this password is sent in the clear over the WAN/LAN, in accordance with the SNMP specification.

Use the following commands to change the SNMP community name.

system community <*new community name*>
save
reboot

Disabling WAN Management

You can allow management of the router on the local LAN, but not over the WAN. If the router has been configured to use NAT, you can define two servers that *do not* exist on the LAN side to handle WAN SNMP and Telnet requests, and thus WAN management of the router cannot occur.

The following example shows how this is done. It assumes there is no computer at 192.168.254.128.

```
system addServer 192.168.254.128 udp snmp
system addServer 192.168.254.128 tcp telnet
system addServer 192.168.254.128 tcp http
save
reboot
```

Dial Backup

The Dial Backup capability provides a backup asynchronous modem connection to the Internet when the default DSL link goes down. The modem connection is provided through the console port. In this case, the console port is used as a serial port and must be connected to an external modem.

Note: The Dial Backup feature is effective using either V.90 or ISDN modems.

Dial Backup is intended for customers with critical applications for which continuous Internet access is vital. If the DSL link for those applications goes down, the router can automatically switch their traffic to the asynchronous modem. Later, after determining that the DSL link is, once again, up and stable, the router automatically switches the modem traffic back to the DSL link.

This feature may also be useful for a customer whose DSL line is not yet installed. The router can begin providing service through an asynchronous modem and later automatically switch to the DSL link when it becomes available.

Dial Backup can be used with a VoDSL (voice over DSL) router. However, when data traffic is switched to the backup modem or restored to the DSL connection, all voice calls are terminated.

Dial Backup with a Tunnel

Dial Backup works with L2TP and IPSec tunneled connections. However, an IPSec tunnel from the backup interface must use IKE aggressive mode, not IKE main mode, because, it is assumed that the ISP assigns an IP address to the backup interface dynamically (see Main Mode and Aggressive Mode, page 152.)

You may wish to restrict an L2TP tunnel or IPSec tunnel to *only* the primary interface or *only* the backup interface:

- If you do not want tunnel traffic to go through the backup asynchronous modem, you should restrict the tunnel to use *only* the primary interface. With this restriction in place, if the primary interface fails, the tunnel is terminated, and it is *not* re-established with the backup interface.
- Or, you might want a tunnel to be established only when the asynchronous modem is being used. In this case, you would restrict the tunnel to the backup interface only.

To set either restriction for an L2TP tunnel, use the command **l2tp set wanif** (page 369). On the command, you specify the remote name that the tunnel is restricted to and the tunnel name. To restrict the tunnel to the backup interface, specify the remote name that you created for the dialup parameters as described in Specifying the Dialup Parameters, page 110.

To set a restriction for an IPSec tunnel, use the command **ike ipsec set interface** (page 379). The interface that you specify on the command is the remote interface that the tunnel is to be restricted to. To restrict the tunnel to the backup interface, specify the remote name that you created for the dialup parameters as described in Specifying the Dialup Parameters, page 110.

Configuring Dial Backup

To set up the router to use the Dial Backup feature, you must:

- Connect an asynchronous modem to the console port of the router.

Special DB9 or DB25 connectors may be required. Special modem kit and/or connector packages are available from Efficient Networks.

- Configure the router software to use the Dial Backup feature.

  To begin Dial Backup configuration, you can select options using the web GUI or begin with the sample configuration file included on the CD as `dsl/samples/backup.txt`. Further configuration may require the CLI commands described in this section.

  Note: Because Dial Backup uses the console port, you *cannot* enter CLI commands using the console port while Dial Backup is enabled. While Dial Backup is enabled, you must access the command line via Telnet (see Telnet Session for Remote Access, page 16).

The following is a general outline of the steps required to configure Dial Backup. These steps are detailed in the following sections. To configure Dial-Backup:

1. Check that the Dial Backup feature is installed in the router. To do so, enter the **vers** command and look for DIAL-BACKUP in the option list. For more information, see Software Option Keys, page 124.

2. Define a remote profile for Dial Backup that specifies the ISP phone number and other dialup parameters.

3. Specify the conditions that determine the status of the DSL link. Default values are provided for:

   - Minimum stability period for the DSL link status signal
   - Minimum retry period before DSL link restoration is attempted

   Optionally, Dial Backup can actively test the status of the DSL link by pinging IP addresses. For this option, you must specify at least one IP address; default values are provided for:

   - Ping interval, number of samples, and minimum success rate

4. Specify the modem parameters (if the default values are not appropriate).

5. Enable Dial Backup by doing *all* of the following:

   - Check that the remote profile created in step 2 is enabled (use the command **remote list**).
   - Enter the command **system backup enable**.
   - Enter the commands **save** and **reboot**.

   Note: The router determines *only* at reboot whether its serial port is to be used for console output or for Dial Backup. If Dial Backup is enabled at reboot, then the serial port is assigned to Dial Backup and console output is *not* sent to the serial port; this cannot change until the next reboot.

Specifying the Dialup Parameters

To use the asynchronous modem to connect to the ISP, the router requires a remote entry defining the connection parameters for the serial port.

Dial Backup can be enabled *only* when a remote entry exists that:

- defines an asynchronous interface using the PPP protocol,
- specifies at least one phone number,
- specifies a user name, *and*
- is enabled.

The remote entry should also turn off authentication and specify a remote route.

The following is an example of commands that define a Dial Backup remote profile named **backup**.

```
remote add backup

# Define the interface as asynchronous and using the PPP protocol.

remote setprefer async backup

remote setprotocol ppp backup

# Specify the primary phone number to be used when dialing out. This phone
# number begins with 9 (to get an outside line), a comma (for a 2-second
# pause), and finally the seven-digit local number.

remote setphone async 1 9,5554218 backup

# Specify the bit rate for the preceding phone number.
# The bit rate can be 38400, 57600, 115200, or 230400.

remote setspeed 115200 async 1 backup

# Specify the alternative phone number to be used and its bit rate.

remote setphone async 2 9,5554219 backup

remote setspeed 115200 async 2 backup

# Specify the name and password provided by the ISP.

remote setoursysname GWBush backup

remote setourpasswd Dubya backup

# Turn off authentication.

remote disauthen backup

# Turn on Network Address Translation.

remote setiptranslate on backup

# Add a default route for the backup entry

remote addiproute  0.0.0.0  0.0.0.0  1  backup

save
```

ISDN Phone Numbers

If you use an ISDN Terminal Adapter (TA) instead of a V.90 modem, the remote profile for the Dial Backup should:

- specify an asynchronous interface (**remote setprefer async**) and,
- if the two B channels require different phone numbers, specify both phone numbers on one **remote setphone** command. The two phone numbers are separated by an & character. For example, the following command specifies the two phone numbers 555-2000 and 555-4000:

  ```
  remote setphone async 1 5552000&5554000 backup
  ```

Setting DSL Link Conditions

After you define the backup connection parameters in a remote profile, the following information is included when you enter the command **system list**:

```
Backup.............................. yes
   Retry Interval In Minutes.......... 30
   Stability Interval In Minutes...... 3
```

By default, Dial Backup determines that the DSL link has failed if it detects **No DSL link status signal.** If the signal remains down for a minimum time (the stability period), the DSL link is assumed to be physically disconnected and down.

Optionally, you may also specify one or more IP addresses to ping to determine that the link is down. This is discussed later under Addresses to Ping, page 113.

Stability Period

DSL link failure is indicated if the DSL link status signal remains down for a minimum time. This minimum time is the **stability** period that guards against frequent switching back and forth between the DSL link and the backup port.

The default stability period is three minutes. To change the stability period, use this command:

```
system backup stability <minutes>
```

The minimum stability period is one minute.

DSL Restoration Retry Period

Once DSL link failure is determined, the router uses its console port as a serial port and data traffic is sent and received through the asynchronous modem connected to that port. This backup port continues to be used until it is time to check whether the DSL link has been restored. This time period between checks is called the **retry** period (default, 30 minutes).

When the retry period expires, the router determines if the DSL link has been restored. To do so, it first determines if the DSL link status signal has been up for the minimum stability period. If it has, then the router stops the data traffic going through the backup asynchronous modem, and checks whether the DSL link can be used instead.

If you have specified one or more ping addresses, the router pings those addresses via the DSL link. If the DSL link fails the ping test, the router once again switches data traffic to the backup port until the retry period expires again.

However, if the DSL link passes the ping test, the DSL link is assumed to be restored and it is used for data traffic until another failure is detected.

The default retry period is 30 minutes. To change the **retry** period, enter this command:

```
system backup retry <minutes>
```

Addresses to Ping

Dial Backup can also actively determine whether the DSL link is up by pinging IP addresses. It does so only if you provide it with one or more IP addresses.

You could choose to ping addresses that are vital to your application. The router pings these addresses at the interval you specify (default, every 5 seconds). It compares a specified number of samples (default, 6) against the specified minimum success rate (default, 50%). If the success rate is less than the minimum, the DSL link is assumed to be down.

If you specify one or more addresses, the router pings those addresses to determine if the DSL link is up. You may request that the router ping any or all of these:

- One or more specific IP addresses (four decimals separated by periods)
- Your gateway address (GW)
- Your domain name server address (DNS).

The router determines your gateway and/or DNS address implicitly via a means such as DHCP, static configuration, PPP negotiation, etc.

If you specify more than one address to ping, you may want to assign the addresses to **groups**. Each group can be assigned its own ping interval, number of samples, and success rate. For example, you might want the success rate for the DNS address to be at least 95%, while a success rate of 50% would be reasonable for a heavily used website. You can also disable and re-enable ping addresses by group. A group is identified by its number (0 through 65535).

To add an address to the ping list, use this command:

```
system backup add <ipaddr> | GW | DNS  [<group>]
```

After you enter a ping address, you can see the ping list using the command **system list**. For example, the addresses in this ping list are the gateway (GW) address and the domain name server (DNS) address:

```
IP Address(es)................... GW
                                  DNS
```

To remove an address from the ping list, use this command:

```
system backup delete <ipaddr> | GW | DNS  [<group>]
```

To remove a group of addresses, enter:

```
system backup delete all [<group>]
```

To clear the ping list of all addresses, enter:

```
system backup delete all all
```

Note: If you clear the ping list of all addresses, pinging is not used to determine if the DSL link is down. Instead, the state of the DSL physical layer is the only criterion used to determine failure and restoration.

Ping Interval, Number of Samples, and Success Rate

After you enter an address in the ping list, the **system list** command lists the following Dial Backup information:

```
Backup.............................. yes
    Retry Interval In Minutes.......... 30
    Stability Interval In Minutes...... 3
    Backup Group....................... 0
      Group Enabled.................... yes
      Ping Interval In Seconds......... 5
      Number Of Ping Samples........... 6
      Target Success Rate.............. 50
      Current Success Rate............. 100
      IP Address(es)................... GW
```

By default, the router pings the addresses every 5 seconds until it has pinged each address 6 times; it requires a minimum success rate of 50%. You may need to adjust these default values to fit your situation; for example, if pings are failing, you may want to lower the required success rate. To change these values, use these commands:

```
system backup pinginterval <seconds> [<group>]

system backup pingsamples <samples> [<group>]

system backup successrate <percentage> [<group>]
```

Note: To disable a group of ping addresses, specify **0** for any of its three values— pinginterval, pingsamples, or success rate.

The same ping interval, number of samples, and success rate apply to all addresses assigned to a group. (Any address not assigned to a group is considered to belong to group 0.) All groups are tested in parallel. As soon as any group fails its success rate test, the DSL link is assumed to have failed and the switchover to the backup is performed.

During the ping test, every address in a group contributes to the current success rate of the group; as soon as the current success rate falls below the minimum success rate, the group has failed. For example, if the minimum success rate is 50% and the sample number is 6, the maximum sample size for a three-address group is 18 (6 times 3); thus, as soon as the group accumulates 10 failures (one more than 9 failures, which is 50% of 18), the group fails.

Specifying Modem Parameters

You need to provide the router with modem parameters so it can effectively use the asynchronous modem connected to the console port. A default modem setup is provided. To see the default settings, enter:

```
# system defaultmodem

# system list

MODEM STRINGS:
  Reset:    ATZ
  Escape:   +++
  Init:     ATS0=0Q0V1&C1&D0X4S12=20
  Off-Hook: ATH1
  Dial:     ATDT
  Answer:   ATA
  Hangup:   ATH0
```

To change the modem settings from the defaults, specify which setting you want to change and the new string. To do so, use this command:

```
system modem reset | escape | init | offhook | dial | answer | hangup  <string>
```

For example, the following command changes the string for the **init** setting:

```
system modem init  ATS0=0Q0V1&C2&D3&K1X4&H1&I0S12=20
```

Init Setting

The modem **init** string should set the following:

DTR	off	Suppress results	on
Verbal	yes	Auto answer	off
Echo	no	Carrier detect	off

Use HyperTerminal directly connected to the modem to determine the modem **init** string before connecting the modem to the router.

Dial Setting

The string for the **dial** setting can be either **ATDT** for tone dialing or **ATDP** for pulse dialing. The default is tone dialing. To select pulse dialing, use this command:

```
system modem dial ATDP
```

Disabling and Re-Enabling Dial Backup

Note: Because Dial Backup uses the console port, you must use the Web GUI interface or a Telnet session to disable Dial Backup.

To *temporarily* disable Dial Backup, enter the following command:

```
system backup disable
```

This command stops Dial Backup. However, temporarily disabling Dial Backup does not change the use of the console port (no console output is sent to the console port).

To re-enable Dial Backup after it has been *temporarily* disabled, either reboot without a save or enter this command:

```
system backup enable
```

Note: You can change the setting of the Dial Backup enable switch at any time, but toggling the switch does not immediately change the use of the console port. The use of the console port is determined *only* at reboot.

To disable Dial Backup *across reboots* and change the use of the console port, enter the following commands:

```
system backup disable
save
reboot
```

Assuming that the Dial Backup remote profile is enabled, you can re-enable the Dial Backup feature using the following commands:

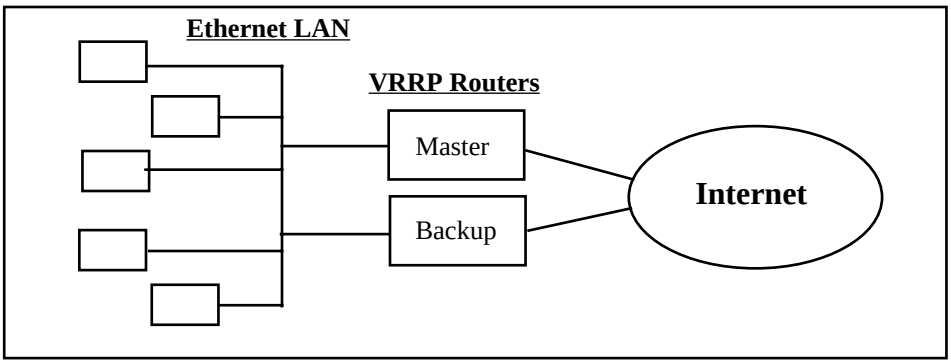
```
system backup enable
save
reboot
```

VRRP Backup

When a router is defined as a static default gateway and no other dynamic routing protocol or router discovery protocol is used (such as RIP, page 83), the gateway becomes a critical link in the network. If that router fails, that critical link would be broken. It, therefore, may be appropriate to set up other routers as backups that can serve as the static default gateway if necessary.

The Virtual Router Redundancy Protocol (VRRP), as defined in RFC 2338, allows other IP routers in a LAN to provide immediate and automatic backup to a failed IP router. VRRP is a protocol that defines how backup routers monitor the status of a master router and take over its function if it fails. The new master router adopts the IP and MAC address of the original master, so that the hosts configured with the single default gateway maintain their network connection.

The following illustration shows two routers connecting a LAN to the Internet. By using VRRP, the backup router can take over as the gateway if the master router fails.



Routers using VRRP send out advertisement packets at intervals to let the other VRRP routers on the LAN know that they are still up. The other VRRP routers realize that a router is down when no advertisement packets have been received for the minimum down interval. The VRRP router assigned the highest priority takes over for the failed router. When the failed router is restored, it can automatically preempt the backup router and resume its function in the network.

VRRP Configuration

To configure a LAN to use VRRP, you must enter configuration commands into every router that is to be provided with backup or that is to serve as backup to another router. Certain values must be the same between the master router and its backups; other values must differ (as discussed in the following sections).

VRRP configuration requires these basic steps:

1. Define logical interfaces.
2. Define the ID of the Virtual Router (VRID).

3. Define the VRRP attributes of the Virtual Router.

4. Save the changes and either restart the VRRP interface or reboot the router.

Defining the VRRP Interface

Each router that is to use VRRP must have at least two logical Ethernet interfaces defined, one to be used as the VRRP interface and the other as the management interface. (Logical interfaces are discussed under IP Subnets, page 79.)

The VRRP interface is for VRRP use only; it cannot be used for any other purpose. Unlike other logical interfaces, the VRRP interface does not use the usual Ethernet MAC address associated with the router. Instead, it uses the VRRP MAC address as defined in RFC 2338, that is, 00005e0001xx where xx is the VRID.

IP Address

Every logical interface is assigned its own IP address, or range of addresses, that is unique on the LAN. The VRRP interface must be assigned the IP address that serves as the default static gateway for other devices on the LAN.

For example, assume that the gateway IP address is 192.168.100.254. If the default logical interface (0:0) is to be the VRRP interface, it is assigned the gateway address. Another logical interface (0:1) is defined to be the management interface and is assigned another IP address.

```
eth ip addr 192.168.100.254 255.255.255.0
eth add 0:1
eth ip addr 192.168.254.253 255.255.255.0 0:1
```

Note: You must assign the *same* IP address to the VRRP interface in the master router and in every router that is to serve as its backup. For example, if the VRRP interface is assigned IP address 192.168.100.254 in router A, the VRRP interface in every backup router for router A must be assigned IP address 192.168.100.254.

RIP Processing

Routers using VRRP do not need RIP protocol processing to discover routes. (See RIP Controls, page 83.) You may, therefore, turn off RIP processing using these commands:

```
eth ip options txrip off
eth ip options rxrip off
```

Defining the VRID

The next step is to define a virtual router ID, or VRID, and associate it with the logical Ethernet interface that is to be the VRRP interface. (The management interface is not assigned a VRID).

For example, the following command assigns the VRID 7 to the logical interface 0:1 that is to serve as the VRRP interface.

```
eth ip vrid 7 0:1
```

A VRID has these characteristics:

- Integer from 1 through 255; thus, a LAN can have up to 255 VRIDs.

- Unique on the LAN, but can be reused on other LANs.
- The same VRID must be defined in all routers that make up the Virtual Router, that is, the original router and all routers that are to serve as its backups. For example, if VRID 7 is defined in router A, then VRID 7 must also be defined in all backup routers for router A.

To see the effect of these commands, specify the logical interface on an **eth list** command. For example, the defined VRID is listed in the following output:

```
# eth list 0:1
ETHERNET INFORMATION FOR <ETHERNET/0:1>, DOWN
  Send IP RIP to the LAN.............. no
    Advertise me as default router..... yes
  Process IP RIP packets received..... no
    Receive default route by RIP....... yes
  IP address translation.............. no
  IP filters defined.................. no
  IP address/subnet mask.............. 192.168.254.253/255.255.255.0
  Management IP address/subnet mask.... 0.0.0.0/0.0.0.0
  Virtual Router ID (vrid)............ 7
  Static Ethernet routes defined...... none
  Virtual Ethernet routes defined...... none
  MTU................................. default
```

Note: A logical interface does not become effective until you **save** your changes and either **restart** the logical interface or **reboot** the router. The VRRP interface also requires the definition of its VRRP record before it becomes effective. See Starting VRRP, page 120.

Defining VRRP Attributes

Each time you define a VRID in a router, you must define an attribute record for it in that router. The following sections describe how to define the record and set the attributes.

Note: The VRRP attribute commands do not require a restart or reboot to take effect. However, you do need to **save** your changes if they are to persist after a restart or reboot.

Adding a VRID Attribute Record

To define a record to contain the attributes for a VRID in a router, use this command:

eth vrrp add <*vrid*> [<*port#*>]

The port number is needed only if the router is an Ethernet hub router with two ports (port 0 and port 1).

To see the VRID attribute records currently defined, use the **eth vrrp list** command, as follows:

```
# eth vrrp list
    VRRP Records for Ethernet Port .... 0
      VRRP Record..................... VRID   7, Priority 100, Interval   1
                                       Flags:preempt, No Authentication
                                       No VRRP interfaces defined
```

Priority Attribute (0-255, default, 100)

The priority value determines which backup router takes over when a router fails. The master router must be assigned the highest priority (255). Lower priorities are assigned to its backup routers, that is, the other routers in which the same VRID is defined.

For example, suppose routers A, B, and C all have VRID 7 defined. If router B should take over if router A fails and if router C should take over if both A and B fail, you would assign priority 255 to A and lower priorities to B and C, such as, priority 100 to B and priority 50 to C.

The priority command is: **eth vrrp set priority** <*priority*> <*vrid*> [<*port#*>]

Time Interval Attribute (default, 1 second)

The time interval value specifies how often VRRP advertisement packets are sent. It also determines how quickly a backup router can recognize that another VRRP router is down.

If the backup does not receive a VRRP packet from another VRRP router during the master down interval, the backup assumes the other router is down. The master down interval is:

Master _Down_Interval = (3 * Time_Interval) + Skew_Time

Skew_Time = (256 - Priority) / 256

Thus, the default skew time is (256 - 100) / 256, or .609375. The default master down interval is (3 * 1) + .609375, or 3.609375 seconds.

Note: The time interval must be the *same* for every router in the Virtual Router, that is, for every router in the LAN with the same VRID. For example, if a VRRP interface in routers A, B, and C has the VRID 7, routers A, B, and C must all specify the same time interval for VRID 7.

The time interval command is:**eth vrrp set timeinterval** <*seconds*> <*vrid*> [<*port#*>]

Password Attribute (no default)

You may specify an optional password of 1 to 8 characters. The password is only used to authenticate VRRP advertisement packets. It is sent as clear text on the LAN. If you do not specify a password, no password authentication is done.

Note: The password must be the *same* for every router in the Virtual Router, that is, for every router in the LAN with the same VRID. For example, if a VRRP interface in routers A, B, and C has the VRID 7, routers A, B, and C must all specify the same password for VRID 7.

The password command is: **eth vrrp set password** <*string*> <*vrid*> [<*port#*>]

The command to clear the password is: **eth vrrp clear password** <*vrid*> [<*port#*>]

Note: Our implementation does not validate the IP addresses in the advertisement packet or authenticate using an authentication header.

Preemption Option (default, preempt)

The preemption option determines what the router does when it recovers from a failure, as follows:

- If the router is the master router for the IP address (it has priority 255), it always immediately preempts the backup router and resumes its function in the network. The preemption option cannot change this.
- However, if the router is a backup router for the IP address and it determines that a router with a lower priority is currently functioning as backup, the preemption option determines whether this router immediately preempts the router with lower priority or waits for the lower priority router to go away before becoming the active VRRP router.

The preemption setting may differ among the backup routers for a VRID.

The preemption command is: **eth vrrp set option** <preempt | nopreempt> <*vrid*> [<*port#*>]

Starting VRRP

After you have defined the VRRP logical interface, defined a VRID, and defined an attribute record for the VRID, you are ready to start VRRP. To do so, you must both **save** your changes and either **restart** the VRRP interface or **reboot** the router.

For example, these commands save all changes, restart the VRRP interface 0:1, and list the VRRP records:

```
# save
# eth restart 0:1
# 04/16/2001-07:49:04:VRRP: VRRP 7 on Interface ETHERNET/0 now active
eth vrrp list
    VRRP Records for Ethernet Port .... 0
      VRRP Record..................... VRID   7, Priority 100, Interval   1
                                       Flags:preempt, No Authentication

                                       Interface: ETHERNET/0:1
                                       VRRP Router active
```

After you start VRRP, you can use the **eth vrrp list** or **eth list** commands to monitor the status of the VRRP router.

Disabling or Deleting VRRP

To disable a Virtual Router in a router, you delete its VRID in that router. To do so, use the command:

eth vrrp delete <*vrid*> [<*port#*>]

This command deletes the VRRP attribute record defined for that VRID. It also disassociates the VRRP IP and MAC addresses from the logical interface.

Note: To re-instate a deleted VRID, you need to redefine both the VRID and the VRRP attribute record.. For example, the following commands disable VRID 7 and then re-enable it for the logical interface 0:0:

```
# eth vrrp delete 7
# eth ip vrid 7
# eth vrrp add 7
# 04/16/2001-08:36:06:VRRP: VRRP 7 on Interface ETHERNET/0 now active
```

To change the VRRP interface for a VRID, you clear the VRRP interface designation and then re-assign it. For example, to change the VRRP interface designation from 0:1 to 0:3 for VRID 7, use these commands:

```
#eth ip vrid 0 0:1
#eth ip vrid 7 0:3
```

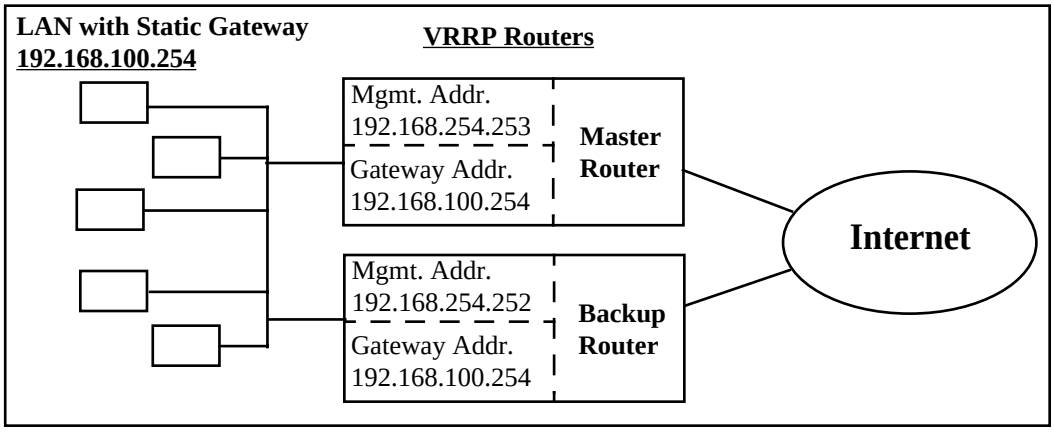
If you wanted to remove VRRP entirely from the router, you would delete the VRID and also delete the extra logical interface you created for its use, with the command:

eth delete <*port#*>:<*logical#*>

Note: Remember, to make these changes permanent, you must **save** the changes before you **restart** or **reboot**.

Sample VRRP Configuration

The sample configuration shown here is for two routers, one master and one backup. It is assumed that either router can route Internet traffic for the Ethernet LAN containing devices that use a static default gateway address 192.168.100.254.



Master Router Configuration File

These are the VRRP configuration commands for the master router.

```
# A new logical interface 0:1 will serve as the management interface.
# It is assigned the IP address 192.168.254.253
eth add 0:1
eth ip addr 192.168.254.253 255.255.255.0  0:1
#
# RIP is not needed for either interface so it is turned off.
eth ip options txrip off
eth ip options rxrip off
eth ip options txrip off 0:1
eth ip options rxrip off 0:1
#
# The default logical interface 0:0 will serve as the VRRP interface.
# It is assigned the default gateway/LAN address is 192.168.100.254.
#
eth ip addr 192.168.100.254 255.255.255.0
#
# The VRRP interface 0:0 is assigned VRID 7.
eth ip vrid 7
#
```

```
# A VRRP attribute record is defined for VRID 7.
eth vrrp add 7
#
# This router is the master router so it is given priority 255.
eth vrrp set priority 255 7
#
# This is a simple password to authenticate VRRP packets.
eth vrrp set password abcdefgh 7
#
# Use the default time interval (1 second) and preemption option (preempt).
#
# Save the changes and then reboot.
save
reboot
```

Backup Router Configuration File

These are the VRRP configuration commands for the backup router.

```
# These commands define a logical interface 0:1 to serve as the management interface.
# It is assigned an IP address unique on the LAN, 192.168.254.252.
eth add 0:1
eth ip addr 192.168.254.252 255.255.255.0  0:1
#
# RIP is not needed for either interface so it is turned off.
eth ip options txrip off
eth ip options rxrip off
eth ip options txrip off 0:1
eth ip options rxrip off 0:1
#
# In this example, the VRRP interface is the default logical interface 0:0,
# (The VRRP interfaces for the master and backup routers may have different numbers.)
#
# The VRRP IP address must be the same as that of the master router.
eth ip addr 192.168.100.254 255.255.255.0
#
# The VRRP interface must be assigned the same VRID as in the master router.
eth ip vrid 7
#
# A VRRP attribute record is defined for VRID 7.
eth vrrp add 7
#
# The backup router must have a priority less than 255. Here, the default, 100,
# is used.
eth vrrp set priority 100 7
#
# The backup router must have the same password as the master router.
eth vrrp set password abcdefgh 7
#
# The backup router must have the same time interval as the master router. In this
# example, the default, 1 second, is used.
# The default preempt option is used; it is not required to be the same as the
# master router.
```

```
#
# Save the changes and then reboot.
save
reboot
```

Chapter 5. Configuring Software Options

The features described in this chapter can be purchased as software option keys. To determine which software options are installed on your router, use the **vers** command. (If a feature has not been enabled, it is listed with a ~ prefix.)

- Encryption
- IP filtering
- L2TP tunneling
- IPSec (Internet Protocol Security) and IKE (Internet Key Exchange)
- 3DES encryption

Software Option Keys

The router has several optional software features that can be purchased as software option keys (feature activation keys) when ordering the router. These optional features are:

- IP routing
- DES or 3DES encryption (see *Encryption, page 126*)
- IP filters (see *IP Filtering, page 129*)
- L2TP tunneling (see *L2TP Tunneling — Virtual Dial-Up, page 137*)
- IPSec (see IPSec (Internet Protocol Security), page 149)

These options are usually ordered with the router. The options are controlled by the presence of a key file in flash memory or a bit set in the CMOS. Both values are checked; if either is set, the option is enabled.

Listing the Installed Software Options

To determine which software options are installed in your router, use the **vers** command. The **vers** command lists all options:

- Options that are disabled are shown with a ~ prefix.
- Options that are enabled by the key file have a + prefix.
- Options that are enabled by a set bit have no prefix.

For example, the following **vers** command output shows that the L2TP and encryption options are disabled, but all the other options are enabled. The IP Routing feature was enabled by a key.

```
Options: FRAME RELAY, SDSL, RFC1490, +IP ROUTING, IP FILTERING, WEB, ~L2TP,
 ~ENCRYPT, BRIDGE, IPX, CMMGMT
```

Adding a New Software Option Key

A software option key is a 44-character string, unique to a particular router, that enables a single feature. After receiving a software option key, you can enter it using either the web GUI or the Command Line Interface.

When using the web GUI, you select the **Upgrade Features** button and enter the key. When using the Command Line Interface, you enter the key using the following command:

key add <keystring>

Note: The new feature is not activated until the router is rebooted.

Encryption Hardware Option

The Rapid Secure Encryption (RSE) hardware option (part no. 060-xxxx-xxx) is available in certain router models. This hardware option is useful if one or more of the encryption software options are installed; it speeds up DES, 3DES, and PPP encryption.

The RSE option requires firmware at release 5.0.0 or later. If the RSE option is present, a message such as the following appears at initialization:

```
INIT: Using accelerated encryption hardware.
```

Also, if present, the RSE hardware option appears in the **vers** command output as the option **HW-DES**.

Encryption

Note: Encryption is a software option. The following section applies only to routers with the encryption option enabled. For more information, see Software Option Keys, page 124. To read about IPSec encryption, see page 149.

Two variants of encrypted data links over PPP have been implemented:

- PPP DES (Data Encryption Standard) (RFC 1969)
- Diffie-Hellman

Encryption requires PPP.

Caution: PPP DES and Diffie-Hellman encryption options may not be exported outside the United States or Canada.

PPP DES (RFC 1969) Encryption

PPP DES (Data Encryption Standard) implementation uses a 56-bit key with fixed **transmit** and **receive** keys that are specified in each router. RFC 1969 requires that users must manage the keys. This implementation has been tested for interoperability with other PPP DES vendors such as IBM and Network Express.

Configuration Commands

To configure PPP DES encryption, add these commands to your standard configuration:

remote setEncryption dese rx <*key*> <*remoteName*>
remote setEncryption dese tx <*key*> <*remoteName*>

Observe the following guidelines:

- PPP DES can only be configured using the Command Line Interface (CLI).
- The choice of keys should be carefully considered. Each key must have eight hexadecimal digits. Values that are considered cryptographically weak should be avoided. Consult a security expert for advice.
- Different keys may be used for different remote destinations.
- Use the console port to view error messages and progress. If you see “Unknown protocol” errors, the router **receive** key and **sender Tx** key don't match.
- For maximum security, Telnet and SNMP access should be disabled, and PPP CHAP authentication should be used by both ends.

Sample Configuration

Suppose that the routers SOHO (the local router) and HQ (the remote router) described in *Sample Configurations, page 65* are to be configured to use PPP DES encryption. To add encryption to their configurations, you would enter the following commands:

For router HQ:

```
remote setEncryption dese rx 1111111111111111 SOHO
remote setEncryption dese tx 2222222222222222 SOHO
save
reboot
```

For router SOHO:

```
remote setEncryption dese tx 1111111111111111 HQ
remote setEncryption dese rx 2222222222222222 HQ
save
reboot
```

Remember that the *transmit* key (**tx**) of SOHO is the *receive* key (**rx**) of HQ. Inversely, the *receive* key of SOHO is the *transmit* key of HQ.

Don't forget to save the configuration and reboot the router (**save** and **reboot** commands).

Diffie-Hellman Encryption

With Diffie-Hellman encryption, each router has an encryption file that is associated with a public key providing 768-bit security. The predefined keys can be replaced by the user. By convention, the key files have the suffix "num" (e.g., dh96.num).

Configuration Commands

To configure Diffie-Hellman encryption, add this command to your standard configuration:

remote setEncryption DESE_1_KEY | DESE_2_KEY [<*fileName*>] <*remoteName*>

Observe the following guidelines:

- Specify DESE_1_KEY if the same key is to be used in both directions. Specify DESE_2_KEY if the keys are to be different. Using the same keys in both directions can significantly reduce the time needed to compute the DES keys from the Diffie-Hellman exchange.
- The optional file name on the command is the name of the file containing the Diffie-Hellman values. If a file is not specified, default values built into the router's kernel are automatically selected. The file format is described later.
- The routers' **receive** key and **sender** Tx key must not match.
- Different keys and key files may be used for different remote destinations.
- For maximum security, Telnet and SNMP access should be disabled, and PPP CHAP should be used. Use the console port to view error messages and progress.

Sample Configuration

Suppose that the routers SOHO (the local router) and HQ (the remote router) described in *Sample Configurations, page 65* are to be configured to use Diffie-Hellman encryption. Also, assume that the same key is to be used in both directions and that the values to be used to generate keys are in the file dh96.num. To add encryption to their configurations, you would enter the following commands:

For router HQ:

```
remote setEncryption DESE_1_KEY dh96.num SOHO
save
reboot
```

For router SOHO:

```
remote setEncryption DESE_1_KEY dh96.num HQ
save
reboot
```

File Format for the Diffie-Hellman Number File

The default values used to generate keys are listed at the end of this section. If you want to use values other than the defaults, you can create your own Diffie-Hellman number file. The file should follow these rules:

- The file should be 192 bytes, in binary format, consisting of two 96-byte numbers, with the most significant byte in the first position. For example, the number 0x12345678 would appear as 000000...0012345678.
- The first 96 bytes form the *modulus*. In the equation *x' = g^x mod n*, *n* is the modulus. According to Diffie and Hellman, the modulus should be prime, and *(n-1)/2* should also be prime.
- The second 96 bytes form the *generator*, or *g* in the above equation. The generator should be a primitive root mod *n*.
- The remaining pieces of the encryption key (*x* and *y*) are randomly generated at connection time and change every time the device connects.

We recommend that you consult an encryption expert to obtain cryptographically sound generator and modulus pairs.

Default Modulus:

```
00000000: c9 b4 ed 33 ba 7f 00 9e - ce e0 83 5d a5 4c 19 25
00000010: e0 2d 99 44 e8 8d cd 16 - 02 0e 6c 26 6d 15 7c 95
00000020: 82 9a 8c 2b 19 d0 56 da - 9b 5b a9 cd cf fb 45 2b
00000030: c9 6a 3c 26 e5 b8 1a 25 - 07 b8 07 22 ed 15 8a 56
00000040: 8b f4 30 f2 28 fc 6b f1 - bf a4 3e 87 f0 be d6 1c
00000050: 33 92 b9 5e d1 b7 20 8c - 92 02 cb e5 26 45 02 1d
```

Default Generator:

```
00000000: 90 f0 09 78 cc 23 79 a8 - 6c 23 a8 65 e0 dc 0f 6d
00000010: fb a7 26 e8 63 0a 21 67 - 5a f8 0f 59 84 09 5c da
00000020: ef af af fc d2 5f 83 e2 - a7 27 05 34 17 94 1a 4f
00000030: b2 87 76 97 e7 48 43 db - 62 29 70 9e 7f eb 2c 6e
00000040: 5d 25 1d a1 65 f0 b4 e6 - 47 4d 25 23 0b 20 b9 93
00000050: 27 f0 56 12 5a 97 f6 c5 - 31 b6 19 fc 67 22 93 f5
```

IP Filtering

IP Filtering is a type of firewall used to control network traffic. The process involves filtering packets received by an interface and deciding whether to forward or to discard them. Filtering is performed for each interface; each Ethernet and WAN interface can have its own set of filters.

Note: IP Filtering is a software option; use the **vers** command to check that it is installed in your router.

When IP filtering is used, the router examines information for each IP packet, such as the source and destination addresses, ports, and protocols, and then screens (filters) the packets based on this information. If the packet matches the conditions of a filter, the router acts as directed by the filter, that is, it accepts, drops or rejects the packet.

Note: To use IP filtering, IP routing must be enabled (see the command **eth ip enable**, page 270).

Built-in Firewall Filters

Although IP filtering offers great flexibility and control, creating the required series of commands may appear complex to a casual user. Therefore, four sets of firewall filters are resident in the flash memory of factory-built routers.

The four sets of filters offer four levels of security: maximum, medium, minimum, and none. You can select and install any of these filter sets from the Set Firewall page of the Web graphic interface. (To learn how to access the Web GUI, refer to the *User Reference Guide* that came in the box in which your router was shipped or find the guide on the Technical Support web site, www.efficient.com.)

The four filter sets are also provided as script files in the *samples* directory on the Installation CD. The file names are *maxsec.txt* (maximum security), *medsec.txt* (medium security), *minsec.txt* (minimum security) and *nosec.txt* (no filters). To execute one of these files from the CLI, first copy the file to the router and then use the **execute** command (see Batch File Command Execution, page 183). For example, to execute the *medsec.txt* file for medium security, enter:

execute medsec.txt

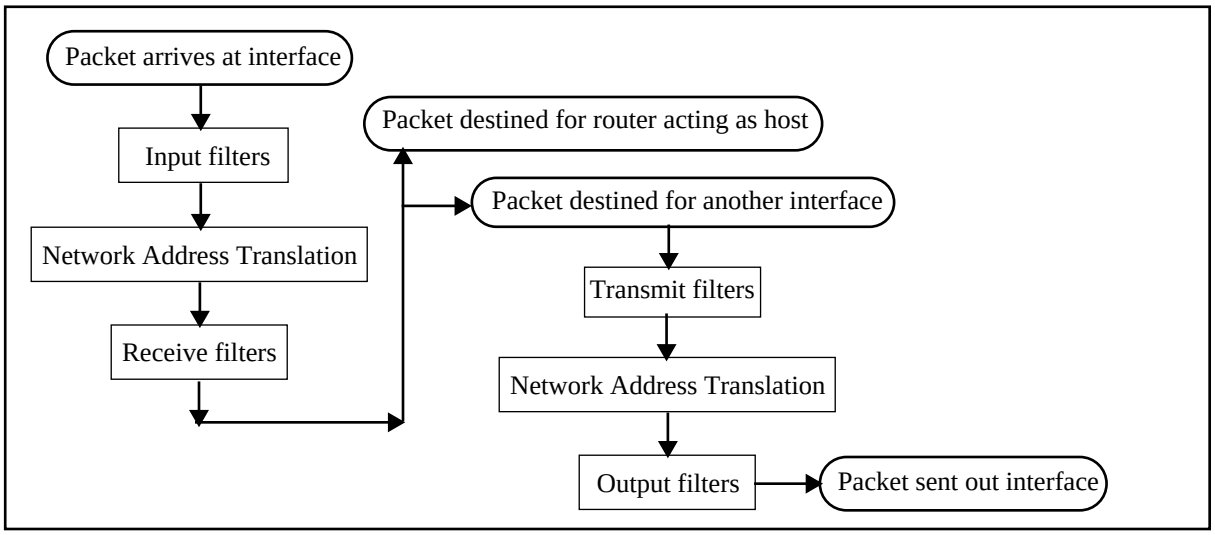
Before executing any script file, you should check its content. Three of the filter sets are listed at the end of this IP Filtering section (Example 3: Maximum Security Firewall, page 132, Example 4: Medium Security Firewall, page 133, and Example 5: Minimum Security Firewall, page 134). Be sure to edit the file to fit your specific configuration and seek expert help if you are not familiar with security.

Note: IP filters only work if IP routing is enabled (see the command **eth ip enable**, page 270).

Filters and Interfaces

Filters screen IP packets; packets are compared to each filter in series. If a packet matches a filter, the filter indicates whether the packet is accepted, dropped, or rejected. If no filter matches the incoming packet, the packet is, by default, accepted.

Filters operate at the interface level. Each interface can have up to four lists of filters associated with it: Input filters, Receive filters, Transmit filters, and Output filters. The following illustrates the filtering process.



1. Input Filters

When a packet arrives at an interface, the router compares the packet to the list of *input* filters. The first filter that matches the packet determines whether the packet is accepted, dropped, or rejected. If no filter matches the packet, the packet is accepted.

If the packet is accepted, the next step is Network Address Translation, if NAT is enabled for the input interface. For more information on Network Address Translation, see Network Address Translation (NAT), page 95.

2. Receive Filters

The router next compares the packet to the list of *receive* filters for this interface. Again, the first filter in the list that matches the packet determines whether the packet is accepted, dropped, or rejected. If no filter matches the packet, the packet is accepted.

Receive filters are applied before the packet destination is determined by the routing table. The packet may be destined for the router itself and/or destined for output to another interface.

Note: If Network Address Translation is disabled, the Receive filter list is checked immediately after the Input filter list. In this case, identical Input and Receive filters have the same effect (see the examples at the end of the IP Filtering section.)

3. Transmit Filters

If the packet is destined for another interface, the router compares the packet to the list of *transmit* filters for this interface. The first filter that matches the packet determines whether the packet is accepted, dropped, or rejected. If no filter matches the packet, the packet is accepted.

If the packet is accepted, Network Address Translation is performed, if NAT is enabled for the output interface.

4. Output Filters

Finally, the router compares the packet to the list of *output* filters for this interface. The first filter that matches the packet determines whether the packet is accepted, dropped, or rejected. If no filter matches the packet, the packet is accepted.

The packet, if accepted, is then sent out the interface.

Note: If Network Address Translation is disabled, the Output filter list is checked immediately after the Transmit filter list. In this case, identical Transmit and Output filters have the same effect

Filter Actions

A filter action can be applied to a packet at each of the four filtering points (Input, Receive, Transmit, and Output). If, at that point, a given filter is the first filter in the list to match that packet, the action specified by that filter determines the fate of the packet. The possible filter actions are:

Accept The router lets the packet proceed for further processing.

Drop The router discards the packet.

Reject The router sends an ICMP REJECT (Internet Control Management Protocol) to reject the packet.

Pass to IPSec Two actions—**inipsec and outipsec**—pass the packet to IPSec for further processing. The **inipsec** action is for packets coming from the other IPSec gateway; it passes the packet to IPSec for decrypting. The **outipsec**action is for packets coming from the local protected network; it passes the packet to IPSec so it can be encrypted and sent to the other IPSec gateway.

Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE, rather than your own filters, to manage your IP security (see IPSec (Internet Protocol Security), page 149).

IP Filter Commands

To define and manage IP filters on an Ethernet interface, use the command **eth ip filter**. To define and manage IP filters on the remote interface, use the command **remote ipfilter**. For more information on these commands, see **eth ip filter**, page 270 and **remote ipfilter**, page 300.

ICMP Redirect

IP filters of Input type are checked *before* the IP packet is redirected by ICMP. This could adversely affect local LANs that use ICMP redirect to dynamically learn IP routes. IP filters of Input type are checked *before* the IP packet is sent to the router itself as a host.

Filter Examples

Example 1: Input Filters Vs. Receive Filters

The following commands add a filter to the beginning of the *Input* filter list.

```
remote ipfilter insert input drop -p tcp -dp 23 internet
save
```

When used, the input filter matches any packet for remote interface **internet** that has protocol **TCP** and destination port **23**. The packets are checked before Network Address Translation, if any; any packets that match the filter are dropped. Thus, this filter stops *any* attempt by a host coming from the remote internet from sending an IP packet to the Telnet port. The router does not see the packet, and the packet is not forwarded.

Consider, next, the following commands that add a filter identical to the above filter to the beginning of the *Receive* filter list:

```
remote ipfilter insert receive drop -p tcp -dp 23 internet
save
```

In the following cases, the Receive filter has the same effect as the Input filter:

- If Network Address Translation is disabled.
- If Network Address Translation is enabled and the Telnet *public* port is mapped to the Telnet *private* port by a **remote addserver** command, such as the following:

  ```
  remote addserver 10.0.1.1 tcp telnet internet
  ```

However, the Receive filter does not have the same effect as the Input filter in the following case:

- If Network Address Translation is enabled and another public port is mapped to the Telnet private port. For example, the following command maps the public port 2000 to the Telnet private port:

  ```
  remote addserver 10.0.1.1 tcp 2000 2000 telnet internet
  ```

  In this case, Network Address Translation would translate the packets with port 2000 to the Telnet port and the Receive filter would drop those packets.

For more information, see Network Address Translation (NAT), page 95 and **remote addserver**, page 293.

Example 2: Filters That Allow Traffic To, But Not Through

Suppose you wanted to allow Telnet packets destined for the router itself, but drop any Telnet packets destined for another interface. This requires two filters. The first filter allows Telnet traffic to the IP address of the router (in this example, 10.0.1.1). The second filter drops all other Telnet traffic.

```
remote ipfilter append input accept -p tcp -dp 23 -da 10.0.1.1 internet

remote ipfilter append input drop -p tcp -dp 23 internet
```

The filter order is important; packets are compared to filters in the order that the filters appear in the filter list. Any Telnet packet that doesn't match the first filter is dropped by the second filter. Thus, command order is important because each of these commands appends its filter to the end of the list.

Example 3: Maximum Security Firewall

The following lists the filters installed when you request maximum security via the graphic interface (file `maxsec.txt`).

```
# For DSL routers
# Allow protocols: HTTP, FTP, DNS, L2TP

# Flush all existing filters
remote ipfilter flush input internet
remote ipfilter flush output internet
```

```
remote ipfilter flush transmit internet
remote ipfilter flush receive internet
eth ip filter flush input
eth ip filter flush output
eth ip filter flush transmit
eth ip filter flush receive

# HTTP from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 80 internet
remote ipfilter insert output accept -p tcp -dp 80 internet

# DNS from LAN to WAN will be accepted
remote ipfilter insert input accept -p udp -sp 53 internet
remote ipfilter insert output accept -p udp -dp 53 internet

# FTP from LAN to WAN accepted
remote ipfilter insert input accept -p tcp -sp 20:21 internet
remote ipfilter insert output accept -p tcp -dp 20:21 internet

# FTP WAN TO LAN accepted
remote ipfilter insert input accept -p tcp -dp 20:21 internet
remote ipfilter insert output accept -p tcp -sp 20:21 internet

# L2TP
remote ipfilter insert input accept -p udp -sp 1701 internet
remote ipfilter insert output accept -p udp -dp 1701 internet

# Deny anything not listed above
remote ipfilter append input drop internet
remote ipfilter append output drop internet

# Watch the results
remote ipfilter watch on internet
save
```

Example 4: Medium Security Firewall

The following lists the filters installed when you request medium security via the graphic interface (file `medsec.txt`).

```
# For DSL routers
# Allow protocols: ICMP, Telnet, SSL, HTTP, FTP, DNS, L2TP, SMTP and POP3

# Flush all existing filters
remote ipfilter flush input internet
remote ipfilter flush output internet
remote ipfilter flush transmit internet
remote ipfilter flush receive internet
eth ip filter flush input
eth ip filter flush output
eth ip filter flush transmit
eth ip filter flush receive

# Allow ICMP replies, requests, and errors from the WAN
remote ipfilter insert input accept -p icmp -sp 0 internet
remote ipfilter insert input accept -p icmp -sp 3 internet
remote ipfilter insert input accept -p icmp -sp 8 internet
remote ipfilter insert input accept -p icmp -sp 11 internet
```

```
# Allow ICMP ECHO REPLY, REQUEST to the WAN
remote ipfilter insert output accept -p icmp -sp 0 internet
remote ipfilter insert output accept -p icmp -sp 8 internet

# Telnet from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 23 internet
remote ipfilter insert output accept -p tcp -dp 23 internet

# SSL accepted
remote ipfilter insert input accept -p tcp -sp 443 internet
remote ipfilter insert output accept -p tcp -dp 443 internet

# HTTP from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 80 internet
remote ipfilter insert output accept -p tcp -dp 80 internet

# FTP from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 20:21 internet
remote ipfilter insert output accept -p tcp -dp 20:21 internet

# DNS from LAN to WAN will be accepted
remote ipfilter insert input accept -p udp -sp 53 internet
remote ipfilter insert output accept -p udp -dp 53 internet

# L2TP will be accepted
remote ipfilter insert input accept -p udp -sp 1701 internet
remote ipfilter insert output accept -p udp -dp 1701 internet

# E-mail - SMTP and POP3 requests from LAN to WAN accepted
remote ipfilter insert input accept -p tcp -sp 25 internet
remote ipfilter insert output accept -p tcp -dp 25 internet

remote ipfilter insert input accept -p tcp -sp 110 internet
remote ipfilter insert output accept -p tcp -dp 110 internet

# Drop all packets
remote ipfilter append input drop internet
remote ipfilter append output drop internet

# Watch the results
remote ipfilter watch on internet
save
```

Example 5: Minimum Security Firewall

The following lists the filters installed when you request minimum security via the graphic interface (file `minsec.txt`).

```
# Minimum security script for DSL routers
# For remote commands, input filters apply to traffic from the WAN, and
# output filters apply to traffic to the WAN.

# Flush all existing filters
remote ipfilter flush input internet
remote ipfilter flush output internet
remote ipfilter flush transmit internet
remote ipfilter flush receive internet
```

```
eth ip filter flush input
eth ip filter flush output
eth ip filter flush transmit
eth ip filter flush receive

# Allow ICMP replies, requests, and errors from the WAN
remote ipfilter insert input accept -p icmp -sp 0 internet
remote ipfilter insert input accept -p icmp -sp 3 internet
remote ipfilter insert input accept -p icmp -sp 8 internet
remote ipfilter insert input accept -p icmp -sp 11 internet

# Allow ICMP ECHO REPLY, REQUEST to the WAN
remote ipfilter insert output accept -p icmp -sp 0 internet
remote ipfilter insert output accept -p icmp -sp 8 internet

# Telnet from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 23 internet
remote ipfilter insert output accept -p tcp -dp 23 internet

# SSL accepted
remote ipfilter insert input accept -p tcp -sp 443 internet
remote ipfilter insert output accept -p tcp -dp 443 internet

# HTTP from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 80 internet
remote ipfilter insert output accept -p tcp -dp 80 internet

# FTP from LAN to WAN will be accepted
remote ipfilter insert input accept -p tcp -sp 20:21 internet
remote ipfilter insert output accept -p tcp -dp 20:21 internet

# DNS from LAN to WAN will be accepted
remote ipfilter insert input accept -p udp -sp 53 internet
remote ipfilter insert output accept -p udp -dp 53 internet

# L2TP will be accepted
remote ipfilter insert input accept -p udp -sp 1701 internet
remote ipfilter insert output accept -p udp -dp 1701 internet

# E-mail - SMTP and POP3 requests from LAN to WAN accepted
remote ipfilter insert input accept -p tcp -sp 25 internet
remote ipfilter insert output accept -p tcp -dp 25 internet

remote ipfilter insert input accept -p tcp -sp 110 internet
remote ipfilter insert output accept -p tcp -dp 110 internet

# Allow SSH from the WAN
remote ipfilter insert input accept -p tcp -dp 22 internet
remote ipfilter insert output accept -p tcp -sp 22 internet

# Allow NETBIOS connections from specific sources on the WAN
# Allow NETBIOS requests from our network
remote ipfilter insert input accept -p tcp -dp 137:139 internet
remote ipfilter insert input accept -p udp -dp 137:139 internet
remote ipfilter insert output accept -p tcp -sp 137:139 internet
remote ipfilter insert output accept -p tcp -dp 137:139 internet
remote ipfilter insert output accept -p udp -dp 137:139 internet

# finger
```

```
remote ipfilter insert output accept -p tcp -sp 1024:65535 -dp 79 internet

# POP2 tcp/udp
remote ipfilter insert output accept -p tcp -sp 1024:65535 -dp 109 internet

# NNTP tcp
remote ipfilter insert output accept -p tcp -sp 1024:65535 -dp 119 internet

# IMAP2 tcp/udp
remote ipfilter insert output accept -p tcp -sp 1024:65535 -dp 143 internet

# certain other non-privileged ports to non-privileged ports
remote ipfilter insert output accept -p tcp -sp 1024:65535 -dp 1024:65535 internet

# Allow NTP, who, Kali, CuSeeMe out to the WAN

# NTP
remote ipfilter insert transmit accept -p udp -dp 123 internet
remote ipfilter insert receive accept -p udp -sp 123 internet

# who
remote ipfilter insert input accept -p udp -sp 513 -dp 1024:65535 internet
remote ipfilter insert output accept -p udp -dp 513 -sp 1024:65535 internet

remote ipfilter insert input accept -b -p udp -sp 2213 -dp 1024:65535 internet
remote ipfilter insert output accept -b -p udp -dp 2213 -sp 1024:65535 internet

remote ipfilter insert input accept -p udp -sp 6666 -dp 1024:65535 internet
remote ipfilter insert output accept -p udp -dp 6666 -sp 1024:65535 internet

remote ipfilter insert input accept -p udp -sp 7648 -dp 7648 internet
remote ipfilter insert output accept -p udp -dp 7648 -sp 7648 internet

# RealAudio
remote ipfilter insert input accept -p udp -dp 7070 internet
remote ipfilter insert output accept -p udp -sp 7070 internet

# traceroute
remote ipfilter insert input accept -p udp -sp 1024:65535 -dp 33434:33500 internet
remote ipfilter insert output accept -p udp -sp 1024:65535 -dp 33434:33500 internet

### Deny any other traffic
remote ipfilter append input drop internet
remote ipfilter append output drop internet

# Turn on ip filter watch for debugging
remote ipfilter watch on internet

save
```

L2TP Tunneling — Virtual Dial-Up

This section has four parts:

- The *Introduction* provides a general overview of L2TP tunneling.
- The *L2TP Concepts* section explains LNS, L2TP client, LAC, dial user, tunnels, and sessions.
- *Configuration* describes preliminary configuration steps and verification steps and lists commands associated with the configuration of L2TP and PPP sessions.
- The *Sample Configurations* section provides two examples with step-by-step instructions: a simple L2TP client configuration example and a complete LNS and L2TP client configuration example.

  The installation CD also contains sample configuration files. These files can be edited for your installation and copied to the router using TFTP or the Windows Quick Start application. For more information on TFTP use, see Batch File Command Execution, page 183.

Advantages of Tunneling

L2TP (Layer 2 Tunneling Protocol) is used to forward a PPP link from a remote site to a corporate site across the Internet, thus creating virtual paths called tunnels. Because tunneling involves encapsulating data, packets can be transported across networks using different protocols. The advantages for tunneling the PPP protocol are listed below:

- Different network protocols such as NetBEUI, IPX, and Appletalk can be transported through the Internet using a tunnel. The protocol packets are encapsulated and routed across the network through the Internet.
- Tunnels provide a way to reduce costs and complexity associated with remote dial-up networking by using a local ISP: users connect to the remote site by dialing into their local ISP and letting the Internet handle the long-distance connections, thus avoiding long-distance phone charges.
- Tunneling PPP allows compression of data through the entire tunnel, which translates into greater throughput.
- By allowing encryption over the PPP link, L2TP contributes to more secure networks over the Internet.
- Remote users can access the company network, even if there is a company firewall (provided, of course, that tunnels can come through the firewall).

Note: This feature can interoperate with any vendor that supports L2TP - Draft II.

L2TP Concepts

This section defines the major L2TP concepts and illustrates them with L2TP client examples. It also describes the creation and destruction of tunnels and sessions.

Definitions

An L2TP tunnel is created between an L2TP client and an L2TP network server (LNS). The client and server control the tunnel using the L2TP protocol.

L2TP Network Server (LNS)

Point where the call is actually managed and terminated (e.g., within a corporate network).

L2TP Access Concentrator (LAC)

Physical hardware (such as a router) used for placing and receiving phone calls.

Dial User

The remote system or router that is either placing the call to the LAC or receiving the call from the LAC. The dial user does not actually dial in to the LNS or receive a call from the LNS, since this is a virtual connection. The dial user is one end of a PPP session. The LNS is the other end of the PPP session.

L2TP Client

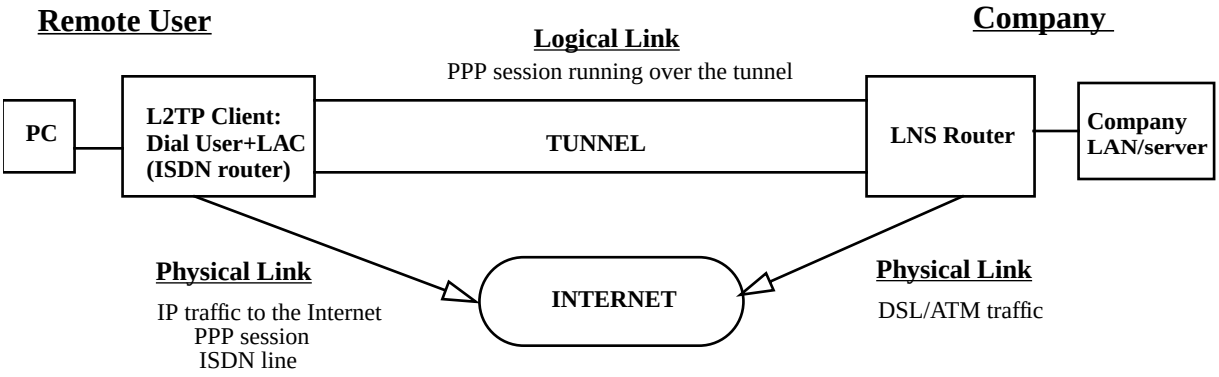
The dial user and LAC combined in the same hardware device. In this case, the PPP session is between the LAC and the LNS.

As shown in the following illustration, an L2TP client is used to tunnel a PPP session between a small office (our router) and a corporate office through the Internet.

L2TP Client Illustration

The tunnel uses UDP/IP traffic as the transport medium over IP. This implementation of L2TP as illustrated below shows a tunnel from a remote user's perspective.

Note: There is one PPP session over ISDN and another PPP session over the tunnel.



LNS and L2TP Client Relationship

The LNS acts as the supervising system. The L2TP client acts both as the dial user and the LAC.

One end of the tunnel terminates at the L2TP client. The other end of the tunnel terminates at the LNS.

One end of the PPP session going through the tunnel terminates at the L2TP client acting as the dial user; the other end terminates at the LNS.

Tunnels

Tunnels are virtual paths that exist between an L2TP client and an L2TP server.

An L2TP server can communicate simultaneously with more than one L2TP client.

An L2TP client can communicate simultaneously with more than one L2TP server.

Some L2TP implementations including the one discussed in this section allow the *same* router to act as *both* an L2TP client and an L2TP server simultaneously, if so configured.

Caution: Verify that the IP address of the other end of the tunnel is correctly routed through the right, local interface/remote and will not appear to be routed through the tunnel. An attempt to route the tunnel endpoint within itself will fail.

Sessions

Sessions can be thought of as switched virtual circuit “calls” carried within a tunnel and can only exist within tunnels. One session carries one “call”. This “call” is one PPP session. Multiple sessions can exist within a tunnel. The following briefly discusses how sessions are created and destroyed.

- #### Session creation

  Traffic destined to a remote entry (located at the end of the tunnel) initiates a tunnel session. When the L2TP client wishes to establish a session to an LNS, the L2TP client assumes the role of a LAC and sends control packets containing incoming call information to the LNS over the tunnel.

- #### Session destruction

  A tunnel session automatically times out after the data session stops. When instructed to destroy a session, the L2TP client closes any PPP session associated with that session. The L2TP client may also send control messages to the LNS indicating that the L2TP client wishes to end the PPP session.

  When the LNS wants to hang up the call, it sends control messages destroying the session.

Configuration

Preliminary Steps to Configure a Tunnel

The following logical steps should be considered before configuring a tunnel:

1. Decide if the router should act as an L2TP Client or LNS.
2. Decide if one side or both sides of the connection should be allowed to initiate a tunnel.
3. Create the L2TP Tunnel Entry with these characteristics:
   - The host name of the L2TP client
   - The host name of the L2TP network server
   - A Tunnel CHAP secret (both sides of the connection must use the same secret)

- The IP address of the other party must be provided to the initiating side of the tunnel
- Type of flow control (pacing, sequence numbers, or none)

4. Create a remote entry for the PPP session. Associate the remote entry with the Tunnel.

Verification Steps

1. Verify that the IP address of the other end of the tunnel is correctly routed through the right, local interface/remote and will not appear to be routed through the tunnel. An attempt to route the tunnel endpoint within itself will fail.

2. Try to establish IP connectivity (using the **ping** or **tracert** commands).

   a. "Pinging" from the L2TP client or LNS to the opposite tunnel endpoint should succeed (this tests the tunnel path).

   b. "Pinging" from a tunnel endpoint IP address to an IP address within the tunnel will probably fail due to the existence of the IP firewall.

Configuration Commands

L2TP configuration commands are used to configure:

- Tunnels
- The PPP session

• Commands to configure tunnels

For additional information, see L2TP — Virtual Dial-Up Configuration Commands, page 363.

L2TP tunnel entry:

l2tp add <*TunnelName*>

The remote tunnel host name:

l2tp set remoteName <*name*> <*TunnelName*>

The local tunnel host name:

l2tp set ourTunnelName <*name*> <*TunnelName*>

CHAP secret:

l2tp set CHAPSecret <*secret*> <*TunnelName*>

Tunnel authentication:

l2tp set authen on | off <*TunnelName*>

Type of L2TP support for tunnel:

Configure the entry to act as a L2TP client,, an L2TP network server (LNS), or as both a LAC and an LNS, or the entry can be disabled.

l2tp set type all | lns | l2tpclient |disabled <*TunnelName*>

Remote tunnel IP address:

l2tp set address <*ipaddr*> <*TunnelName*>

Note: Verify that the IP address of the other end of the tunnel is correctly routed. It should not be routed through the tunnel itself, but over a physical link.

You may also specify the source IP address for the tunnel as an address other than the WAN interface IP address, such as the Ethernet IP address.

l2tp set ourAddress <*ipaddr)*> <*TunnelName*>

Our PPP system name and secret/password:

The following commands specify the router's name and password/secret for authentication purposes on a per-tunnel basis.

l2tp set ourSysName <*name*> <*TunnelName*>
l2tp set ourPassword <*password*> <*TunnelName*>

Other commands:

Commands are also available to delete a tunnel, close a tunnel, or set up advanced L2TP configuration features such as traffic performance fine-tuning (see L2TP — Virtual Dial-Up Configuration Commands, page 363).

- **Commands for PPP Session Configuration**

Two commands are used to extend a PPP link from a remote site to a corporate site across the Internet and establish a tunnel. For more information, see L2TP — Virtual Dial-Up Configuration Commands, page 363.

remote setLNS <*TunnelName*> <*remoteName*>
remote setl2tpclient <*TunnelName*> <*remoteName*>

Sample Configurations

Two sample configurations are described in this section:

- A simple configuration. This example describes the information needed to configure one side of the tunnel (the client side).
- A complete configuration. This example describes the information needed to configure both sides of the tunnel (client and server sides).

Simple L2TP Client Configuration Example

This example shows how a telecommuter working at home (client side) can configure his/her router SOHO to tunnel to the company's LAN (server side).

The information given in the Configuration Process section below provides a framework reference for this type of L2TP Client configuration.

- **Assumptions**

In this example, the following information is assumed:

- The server side (the company) has an LNS router connected to the Internet.
- The client side has an existing route to the Internet with the remote “Internet” (refer to the following Note, if you need sample configuration commands).
- IP routing is enabled (refer to the following Note, if you need sample configuration commands).

Note: Below is an example of configuration commands that can be used to enable IP routing and establish a route to the Internet.

```
remote add internet
remote disauthen internet
remote setoursysname name_isp_expects internet
remote setourpass secret_isp_expects internet
remote addiproute 0.0.0.0 0.0.0.0 1 internet
remote setphone isdn 1 5551000 internet
remote setphone isdn 2 5553000 internet
eth ip enable
eth ip address 192.168.254.254 255.255.255.0
```

• Configuration Process

The following sets of questions, answers, and configuration commands specific to the L2TP tunnel and the PPP remote will assist you in configuring the client side router SOHO (also referred to as home router). Note that the server side is referred to as either company router or router at work.

L2TP tunnel configuration

L2TP tunnel-specific questions

1. What is the host name of the router at home that the user is configuring?
2. What is the host name of the company router at work to which the user will tunnel?
3. What is the shared CHAP secret used for tunneling between the home router (client) and the company router (server)?
4. What is the IP address of the company router to which the user will tunnel?

L2TP tunnel answers. For our example, let’s assume the answers to the above tunnel-specific questions are as follows:

1. Home_Router
2. Work_Router
3. Shared_Secret
4. 10.0.0.1

L2TP tunnel configuration commands. These commands would be used to set up the L2TP tunnel information for our example:

```
l2tp add Work_Router
l2tp set ourtunnel Home_Router Work_Router
```

```
l2tp set chapsecret Shared_Secret Work_Router
l2tp set address 10.0.0.1 Work_Router
```

PPP remote configuration

PPP remote-specific questions:

1. What is the home router's name for PPP authentication?
2. What is the home router's secret for PPP authentication?
3. Does the home router need PPP authentication for the remote router (company router)?

   If yes:

   a. What is the remote router's name for PPP authentication?
   b. What is the remote router's secret for PPP authentication?

   If no:

   a. Use the command **remote disauthen** <*remoteName*> where <*remoteName*> is the name used to refer to the company's router.

4. Does the remote router dynamically assign an IP address for this PPP session?

   If *yes*:

   Use IP address translation (NAT)

   If *no* and the home router is to behave as a LAN at home:

   Which IP address and network mask does the home router use for its LAN at home? Use the **eth ip addr** command to set the LAN at home. Do not enable IP address translation (NAT) for the remote (company) router.

   If *no* and the home router is to behave as a host at home:

   Which IP address does it use at home? Assuming an IP address of www.xxx.yyy.zzz, use the command:

   remote setsrcipaddr www.xxx.yyy.zzz 255.255.255.255 <*remoteName*>

   remote setiptranslate on <*remoteName*>

5. Which IP and network addresses does the home router access at work through this PPP session?

PPP remote answers. For our example, let us assume the answers to the above PPP remote-specific questions are as follows:

1. ppp_soho
2. ppp_soho_secret
3. We assume that this router will authenticate the router at work with the following information:

   a) the company router's name is: ppp_work
   b) the company router's PPP secret is: ppp_work_secret

4. We assume that the company's router will dynamically assign an IP address to the home router.

5. 172.16.0.0/255.240.0.0

PPP remote configuration commands. For our example, these commands would be used to set up the PPP remote information for tunneling to work:

```
remote add ppp_work
remote setlns Work_Router ppp_work
remote setpasswd ppp_work_secret ppp_work
remote setiptranslate on ppp_work
remote addiproute 172.16.0.0 255.240.0.0 1 ppp_work

l2tp set oursysname ppp_soho Work_Router
l2tp set ourpassword ppp_soho_secret Work_Router
```

Complete LNS and L2TP Client Configuration Example

The following information and illustration (Figure 1) provide a configuration example of an LNS and L2TP Client.

- ### Assumptions

IP Addresses

The LNS server's LAN IP address is 192.168.100.1 (**LNSserver**) with a mask of 255.255.255.0.

The LNS has a WAN IP address of 192.168.110.1, which is used as the tunnel endpoint.

The LNS connects to the remote **internet**.

The L2TP Client's LAN IP address is 192.168.101.1 (**soho**) with a mask of 255.255.255.0. Additionally, 192.168.101.1 is also the tunnel endpoint within the L2TP client. The router **soho** connects to the remote **isp**.

Secret/password

A shared tunnel secret of "tunnelsecret" will be used.

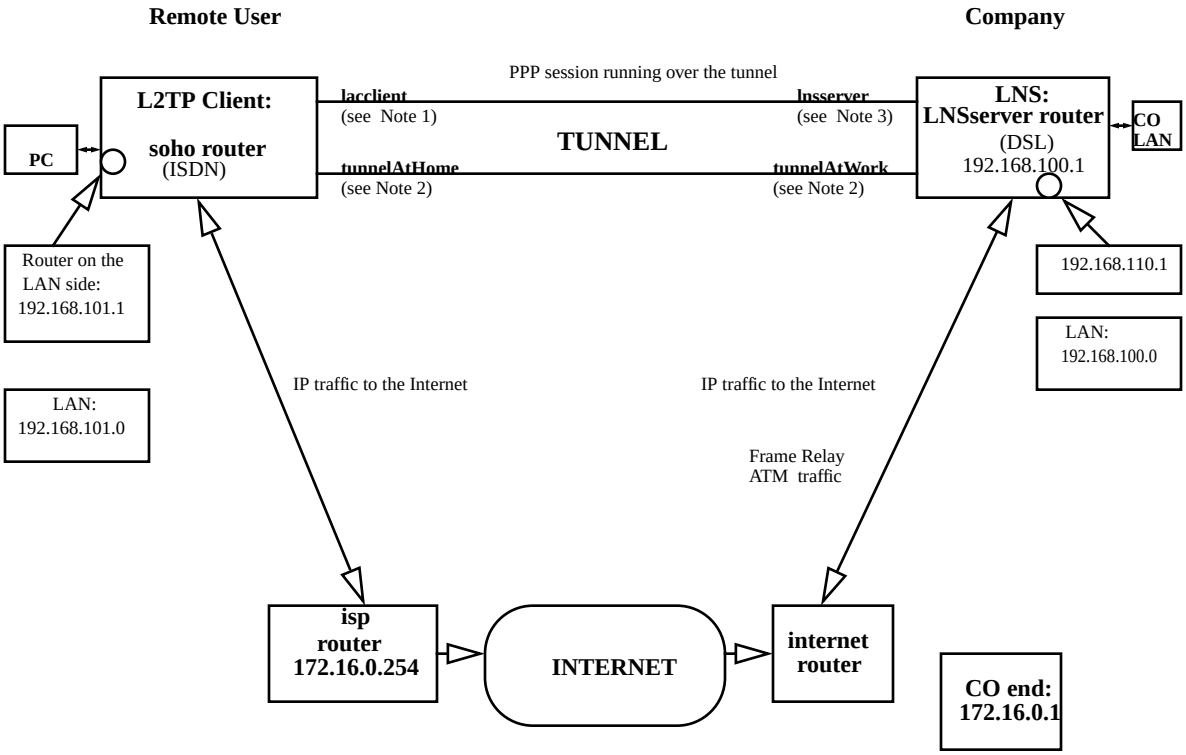
PPP Authentication

The LNS will authenticate the client using PPP. The client will not try to authenticate the LNS using PPP. For PPP authentication, the L2TP client will be known as "lacclient" with a password of "clientpassword".

Tunnel

Only the L2TP client (**soho**) will initiate the tunnel and make the connection. The tunnel is routed through the remote **internet** which is the default route. The LNS server never calls the L2TP client (**soho**).

Figure 1



Note 1: The CHAP secret is “clientPassword”.

Note 2: The CHAP secret is “tunnelSecret”.

Note 3: No CHAP secret is needed; the client does not authenticate the LNS server.

Configuration Process

The following sample scripts list the commands used to configure the routers **soho** (L2TP client), **LNSserver** (LNS), **internet**, and **isp**.

- ### Configuration commands for soho (L2TP client)

Note: soho is an ISDN router.

Define soho:

```
system name soho
system passwd sohopasswd
system msg configured_12/15/98
system securitytimer 60
```

Enable IP routing for soho:

```
eth ip enable
eth ip addr 192.168.101.1 255.255.255.0
```

Set up ISDN parameters:

```
isdn set switch ni1
isdn set dn 5551000 5553000
isdn set spids 0555100001 0555300001
```

Define DHCP settings for DNS servers, domain, wins server:

```
dhcp set value DOMAINNAMESERVER 192.168.100.68
dhcp set value DOMAINNAME efficient.com
dhcp set value WINSSERVER 192.168.100.73
```

Define a remote for the tunnel:

```
remote add lnsserver
remote disauthen lnsserver
remote setoursysname lacclient lnsserver
remote setourpasswd clientpassword lnsserver
remote setLNS tunnelAtWork lnsserver
remote addiproute 192.168.100.0 255.255.255.0 1 lnsserver
```

Define a remote isp:

```
remote add isp
remote setphone isdn 1 5552000 isp
remote setphone isdn 2 5554000 isp
remote disauthen internet remote addiproute 0.0.0.0 0.0.0.0 1 isp
```

Define the tunnel:

```
l2tp add tunnelAtWork
l2tp set chapsecret tunnelsecret tunnelAtWork
l2tp set ourtunnelname tunnelAtHome tunnelAtWork
l2tp set address 192.168.110.1 tunnelAtWork
save
reboot
```

- ## Configuration commands for internet

Note: internet is a DSL router. The router internet establishes a link to the LNS.

Define internet:

```
system name internet
system passwd internet
system msg configured_12/15/98
system securitytimer 60
```

Enable IP routing and add routes:

```
eth ip enable
eth ip addr 172.16.0.1 255.255.255.0
eth ip opt rxdef off
eth ip addroute 192.168.101.1 255.255.255.0 172.16.0.254 1
```

Create a DHCP pool of addresses:

```
dhcp add 172.16.0.0 255.255.255.0
dhcp del 192.168.254.0
dhcp set addr 172.16.0.2 172.16.0.20
```

Set up DSL parameters:

```
sd term co sd speed 1152
```

Define a remote LNSserver

```
remote add lnsserver
remote setauthen chap lnsserver
remote setpasswd serverpassword lnsserver
remote addiproute  192.168.110.1 255.255.255.255 1 lnsserver
remote setprotocol ppp lnsserver
remote setpvc 0*38 lnsserver
save
reboot
```

- ## Configuration commands for isp

Note: isp is an ISDN router. The router soho calls the router isp.

Define isp:

```
system name isp
system passwd isppasswd
system msg configured_12/15/98
system securitytimer 60
```

Enable IP routing:

```
eth ip enable
eth ip addr 172.16.0.254 255.255.255.0
```

Add a route to the other end of internet:

```
eth ip defgate 172.16.0.1
eth ip opt txdef off
```

Disable DHCP:

```
dhcp disable all
```

Set up ISDN parameters:

```
isdn set switch ni1
isdn set dn 5552000 5554000
isdn set spids 0555200001 0555400001
```

Define a remote (soho):

```
remote add soho
remote setauthen chap soho
remote setpassw sohopasswd soho
remote setphone isdn 1 5551000 soho
remote setphone isdn 2 5553000 soho
remote addiproute  192.168.101.0 255.255.255.0 1 soho
save
reboot
```

- ## Configuration commands for LNSserver

  Note: LNSserver is a DSL router.

  Define LNSserver:

```
system name lnsserver
system passwd serverpassword
system msg Script_for_LNS_called_HQ
system securitytimer 60
```

  Enable IP routing:

```
eth ip enable
eth ip addr 192.168.100.1 255.255.255.0
```

  Define DHCP settings for DNS servers, domain:

```
dhcp set value domainname efficient.com
dhcp set value domainnameserver 192.168.100.68
```

  Set up DSL parameters:

```
sd speed 1152
```

  Define a remote for the Tunnel:

```
remote add lacclient
remote setpass clientpassword lacclient
remote setLAC tunnelAtHome lacclient
remote setauthen chap lacclient
remote addiproute 192.168.101.0 255.255.255.0 1 lacclient
```

  Define a remote (internet):

```
remote add internet
remote setphone isdn 1 5552000 internet
remote setphone isdn 2 5554000 internet
remote setauthen chap internet
remote setpasswd internet internet
remote addiproute 0.0.0.0 0.0.0.0 1 internet
remote setsrcipaddr 192.168.110.1 255.255.255.255 internet
remote addiproute 192.168.101.1 255.255.255.255 1 internet
remote setprotocol ppp internet
remote setpvc 0*38 internet
```

  Define the actual tunnel:

```
l2tp add tunnelAtHome
l2tp set chapsecret tunnelsecret tunnelAtHome
l2tp set ourtunnelname tunnelAtWork tunnelAtHome
save
reboot
```

IPSec (Internet Protocol Security)

Note: IPSec security is a software option for your router. The option becomes available after purchase and installation of the software option key (see Software Option Keys, page 124). Use the **vers** command to check that IPSec is available on your router.

Note: Almost all IPSec capabilities can be selected using the graphic interface. However, a few policy selections are available only through the Command Line Interface described in this section. (The graphic interface is described in the *User Reference Guide* that came with your router and is also available on the web site www.efficient.com.)

IPSec is an open standard that defines optional authentication and encryption methods at the IP packet level. It is a true network layer protocol that provides authentication, privacy, and data integrity. Its protocol suite is comprised of:

- **ESP** (Encapsulated Security Payload)—a security protocol that completely encapsulates and optionally encrypts and/or authenticates user data.
- **AH** (Authentication Header)—a security protocol that authenticates each data packet.
- **IKE** (Internet Key Exchange)—a security protocol used to establish a shared security policy and authenticated keys before an IPSec data transfer begins.

IPSec sessions are initiated through Security Associations (SAs), which allow peers to negotiate a common set of security attributes. In a nutshell, IPSec assures source authenticity, data integrity and confidentiality of IP packets, providing the level of security required by Virtual Private Networks (VPNs).

IPSec can be used in conjunction with L2TP (see L2TP Tunneling — Virtual Dial-Up, page 137). IPSec offers greater security than L2TP, but it does not support as many network protocols. However, bridged and lower layer protocol traffic may be transmitted across an IPSec network if packets are first encapsulated by L2TP, and then by IPSec.

IPSec does not require modification of individual applications or devices for secure data transport. Although it does require global IP addresses for all peers, Network Address Translation (NAT) may be used with IPSec. (See Network Address Translation (NAT), page 95.)

Transport and Tunnel Encapsulation Modes

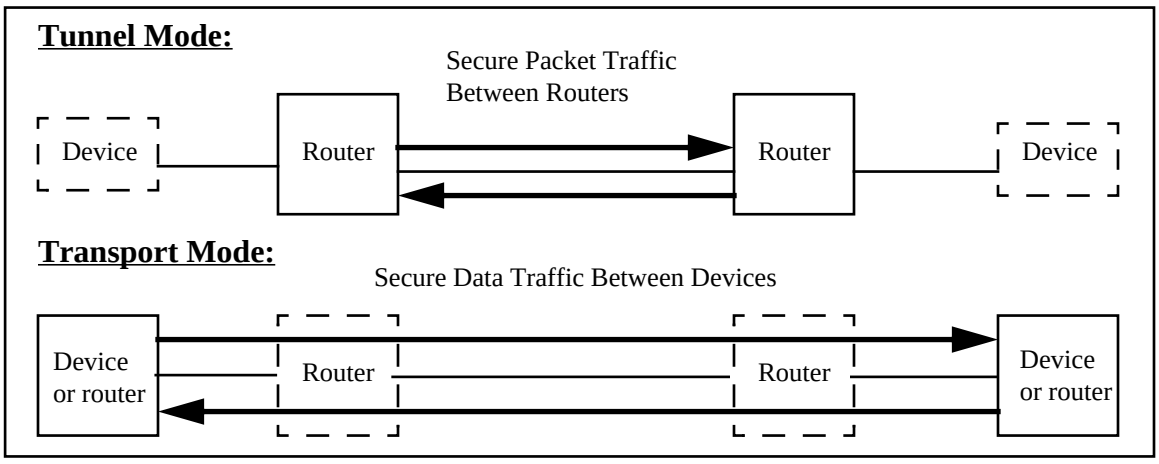
IPSec has two encapsulation modes: transport mode and tunnel mode. Transport mode protects traffic between two nodes or peers (the endpoints of the communication). Tunnel mode protects traffic between peers and/or gateways, such as traffic on a VPN or on any other connection where one or both of the endpoints might *not* be IPSec systems.

The router supports both IPSec encapsulation methods. It can serve as the endpoint of a *tunnel* mode connection or as the endpoint of a *transport* mode connection. Also, while operating in tunnel mode, the router does allow transport mode traffic to flow through it.

Tunnel mode is the default encapsulation mode for the router. It is used when the IPSec packet comes from either another device or from the encrypting device. In tunnel mode, the IP header is encrypted as part of the payload, and an entirely new IP header is added to the packet. Tunnel mode prevents network traffic analysis. A network attacker could determine the tunnel endpoints (the gateway addresses), but not the true source and destination of the tunneled packets, even if they are the same as the tunnel endpoints

Transport mode is used when the IPSec packet originates in the encrypting device. In transport mode, only the payload (data portion) of each IP packet is encapsulated and/or encrypted. An IPSec header is inserted between the IP header and the upper layer protocol header.

The router should be configured for transport mode when a client is communicating directly with the router. For example, use transport mode when a remote user wants to access the HTML setup pages or Telnet into the router. It can also be used for L2TP over IPSec. The routers at either end of the L2TP tunnel do both the IPSec and L2TP encapsulations so the routers can use transport mode for communications.



ESP and AH Security Protocols

An IPSec connection must use either the AH or the ESP security protocol.The protocol selected determines the encapsulation method used. In addition, the protocol also determines whether encryption may be performed. If the AH protocol is selected, only packet authentication can be performed, *not* encryption. If the ESP protocol is selected, it can perform encryption, authentication, or both encryption and authentication.

If ESP encryption is selected, ESP automatically encrypts the data portion (payload) of each packet using the chosen encryption method, DES (56-bit keys) or 3DES (168-bit keys).

Caution: Restrictions may exist on the export of the DES and 3DES encryption options outside the United States or Canada.

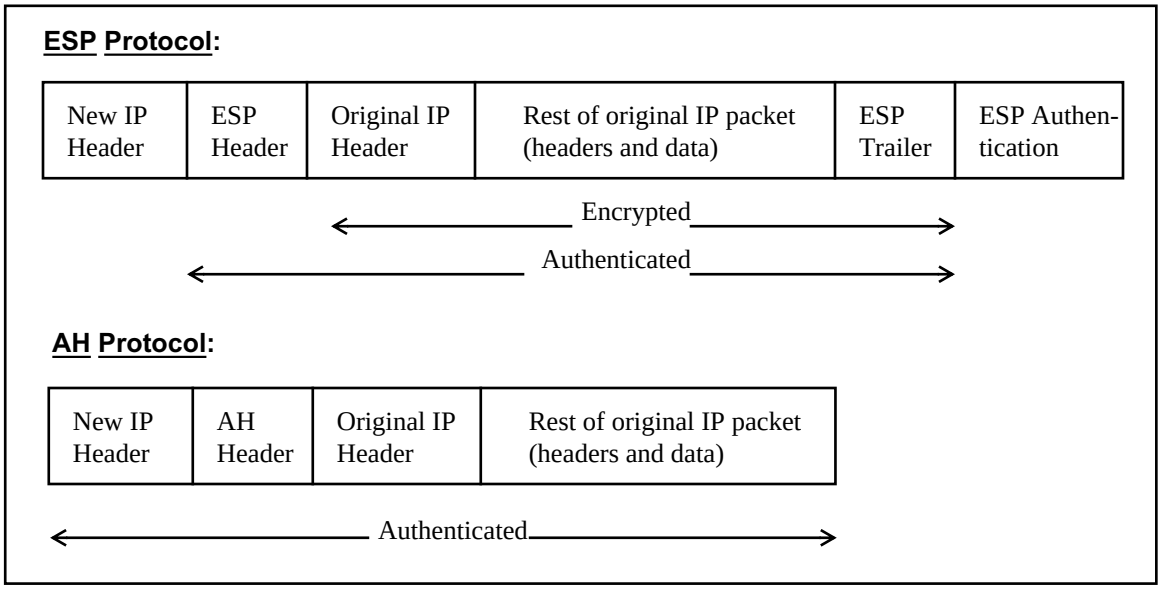
Although encryption cannot be specified for individual applications, a server could be partitioned to achieve the same effect. Given that packets can be encrypted using any combination of security association (SA), protocol, source port, and destination port, you could specify that traffic to and from one database be encrypted while allowing unencrypted traffic to pass freely to and from other databases on the server.

Both the ESP and AH protocols support authentication and replay detection. Replay detection uses sequence numbers to reject old or duplicate packets. The packet is authenticated using a message digest derived from either of two hashing algorithms—SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5).

The ESP protocol can authenticate the data origin and data integrity; it does not authenticate the entire packet. More specifically, the message digest is inserted following, not before, the payload. Both the message digest and payload are sandwiched between the ESP header and ESP trailer.

The AH protocol can perform packet authentication. The AH header protocol defines authentication methods for both the packet's outer IP header and its payload. Unlike ESP authentication, the message digest is inserted in front of the payload.

The following figure shows the transformed IP packet after the ESP or AH protocol has been applied in tunnel mode.



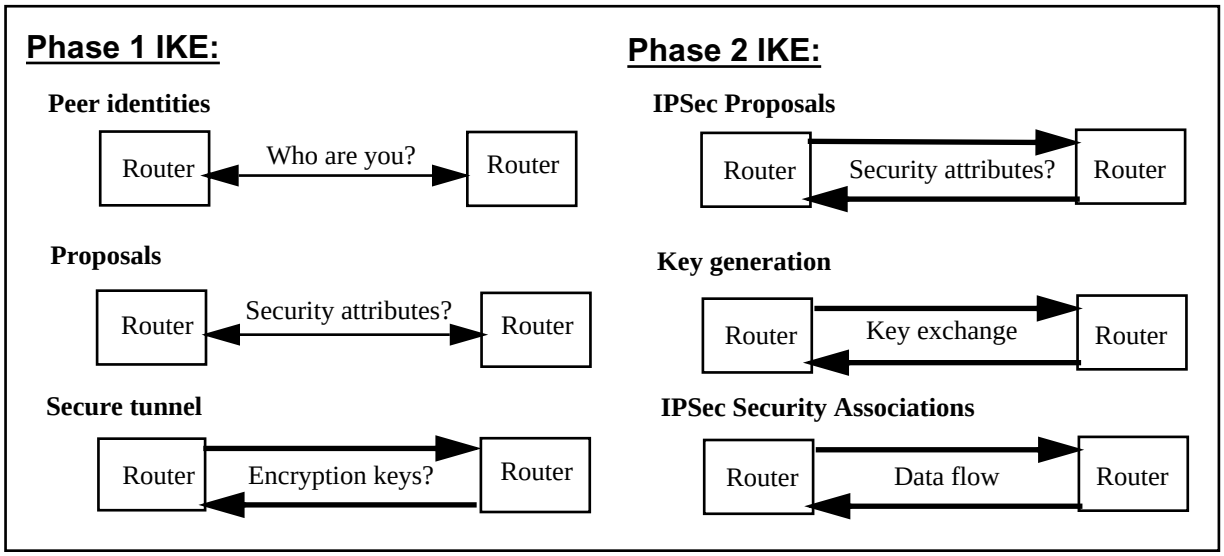
IKE Management

Internet Key Exchange (IKE) management makes encryption key exchange practical, even in large networks where there are many unknown intermediate links between sending and receiving nodes. Unlike protocols that allow only one key exchange per session, IKE can generate and transfer multiple keys between peers during a single tunnel session. Users may specify the duration for which keys are valid. This dynamic type of Diffie-Hellman key exchange greatly reduces the chances of a network attacker finding an entry into a tunnel.

If you wish, you may also select Perfect Forward Secrecy (PFS) to increase the security of the key exchange. PFS ensures that the compromise of a single key permits access to only data protected by that particular key. However, PFS requires use of a Diffie-Hellman group for each rekey, adding overhead to the process and causing IKE to run more slowly. Thus, PFS is not always desirable.

Because VPN users are likely to be using a variety of protocols, a common set of security attributes must be negotiated at the beginning of any tunnel session. Phase 1 IKE is responsible for negotiating these security attributes and establishing peer identities. A secure tunnel for the exchange of encryption keys is also created

during this phase. Phase 2 IKE then exchanges proposals for IPSec security attributes, generates the encryption keys and sets up IPSec Security Associations (SAs) for moving user data.

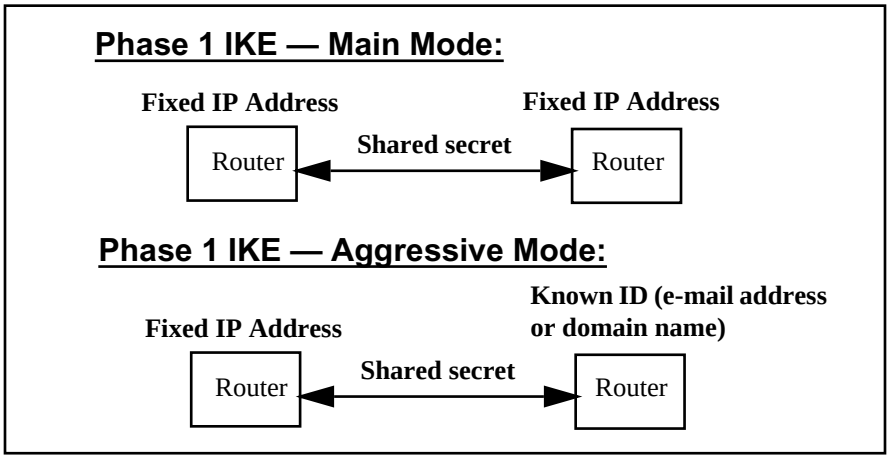


Main Mode and Aggressive Mode

The router supports two Phase 1 IKE modes: main mode and aggressive mode. These modes apply only to the Phase 1 negotiations, not to the ensuing data transmission.

Main mode is used when both source and destination IP addresses are known. In main mode, only two options require definition initially—the remote peer IP address and the shared secret.

Aggressive mode is used when either the source or destination IP address could change, as with a remote modem or DSL connection. In aggressive mode, additional information must be specified at the beginning of a session. This additional information includes the remote gateway's IP address, the local and remote peer IDs, and an ID type. This information is checked against the router's Security Association (SA) database. If a match is found, a tunnel session can be established.



Additional IKE Settings

In addition to the peer identification and shared secret described earlier, IKE requires that the router be configured with the following information:

- Session authentication
- Phase 1 IKE message authentication
- Phase 1 IKE message encryption
- One of the following for each IKE proposal:
  —IPSec AH packet authentication
  —IPSec ESP data authentication
  —IPSec ESP data encryption
  —IPSec ESP data authentication *and* data encryption
- Diffie-Hellman key generation group
- IPSec policy (filter) setup
- Policy and peer associations
- Policy and proposal associations

Security Associations (SAs)

A Security Association (SA) is an instance of security policy and keying material applied to a data flow. Both IKE and IPSec use SAs. An IPSec SA is unidirectional, applying to only one direction of data flow. An IKE SA is bi-directional, and thus, only one IKE SA is needed for a secure connection.

After an IKE SA is established, any number of IPSec SAs may be created. Although IPSec SAs can be configured manually, most networks rely on IKE to set them up. IKE negotiates and establishes SAs on behalf of IPSec. SAs are negotiated between the two endpoints of the tunnel and contain information on sequence numbering for anti-replay.

IPSec SAs are unidirectional so a set of SAs is needed for a secure connection. For each security protocol used, one SA is needed for each direction (inbound and outbound). Usually, only one protocol (ESP or AH) is used so the connection would use two SAs (one inbound and one outbound). However, it is possible for a connection to use four SAs if it uses two proposals and policies, one requiring the ESP protocol and the other requiring the AH protocol.

IKE negotiates SAs in the following sequence:

Phase 1 IKE:

The session initiator creates a cookie and sends it to the responder, with a zero placeholder in the responder cookie area. The responder then creates a cookie and fills in the zeros. All packets will contain these two cookies until the Phase 1 SA expires. IKE Peer commands next establish the identity of local and remote peers. Then IKE Proposal commands specify how packets will be encrypted and/or authenticated for the initial exchange.

Phase 2 IKE

IKE IPSec Proposal commands specify *how* packets will be encrypted/authenticated for the final SA. Then IKE IPSec Policy commands specify *which* packets will be encrypted/authenticated for the final SA.

IKE Commands

The Internet Key Exchange (IKE) process consists of two phases. In phase 1, a moderately secure connection is established between the two security endpoints. This connection is used to exchange key and connection information for the final security association (SA), which is used to exchange user data.

Use the following command to clear all IKE configuration information from the router.

ike flush

The other IKE commands relate to the four categories of information required to set up IKE in the router.

1. IKE *Peer* commands establish the identity of the local and remote peers.
2. IKE *Proposal* commands define the proposals exchanged during the Phase 1 exchange.
3. IKE *IPSec Proposal* commands specify the parameters for the final SA.
4. IKE *IPSec Policy* commands specify the filtering parameters for the final SA.

IKE Peer Commands

The IKE peer commands establish the identity of the local and remote peers.

ike peers add <*PeerName*> Defines the name of a new IKE peer.

ike peers delete <*PeerName*> Deletes an existing IKE peer.

ike peers list Lists the IKE peers.

The following commands define the peer connection.

ike peers set mode <MAIN | AGGRESSIVE> <*PeerName*>

Sets the peer connection to either main or aggressive mode. Main mode is used when the IP addresses of both ends are known. Aggressive mode is used when the address of one end can change, as with a typical modem or DSL connection.

For a *main mode* connection, set only the IP address and the secret:

ike peers set address <*IPaddress*> <*PeerName*>

Sets the IP address of the other endpoint. In a main mode configuration, the other endpoint is constant.

ike peers set secret <*secret*> <*PeerName*>

Sets the shared secret for the peer. The secret must be identical for both ends. It can be up to 256 characters long; do not use spaces or non-printable characters.

For an *aggressive mode* connection, you must set the IP address and secret and several more options.

ike peers set address <*IPaddress*> <*PeerName*>

Sets the IP address of the other endpoint. One end, the gateway, has a fixed IP address. The other end, the client, has a changing address. When configuring the client, set the peer IP address to the gateway's fixed address. When configuring the gateway for aggressive mode, set the IP address to **0.0.0.0**.

ike peers set secret <*secret*> <*PeerName*>

Sets the shared secret for the peer. The secret must be identical for both ends.It can be up to 256 characters long; do not use spaces or non-printable characters.

ike peers set localid <*AggressiveModeID*> <*PeerName*>

Sets the local ID. This must match the peer ID on the other end.

ike peers set localidtype <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

Sets the type of the local ID (IP address, domain name, or e-mail address). This must match the peer ID type on the other end.

ike peers set peerid <*AggressiveModeID*> <*PeerName*>

Sets the peer ID. This must match the local ID on the other end.

ike peers set peeridtype <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

Sets the type of the peer ID (IP address, domain name, or e-mail address).This must match the local ID type on the other end.

IKE Proposal Commands

The IKE proposal commands define the proposals exchanged during the Phase 1 SA.

ike proposals add <*ProposalName*>	Defines the name of a new IKE proposal.
ike proposals delete <*ProposalName*>	Deletes an existing IKE proposal.
ike proposals list	Lists the IKE proposals.

The following commands specify the contents of the proposals exchanged.

ike proposals set session_auth <PRESHARE> <*ProposalName*>

Proposes the session authentication; preshared key is currently the only option.

ike proposals set encryption <DES | 3DES> <*ProposalName*>

Proposes the encryption method used, as follows:

DES Encryption using a 56-bit key.

3DES Encryption using three 56-bit keys, thus, producing 168-bit encryption.

ike proposals set message_auth <NONE | MD5 | SHA1> <*ProposalName*>

Proposes the message authentication performed. It can propose no message authentication or authentication using the hash algorithm Message Digest 5 (MD5) or Secure Hash Algorithm-1 (SHA1).

ike proposals set dh_group <NONE | 1 | 2 > <*ProposalName*>

Proposes the Diffie-Hellman (DH) key generation group used (no group or group 1 or 2).

ike proposals set lifetime <*seconds*> <*ProposalName*>

Proposes the length of time (in seconds) before the Phase 1 SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

IKE IPSec Proposal Commands

The IKE IPSec proposal commands define the proposals exchanged to set up an IPSec SA, that is, an SA for the user data transfer.

ike ipsec proposals add <*ProposalName*>	Defines the name of a new IKE IPSec proposal.
ike ipsec proposals delete <*ProposalName*>	Deletes an existing IKE IPSec proposal.
ike ipsec proposals list	Lists the IKE IPSec proposals.

The followings *proposals set* commands specify the contents of the proposals exchanged.

Note: The next three commands (**set espenc**, **set espauth**, and **set ahauth**) determine the encapsulation method (AH or ESP) used and the authentication and/or encryption requested by the proposal.

You cannot request both AH and ESP encapsulation in the same proposal. (It is possible for a connection to use two proposals, one that requests AH and the other that requests ESP.)

In any one proposal, you can request any *one* of the following:

• AH authentication • ESP encryption • ESP authentication • ESP encryption *and* authentication

ike ipsec proposals set espenc <DES | 3DES | NULL | NONE> <*ProposalName*>

Determines whether ESP encryption is requested and, if it is requested, the encryption method used.

DES	Use ESP encapsulation and 56-bit encryption
3DES	Use ESP encapsulation and 168-bit encryption (if 3DES is enabled in the router; see Software Option Keys, page 124.)
NULL	No encryption, but use ESP encapsulation. Headers are inserted as though the data was encrypted. This allows verification of the source, but sends the data in the clear, increasing throughput.
NONE	No encryption and no ESP encapsulation. (If you select this option, the encapsulation method must be requested by a **set espauth** or **set ahauth** command.)

ike ipsec proposals set espauth <MD5 | SHA1 | NONE> <*ProposalName*>

Determines whether ESP message authentication is requested and, if it is requested, the hash algorithm used.

MD5	Use ESP encapsulation and authenticate using hash algorithm Message Digest 5.
SHA1	Use ESP encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE No ESP encapsulation and no ESP message authentication. (If you select this option, the encapsulation method must be requested by a **set espenc** or **set ahauth** command.)

ike ipsec proposals set ahauth <MD5 | SHA1 | NONE> <*ProposalName*>

Determines whether AH message authentication is requested and, if it is requested, the hash algorithm used.
Note: The proposal cannot request both AH encapsulation and ESP encapsulation.

MD5 Use AH encapsulation and authenticate using hash algorithm Message Digest 5.

SHA1 Use AH encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE No AH encapsulation and no AH message authentication. (If you select this option, the encapsulation method must be requested by a **set espenc** or **set espauth** command.)

ike ipsec proposals set ipcomp <NONE | LZS> <*ProposalName*>

Proposes either no compression or LZS compression.

ike ipsec proposals set lifetime <*seconds*> <*ProposalName*>

Proposes the length of time (in seconds) before the IPSec SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

ike ipsec proposals set lifedata <*kbytes*> <*ProposalName*>

Proposes the maximum number of kilobytes for the IPSec SA; 0 means unlimited. After the maximum data is transferred, IKE renegotiates the connection. By limiting the amount of data that can be transferred, you reduce the likelihood of the key being broken.

IKE IPSec Policy Commands

The IKE IPSec policy commands specify the filtering parameters for the IPSec SA.

ike ipsec policies add <*PolicyName*>	Defines the name of a new IPsec policy.
ike ipsec policies delete <*PolicyName*>	Deletes an existing IPSec policy.
ike ipsec policies list	Lists the IPSec policies.
ike ipsec policies enable <*PolicyName*>	Indicates that the specification of this IPSec policy is complete and enables use of the policy.
ike ipsec policies disable <*PolicyName*>	Disables an IPSec policy.

The following commands define the filtering parameters for the policy.

ike ipsec policies set peer <*PeerName*> <*PolicyName*>

Specifies an IKE peer that may be used for the connection. (The peer must have been defined by IKE peer commands.)

ike ipsec policies set mode <TUNNEL | TRANSPORT> <*PolicyName*>

Specifies the encapsulation mode (tunnel or transport) that may be used for the connection. The default is tunnel mode.

ike ipsec policies set proposal <*ProposalName*> <*PolicyName*>

Specifies an IKE IPSec proposal that may be used for the connection. (It must have been defined by IKE IPSec proposal commands.) The policy may allow more than one value for the proposal parameter. For example, two **set proposal** commands could specify two proposals, either of which could be used by the connection.

ike ipsec policies set pfs <none | 1 | 2 > <*PolicyName*>

Sets the Perfect Forward Secrecy negotiation and specifies the Diffie-Hellman group used for each rekey (none or group 1 or 2). Perfect Forward Secrecy increases the security of the key exchange; compromise of a single key permits access to only the data protected by that particular key. However, the additional encryption slows the IKE process so it is not always desirable.

ike ipsec policies set source <*IPaddress*> <*IPmask*> <*PolicyName*>

Requires that the data come from the specified source IP address and mask.

ike ipsec policies set dest <*IPaddress*> <*IPmask*> <*PolicyName*>

Requires that the data be intended for the specified destination IP address and mask.

ike ipsec policies set translate on | off <*PolicyName*>

Determines whether the router applies NAT (network address translation) before the packets are encrypted by IPSec. If **translate** is set to **on**, the packets are sent using the host router's public IP address. The remote must have IP address translation enabled (see NAT on page 95). The address that NAT translates to should be the source or destination address for the policy (use the **set source** or **set dest** commands).

ike ipsec policies set protocol <*ProtocolNumber* | TCP | UDP | *> <*PolicyName*>

Requires a specific protocol that must be used or allows any protocol (*).

ike ipsec policies set sourceport <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

Requires a specific source port for the data or allows any source port (*) (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

ike ipsec policies set destport <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

Requires a specific destination port for the data or allows any destination port (*). (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

ike ipsec policies set interface <*interface*> <*PolicyName*>

Requires a specific interface that must be used or allows all interfaces (**all**). The policy is only used when the specified interface is connected. The specified interface must be the interface to the IKE peer.

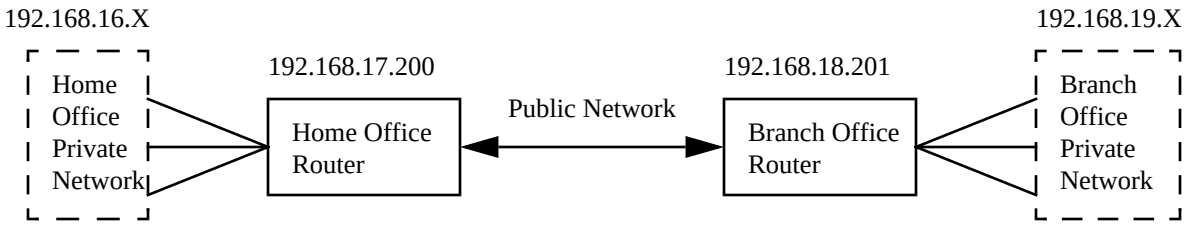
IKE Configuration Examples

This section shows two simple IKE configurations. The installation CD also contains sample configuration files. These files can be edited for your installation and copied to the router using TFTP or the Windows Quick Start application. For more information on TFTP use, see Batch File Command Execution, page 183.

The first example in this section shows an IKE configuration that uses main mode for a secure connection between two routers with fixed IP addresses. The second example shows how the first configuration must change when one of the routers no longer has a fixed IP address thus, requiring aggressive mode.

Main Mode Example

The following example lists two setup files that configure two routers for an IKE main mode connection. The two routers are referred to as the home office router and the branch office router.



The configuration sets up a secure connection between the two routers across a public network, thus, the routers are identified by their public IP addresses on the **ike peers** commands. The packets that are transmitted through this secure connection are from devices in the home office and branch office networks. These networks use private addresses, and thus the packets contain private IP addresses. The **ike ipsec policies** commands specify these private source and destination addresses.

This is the file for the home office router:

```
# Home office example using IKE
# Home router private network addresses are 192.168.16.X
# Home router public address is 192.168.17.200
# Branch router private network addresses are 192.168.19.X
# Branch router public address is 192.168.18.201

# Describe the branch office peer
# IKE main mode is used because the branch office has a fixed IP address
# (192.168.18.201). The shared secret is ˝ThisIsASecret12345;)˝
ike peers add branch_peer
ike peers set mode main branch_peer
ike peers set address 192.168.18.201 branch_peer
ike peers set secret ThisIsASecret12345;) branch_peer

# Describe the branch office IKE phase 1 connection
# DES encryption
```

```
# MD5 authentication
# Diffie-Hellman group 2 key exchange
# 24-hour timeout
# Unlimited data
ike proposals add branch_proposal
ike proposals set encryption des branch_proposal
ike proposals set message_auth md5 branch_proposal
ike proposals set dh_group 2 branch_proposal
ike proposals set lifetime 86400 branch_proposal

# Describe the desired IPSec connection
# Triple-DES encryption
# SHA1 authentication
# 30-minute timeout
# Unlimited data
ike ipsec proposals add branch_ipsec_prop
ike ipsec proposals set espenc 3des branch_ipsec_prop
ike ipsec proposals set espauth sha1 branch_ipsec_prop
ike ipsec proposals set lifetime 1800 branch_ipsec_prop
ike ipsec proposals set lifedata 0 branch_ipsec_prop

# Describe the packets to be encrypted
# All packets from network 192.168.19.0/24 to network 192.168.16.0/24
ike ipsec policies add branch_policy
ike ipsec policies set source 192.168.16.0 255.255.255.0 branch_policy
ike ipsec policies set dest 192.168.19.0 255.255.255.0 branch_policy
ike ipsec policies set peer branch_peer branch_policy
ike ipsec policies set proposal branch_ipsec_prop branch_policy

# Enable the IKE connection
ike ipsec policies enable branch_policy

# Save the setup and reboot
save
reboot
```

This is the file for the branch office router:

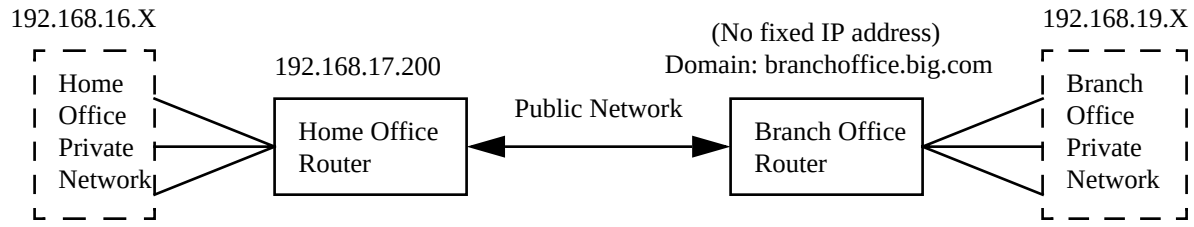
```
# Branch office example using IKE
# Home router private network addresses are 192.168.16.X
# Home router public address is 192.168.17.200
# Branch router private network addresses are 192.168.19.X
# Branch router public address is 192.168.18.201
```

```
# Describe the home office peer
# IKE main mode is used because the home office has a fixed IP address
# (192.168.17.200). The shared secret is ˝ThisIsASecret12345;)˝
ike peers add home_peer
ike peers set mode main home_peer
ike peers set address 192.168.17.200 home_peer
ike peers set secret ThisIsASecret12345;) home_peer

# Describe the home office IKE phase 1 connection
# DES encryption
# MD5 authentication
# Diffie-Hellman group 2 key exchange
# 24-hour timeout
# Unlimited data
ike proposals add home_proposal
ike proposals set encryption des home_proposal
ike proposals set message_auth md5 home_proposal
ike proposals set dh_group 2 home_proposal
ike proposals set lifetime 86400 home_proposal

# Describe the desired IPSec connection
# Triple-DES encryption
# SHA1 authentication
# 30-minute timeout
# Unlimited data
ike ipsec proposals add home_ipsec_prop
ike ipsec proposals set espenc 3des home_ipsec_prop
ike ipsec proposals set espauth sha1 home_ipsec_prop
ike ipsec proposals set lifetime 1800 home_ipsec_prop
ike ipsec proposals set lifedata 0 home_ipsec_prop

# Describe the packets to be encrypted
# All packets from network 192.168.16.0/24 to network 192.168.19.0/24
ike ipsec policies add home_policy
ike ipsec policies set source 192.168.19.0 255.255.255.0 home_policy
ike ipsec policies set dest 192.168.16.0 255.255.255.0 home_policy
ike ipsec policies set peer home_peer home_policy
ike ipsec policies set proposal home_ipsec_prop home_policy

# Enable the IKE connection
ike ipsec policies enable home_policy

# Save the setup and reboot
```

```
save
reboot
```

Aggressive Mode Example

This example supposes, like the preceding main mode example, that a secure connection is needed between a home office router and a branch office router. However, now the DSL connection for the branch office router does not provide a fixed IP address for the branch office router. Thus, an aggressive mode IKE configuration is required.



To change the main mode configuration to an aggressive mode configuration, you only need to change the **ike peers** commands. All the other IKE commands remain the same. Change the mode to aggressive and change the address of the router that has no fixed address to 0.0.0.0, and specify either its e-mail address or domain name.

Note: Remember to save and reboot each router after entering the configuration changes.

Change the **ike peers** commands in the home office router configuration to the following:

```
#Describe the branch office peer
#IKE aggressive mode is required because the branch office does not have
#a fixed IP address. The shared secret is “ThisIsASecret12345;)”
ike peers add branch_peer
ike peers set mode aggressive branch_peer
ike peers set address 0.0.0.0 branch_peer
ike peers set secret ThisIsASecret12345;) branch_peer
ike peers set peeridtype domainname branch_peer
ike peers set peerid branchoffice.big.com branch_peer
ike peers set localidtype ipaddr branch_peer
ike peers set localid 192.168.17.200 branch_peer
```

Change the **ike peers** commands in the branch office router configuration to the following:

```
#Describe the home office peer
#IKE aggressive mode is required because the branch office does not have
#a fixed IP address. The shared secret is “ThisIsASecret12345;)”
ike peers add home_peer
ike peers set mode aggressive home_peer
ike peers set address 192.168.17.200 home_peer
ike peers set secret ThisIsASecret12345;) home_peer
ike peers set peeridtype ipaddr home_peer
ike peers set peerid 192.168.17.200 home_peer
```

```
ike peers set localidtype domainname home_peer
ike peers set localid branchoffice.big.com home_peer
```

IPSec Commands

The following commands allow you to define an IPSec connection without IKE.

Note: If you define a tunnel using IPSec commands, the keys will remain static. This could pose a security risk and is not recommended. Use of IKE for key management is recommended.

ipsec flush	Clears all IPSec definitions.
ipsec add <*SAname*>	Defines an SA name.
ipsec del <*SAname*>	Deletes an existing SA.
ipsec list [<*SAname*>]	Lists one or all SA entries.
ipsec enable <*SAname*>	Enables a defined SA entry.
ipsec disable <*SAname*>	Disables a defined SA entry.

The following commands define parameters for the specified Security Association (SA).

ipsec set mode <TUNNEL | TRANSPORT> <*SAname*>

Requests the encapsulation mode (tunnel or transport) for the SA. The default is tunnel mode.

ipsec set direction <INBOUND | OUTBOUND> <*SAname*>

Defines the direction of the SA.

ipsec set gateway <*IPaddress*> <*SAname*>

Defines the IP address of the gateway.

ipsec set encryption <NULL | DES-CBC | 3DES> <*SAname*>

Selects no encryption, DES (56-bit) encryption or 3DES (168-bit) encryption.

ipsec set authentication <SHA1 | MD5> <*SAname*>

Selects authentication using either SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5)

ipsec set enckey <*key*> <*SAname*>

Specifies the encryption key (in hexadecimal, 64 bits for DES or 192 bits for 3DES).

ipsec set authkey <*key*> <*SAname*>

Specifies the authentication key (hexadecimal).

ipsec set ident <*ident*> <*SAname*>

Specifies the identifier (SPID) for the IPSec tunnel. It must match the SPID at the other end of the tunnel, that is, the tx SPID on this end must match the rx SPID on the other end.

ipsec set service <ESP | AH | BOTH> <*SAname*>

Selects the authentication and/or encryption services used: AH authentication, ESP encryption, or both ESP encryption and ESP authentication (encryption applied first and then authentication).

ipsec set compression <NONE | LZS> <*SAname*>

Selects either LZS compression or no compression.

Chapter 6. Managing the Router

This chapter describes facilities for managing, monitoring, and securing the router. The options discussed include:

- SNMP, Syslog, TFTP, Telnet, and BootP support
- Booting software
- Upgrading the router with new releases of software
- Backing up and restoring configuration files
- flash memory recovery
- Password recovery
- Script execution

SNMP Support

SNMP (Simple Network Management Protocol), a member of the TCP/IP protocol suite, was designed to provide network management interoperability among different vendors' management applications and equipment. SNMP provides for the exchange of messages between a management client and a management agent. The messages contain requests to get or set variables that exist in network nodes, thus allowing a management client to obtain statistics, set configuration parameters and monitor events. These variables (or objects) are defined in Management Information Bases (MIBs), some of which are general or standard SNMP-defined bases. Other bases, such as Enterprise Specific MIBs are defined by different vendors for specific hardware.

The router provides SNMP agent support and support for standard as well as Enterprise Specific MIBs. SNMP is also used internally for configuration of the router. The active SNMP agent within the router accepts SNMP requests for status, statistics, and configuration updates. Communication with the SNMP agent occurs over the LAN or WAN connection.

The supported MIBs and a description of their contents are listed in the following table:

MIB II	Internet-standard MIB contains only essential elements such as system, interface, addressing, protocol (e.g., IP) and SNMP objects
Bridge MIB	State/statistics (including spanning tree states) within bridging system
Ethernet MIB	State/statistics of Ethernet port (e.g., collisions)
IP Forwarding MIB	State of routing tables (updates MIB II)
PPP MIB For LCP	State/statistics for each PPP link
Enterprise MIB for configuration	Router-specific objects for configuration purposes

Any management application using SNMP over UDP/IP has access to the local SNMP agent. SNMP network management tools vary but often have features to display network maps of SNMP nodes, poll nodes at intervals,

trigger alarms on thresholds, graph or list node statistic counters, view and edit individual MIB variables, and print reports.

An example of useful information that can be obtained from a remote SNMP client would be the current status of the router's WAN link and Ethernet interfaces, including protocol (PPP, CSMA-CD), line speed, maximum frame (transmission unit) size, physical address, operating status, or packet traffic rates.

Telnet Remote Access

The router supports Telnet access. Telnet allows you to log in to the router as if you are directly connected through the console port. You can issue commands, using the command line interface, to configure the router and perform status monitoring from any remote location.

To access the router using Telnet, use one of the available TCP/IP packages containing the Telnet application. Issue the appropriate command syntax and assign the IP address of the router. You are then directly connected to the router and can issue commands. (For an example, see Telnet Session for Remote Access, on page 16.) To end the Telnet session, exit the application by entering **logoff** or another appropriate command.

By default, a system security timer logs out a Telnet session after 10 minutes of inactivity. To change the timer period, use the **system securitytimer** command, page 252.

To disable and re-enable Telnet access, use the command **system telnetport**, page 259. For more information on controlling Telnet access to the router, see page 107.

Client TFTP Facility

A client Trivial File Transfer Protocol (TFTP) facility is built into the router that is capable of reading from and writing to the network. A TFTP server must be properly configured to communicate with the router for file transfers to be successful. The client TFTP facility can be used to boot software from a TFTP server, perform software upgrades and copy configuration files to a TFTP server. A TFTP server is integrated into the Windows' Configuration Manager and can also be used as a stand-alone application.

TFTP Server

The TFTPD (Trivial File Transfer Protocol Daemon) program is installed on your PC as part of the DSL Tools software. TFTPD waits for incoming TFTP requests from TFTP clients. It can put a file on your computer's hard disk or get one from it.

Because there is no security built into TFTPD, it is important to specify a root directory where all accessible files are located. When a file is requested, it must be at or below the level of this root directory on your directory tree or the request is denied. If a TFTP client attempts to put a file on your PC, the file must already exist for writing.

The **Options** menu of the TFTPD program allows the user to configure additional parameters, such as the number of retries and the time between retries. The root directory can also be specified from the **Options** menu.

The DOS command line usage for TFTPD is:

TFTPD rootdirectory

The TFTPD operational parameters are kept in file ROUTER.INI in the form:

```
rootdir=rootdirectory
retries=maxtries
timeout=timeout
```

TFTPD is automatically called by BootP and Configuration Manager.

BootP Service

This section first discusses what BootP is and then describes the BootP service available from the router.

BootP Concepts

BootP refers to the Bootstrap Protocol. In general, BootP requests have these purposes:

- To obtain an IP address to use.
- To obtain a TFTP server address and file information to continue the booting up process.

For example, a diskless workstation could use a BootP request to get an IP address for itself, the TFTP server address where it is to get the kernel it is to load and run, and the file name of that kernel.

A BootP server waits for incoming BootP broadcasts from BootP clients. The server looks up the MAC addresses of the incoming BootP request in its database. If the MAC address is found, the server normally responds to the requestor with an IP address. It may also respond with boot information, that is, the IP address of a TFTP server, and the name of a file.

BootP Service by the DHCP Server

BootP is a subset of DHCP. The router has a DHCP (Dynamic Host Configuration Protocol) server (as described in detail on page 85). By default, the DHCP server ignores BootP requests. However, if desired, you can enable the DHCP server in the router to process BootP requests. BootP processing can be enabled globally, on a per subnetwork basis, or on a per client (IP address) basis. For more information, see Managing BootP, on page 91.

If the DHCP server in the router is disabled, it, of course, cannot process BootP requests even if BootP processing is enabled. The DHCP server in the router disables itself if one of the following occurs:

- If another DHCP server is active on the network.
- If you enter the commands **dhcp disable all** and **save**.
- If the DHCP relay list contains one or more IP addresses.

Relaying BootP Requests

The DHCP relay list is an optional list of IP addresses of servers on the network. You create the list manually; addresses are not automatically added or removed. You add addresses to the list using the command **dhcp addrelay** (page 352) and remove addresses from the list using the command **dhcp delrelay** (page 355).

While the relay list contains at least one address, the DHCP server in the router is disabled, and the router forwards all DHCP requests and BootP requests to all servers in the relay list. It forwards every reply received from any of the servers in the relay list to the appropriate LAN.

If you remove all addresses from the DHCP relay list, the DHCP server is re-enabled and resumes processing DHCP requests and also BootP requests if BootP processing is enabled.

Syslog Client

The router can act as a Syslog client, automatically sending system event messages to one or more Unix Syslog servers. (For example, if you request an IP filter watch, the messages are sent to the Syslog servers; see ETH IP FILTER, on page 270.) Messages generated by the router and sent to a Syslog server are sent to facility *local0* with priority *notice*.

To send messages to Syslog servers, the router must know:

- The Syslog port number, and
- The IP address(es) of the Syslog servers.

To disable, re-enable, or redefine the Syslog port, use the command **system syslogPort** (page 259).

The router can learn the IP addresses of Syslog servers in two ways:

- Via DHCP. The router can, under certain circumstances, send out a DHCP message and learn the IP address(es) of Syslog servers. For more information, see DHCP Client Requests, on page 85.
- By explicit configuration. To configure the IP address of a Syslog server, use the command **system addSyslogServer** (page 235).

You can limit the Syslog server addresses that the router learns through DHCP. To do so, set a filter for valid Syslog server addresses using the command **system addSyslogFilter** (page 234).

Boot Code Options

The router provides a number of options for booting router software.

- You can boot from the router's flash memory, the most common option.
- Or, you can boot across the LAN network from a TFTP server, perhaps to test a new level of router software before downloading it to flash memory.
- You can also boot through a gateway to a WAN. The router allows you to set permanent network boot parameters used during network booting, and it enables you to temporarily override those parameters.
- Finally, the router lets you define the order in which the router boot procedures are performed. You can make changes to the boot procedures and specify network boot parameters by entering manual boot mode.

The next section describes the purpose and functions of the boot code. The section following it, Manual Boot Mode, on page 170, describes a menu of manual boot options.

Note: For routers with a reset button, see *Recovering Kernels for Routers with a Reset Button, on page 181*.

What is the Boot Code?

The boot code is responsible for initializing the hardware from an initial power up state and then transferring control to the operating system (kernel).

It does the following major tasks:

- Reads flash memory and does a CRC check and magic number before proceeding
- Performs a power on self test (POST)
- Initializes interface controllers, RAM, and LEDs
- Detects interface types (WAN, console, Ethernet)
- Detects optional VPN hardware (Rapid Secure DES)
- Reports to the console: CRC check, flash memory and RAM sizes, DSL type, and POST results
- Checks whether the reset switch is depressed and skips ASIC load if requested
- Loads the file ASIC.AIC if present
- Reports to the console: the MAC address, WAN modem ID, date/time and the reason for the reboot
- Initializes all RAM to a known content (all zeroes).
- Loads the file KERNEL.F2K from flash memory
    - If the load succeeds, transfers control to the OS (kernel)
    - If load fails, issues a Bootp request
        - If no response, displays the boot menu (see Manual Boot Mode, on page 170).

The boot code communicates to the application it launches (usually, the kernel) information about the hardware capabilities of the router model, including the amount of RAM, the flash memory available for the file system,

ports (Ethernet, xDSL, etc.), the CPU type, and clock speed. It continues to provide basic I/O services to the launched application, including the erasure and programming of flash memory.

Manual Boot Mode

When the router is shipped, it is set for automatic boot from flash memory. To change these boot defaults, you must enter *manual boot mode*.

In manual boot mode, you can:

- change the boot options to allow for network booting.
- change the order of boot procedures.
- perform a manual boot.

The router enters *manual boot mode* if either the kernel is not found in flash memory or a Bootp load from the network fails.

Note: If the router has configuration (dip) switches on its back panel, you can select manual boot mode by setting switch 6 down and rebooting or powering up the router. To return to automatic boot mode, set switch 6 up and reboot by selecting menu option 1, 2, 3, or 4.

In manual boot mode, the router displays this menu of options:

```
1. Retry start-up
2. Boot from Flash memory
3. Boot from network
4. Boot from specific file
5. Configure boot system
6. Set date and time
7. Set console baud rate
8. Start extended diagnostics

Enter selection:
```

Note: Options 6, 7, and 8 do not appear on the model 5950.

Option 1: Retry Start-Up

Select option **1. Retry start-up** to reboot the router in the boot procedure order. The boot procedure order is either the one you have specified or the default order. The default order is to boot from flash memory and then from the network (if defined). If you wish to boot from the network and/or alter the boot procedure order, refer to Option 3: Boot from Network, on page 170.

Option 2: Boot from Flash Memory

Select option **2. Boot from Flash memory** to perform a manual boot from flash memory. If the boot is unsuccessful, the router returns to manual boot mode. (When you first receive the router, it defaults to booting from flash during power-up or automatic reboot.)

Option 3: Boot from Network

Once you have installed router software on a network TFTP server, you can have the router boot across the LAN. Option **3** requests a manual boot from the network. It uses the network boot parameters you have defined using option **5**.

If you have not defined network boot parameters, the router attempts to locate a BOOTP or RARP server on the network.

- BOOTP can be used to supply an IP address, a TFTP server IP address, and a file name.
- RARP can obtain an IP address, if it knows the MAC address. The router assumes that the RARP server is also capable of performing the duties of a TFTP server and so the router requests the file `KERNEL.F2K` (or the filename assigned when permanent network boot parameters are set.)

If a BOOTP or RARP server exists and is properly configured with the router's MAC address, the router boots from the network.

If the boot from the network is unsuccessful, the router returns to manual boot mode.

Option 4: Boot from Specific File

Select option **4** to temporarily override permanent network boot parameters when you perform a network boot.

1. After you select option **4**, the current default (permanent) parameters are shown.
2. Set new temporary values for the network boot parameters.
3. Press the return key and the router boots from the network using the temporary boot parameters.

If the boot is unsuccessful, the router returns to manual boot mode.

Option 5: Configure Boot System

Select option **5** to specify permanent network boot parameters. This menu is displayed:

```
1. Configure boot order, currently "flash, then network"
2. Set permanent IP address, currently not defined
3. Set permanent TFTP boot server, currently not defined
4. Set permanent IP gateway (boot only), currently not defined
(Option 5 for model 5950 only)
5. Set file name to boot from (FLASH and TFTP), currently
"kernel.f2k"

Hit <return> to leave this menu

Enter selection:
```

1. Select options **2**, **3**, and **4** to set the three boot parameters (boot IP address, TFTP boot server address, and router software file name on the server). To reset any parameter, press **`enter`** following the prompt.

   The boot IP address is the router LAN IP address used *during* the boot procedure. This address may differ from the LAN IP address that the router is ultimately assigned. This address is different so that a system can be booted from one subnetwork and then moved to its operational network, if necessary.

The TFTP boot server address is the LAN IP address of the boot server (4 decimals separated by periods).

Note: Once you have set a TFTP server address, it is assigned to the router software TFTP facility. This server address is then used whenever a server address is not explicitly specified, including when the **copy** command is in the form: **copy tftp:***filename* **kernel.f2k**

The router software file name must be in the format: **yyyyyyyy.yyy** (similar to the DOS filename format).

2. Set the boot procedure order. You can specify whether the router boots from flash memory first, from a network TFTP server first, or never automatically reboots.

   a. Select step **1** under `Configure Boot System`, option **5**.

   b. To boot from flash memory first, select option **1**;
   To boot from the network first, select option **2**.
   If you select option **3**, the router will always go into manual boot mode; that is, you must always select the boot procedure to be performed.

3. Select option **4** to `Boot through the IP gateway`. In this procedure, the router on the local LAN can boot from a boot server that is not connected directly. Instead, the path to the boot server can include other networks (including the WAN, if adequate routers exist). The gateway must be located on the local LAN and be reachable by the local router.

4. **(Model 5950 only).** On the model 5950, you can boot from either of two files in flash memory. This can be used to run a test kernel and back up the previous version. Thus, if you select option 5, you see this prompt:

   ```
   Enter the file name you want to boot from [kernel.f2k]:
   ```

   Enter the file name after the prompt (for example, **test.bin**).

Option 6: Set Time and Date

Select option **6** to set the current time and date. Set the new date in the format mm[/dd[/yy (or yyyy)]]. Set the new time in military format hh[:mm[:ss]]). You are shown the current date and time.

Note: Your router is Y2K compliant. If you choose to enter only two digits to specify the year, values greater than 93 translate to 19xx. Values less or equal to 93 translate to 20xx. The router has a one-hundred-year date range (from 1994 to 2093).

If the date is set to zero (0/0/00), the real-time clock is disabled for long-term storage.

When the router is configured by a PC, the GUI overwrites the time and date fields. The router time and date values are copied from the PC time and date values.

Option 7: Set Console Baud Rate

Select option **7** to alter the baud rate that the router uses to communicate over the console port with a terminal emulation program. You can override the default rate of 9600. Remember to set the identical baud rate in your terminal emulation program.

Option 8: Start Extended Diagnostics

Select option 8 to run extended diagnostics. Boot diagnostics are only available on routers with the MC68EN360 processor. These diagnostics run automatically when you power up or reboot the router. You may want to run extended diagnostics if you suspect a hardware problem.

When you select option **8**, the following menu is displayed:

```
[1] DRAM test
[2] Parity test
[3] POST firmware CRC test
[4] Real-Time Clock chip test
[5] Timers and Interrupts test
[6] Multi-port UART (internal loopback) test
[7] Multi-port HDLC (internal loopback) test
[8] SCC2 External Loopback test
[9] SCC3 External Loopback test
[a] SCC4 External Loopback test
[b] Ethernet Transceiver (internal loopback) test
[-] Deselect all tests
[+] Select all tests
[.] Run selected tests
[#] Enter debugger
[/] Exit extended diagnostics (reboot)
```

Enter the number of each test that you would like to run, or select all tests (+). Then enter . (period) to begin diagnostic testing.

The debugging mode (option #) is available for use primarily when you encounter a serious problem, in consultation with customer support services.

Identifying Fatal Boot Failures

Fatal boot failures can be identified by the light patterns shown by the LEDs on the front panel of the router.

Non-fatal errors are not indicated by the LEDs, but they do prompt the system to send an explanatory message to the console port.

Normal LED states are described in the *Hardware Specifications* section of the *User Reference Guide*. (A copy of the *Guide* comes with your router and is available on the web site www.efficient.com.) The normal progression of LED states during startup are described in Using LEDs, on page 184.

Normally, during ready state, the TEST LED flashes every two seconds. If this normal “heartbeat” stops, it indicates that the router is locked up and you need to cycle power to reset it.

Routers with Four LEDs

If your router has four LEDs, the pattern of the three LEDs (except the POWER LED) may indicate a fatal error.

Note: On some router models, the LINK LED is labeled LAN or RX0/TX0 and/or the WAN LED is labeled VOICE or RX1/TX1.

The error patterns are listed in the following table. (Any other pattern of flashing LEDs indicates an internal error. Should this occur, return the router to the factory for repair or replacement.)

TEST	WAN	VOICE or LAN	Fatal Error
Off	Off	Off	Boot ASIC Load error *or* CPM failure
Off	Off	Blinking green	Timer failure *or* Bad FCS
Off	Blinking green	Off	DRAM failure *or* Interrupt failure
Off	Blinking green	Blinking green	SCC failure *or* Manufacturing information error
Blinking amber	Off	Off	CPU step failure *or* Ethernet loop failure
Fast blink green	Off	Off	Wait stuck in the boot menu; kernel file could be missing.
Blinking green	Off	Off	The router is issuing BootP requests (10-second blink).

Routers with Six LEDs

If your router has six LEDs, the pattern of the four LEDs labeled TEST, LINK, WAN, and LANT may indicate a fatal error. The error patterns are listed in the following table. (Any other pattern of flashing LEDs indicates an internal error. Should this occur, return the router to the factory for repair or replacement.)

TEST	LINK	WAN	LANT	Fatal Error
Off	Off	Off	Blinking green	CPM failure
Off	Off	Blinking green	Off	Timer failure
Off	Off	Blinking green	Blinking green	Bad FCS
Off	Blinking green	Off	Off	DRAM failure
Off	Blinking green	Off	Blinking green	Interrupt failure
Off	Blinking green	Blinking green	Off	SCC failure
Blinking amber	Off	Off	Off	CPU step failure
Blinking amber	Off	Off	Blinking green	Ethernet loop failure
Fast blink green	Off	Off	On, off, or blinking	Wait stuck in the boot menu; kernel file could be missing.
Blinking green	Off	Off	On, off, or blinking	The router is issuing BootP requests (10-second blink).

Software Kernel Upgrades

You can upgrade the software kernel by downloading a new version from the LAN or from the WAN.

What is the Software Kernel?

The software kernel is the router operating system; it handles task management, memory management, events coordination, and configuration control. Included with the kernel is a complete DOS-like file system using the on-board flash memory, remote debugging support, console handling, and the software update mechanism.

Specific components include:

- Task Scheduler
- Loadable Module Services
- Event Notification Services
- Memory Management
- Buffer Management
- DOS-like Flash File System
- Inter-Process Communications (IPC)
- Power On Self Test (POST) & Boot Code

Booting and Upgrading from the LAN

You can download a new version of the router software kernel using a TFTP server that already exists on the LAN. The following steps demonstrate how to boot the router software from the network and copy the image from the network into the router's flash memory. When it first connects to the router, the GUI backs up all the files to a directory called Sxxxxx, where x is the router's serial number.

Note: We strongly recommend that you use the Configuration Manager's **Upgrade/Backup** tool to upgrade or back up the kernel. The Configuration Manager's tool is more convenient to use than the Command Line Interface.

Upgrade Instructions

Read the following steps very carefully before you perform an upgrade:

1. **Warning:** Before performing this procedure, make sure that you can successfully boot from the network using the manual boot procedure option 3 or 4. Refer to the section Option 3: Boot from Network, on page 170.

2. Copy the router software file KERNEL.F2K (or KERNEL.FPL for an IDSL router) to a directory where it can be accessed by a TFTP server. The TFTP server must be on the same LAN as the target router; i.e., there must not be a router or gateway between the target system and the TFTP server. If the TFTP sever is not on the same network as the target router, enter the gateway from the boot menu as described in the previous section.

3. Log into the Command Line Interface.

4. Enter the **reboot** command to synchronize the file system and reboot the router. Because the kernel is no longer stored in flash memory, the router tries to boot from the network. If you have never set permanent boot parameters, the router attempts to locate a BOOTP or RARP server. If the router successfully reboots from the server, go to step 7.

5. Select option **4** to boot router software from the TFTP server using temporary network boot parameters. You are prompted for:

   - the router's boot LAN IP address,
   - the TFTP server's IP address,
   - the load address, and
   - the filename of the router's kernel saved on the server.

6. Note that the LAN IP address is the proper address to use during the network boot and this may differ from the IP address ultimately assigned to the router. Enter the temporary network boot parameters (hit the **return** key for the load address). If all entered information is valid, the router boots from the network. An example follows:

```
Enter selection: 4
  Enter my IP address:
128.1.210.65
  Enter server IP address:
128.1.210.70
  Enter load address [80100]:
  Enter file name: kernel.f2k
```

   Alternatively, select option **5** to set permanent network boot parameters and then boot from the network using option **3**. You would use this option if you wish to boot from the network for a period of time before copying the software to flash memory.

7. After the boot is complete, verify that the kernel is running successfully.

8. When you are satisfied that the new kernel is performing as expected, copy the kernel into flash memory in the router by typing the two following commands:

```
copy tftp@xxx.xxx.xxx.xxx:sfilename kernel.f2k
sync
```

   where xxx.xxx.xxx.xxx is the TFTP server IP address, SFILENAME is the server filename of the kernel, and KERNEL.F2K is the name of the file loaded from flash memory by the boot procedure. If you do not specify the server address, a permanent or more recent override TFTP server address will be used, if you have previously defined one. Enter the **sync** command to commit the changes to flash memory.

   Warning: After the kernel is copied, *do not* power down the router until you have issued either a **sync** or **reboot** command to reboot the router. Otherwise, the file will not be written to flash memory.

9. After successfully copying the kernel to the router, reset configuration switch 2 or 6 to the up position (if the router has configuration switches)**.** Then reboot the router from flash memory with the **reboot** command. If you have altered the boot procedure order in any way, reset to boot from flash memory first. Verify the software revision number with the **vers** command.

The system is now ready to be re-configured, if necessary. The configuration files are unchanged by the upgrade process.

Upgrading from the WAN

You can download a new version of the router software kernel by using a TFTP server on the WAN. The following steps show you how to copy the software from the WAN into the router's flash memory.

Warning: Before performing this procedure, make sure that you can successfully access the software from the TFTP server.

1. Copy router software KERNEL.F2K to a directory where it can be accessed by a TFTP server.
2. Log in to the Command Line Interface.
3. Copy the kernel into flash memory in the router using the following commands:

   copy tftp@xxx.xxx.xxx.xxx:sfilename kernel.f2k
   sync

   where xxx.xxx.xxx.xxx is the TFTP server IP address, **sfilename** is the server filename of the kernel, and KERNEL.F2K is the name of the file. If you do not specify the server address, a permanent or more recent override TFTP server address will be used, if you have previously defined one.

   Warning: After the kernel is copied, *do not* power down the router until you have either issued a **sync** command or rebooted the router. Otherwise, the file is not written to flash memory.

4. After successfully copying the kernel to the router, reboot the router from flash memory via the **reboot** command. If a problem occurs during the upgrading process, try the command again (do not reboot until you have successfully copied the kernel). If you have altered the boot procedure order in any way, be sure to reset the router system to boot from flash memory first. Verify the software revision number by issuing the **vers** command.

   The router system is now ready to be re-configured if necessary. The configuration files are unchanged by the upgrade process.

Backup and Restore Configuration Files

To successfully save configuration files to the server, those files must already exist and be writeable by everyone. This restriction is part of the TFTP protocol. Moreover, all the files accessed by the TFTP server must be under a single root directory. Multiple sub-directories can exist below this root directory, but they must be created manually at the server. Neither the sub-directories nor the files can be created remotely.

Note: Remember to start the TFTP server from the **DSL Tools** menu.

The **copy** command lets you upload configuration files to the TFTP server where the destination is in the form:

tftp@xxx.xxx.xxx.xxx:filename.ext

Backup Configuration Files (Recommended Procedure)

1. Create a directory under the TFTP root directory corresponding to the system name you want to back up.
2. Create files called SYSTEM.CNF, DHCP.DAT, and FILTER.DAT in this subdirectory. The files can be empty or not, but should be writeable by everyone.

   Note: SYSTEM.CNF, FILTER.DAT, and DHCP.DAT are three key files that should be backed up. To see other files that you may also want to save, type the command **dir**.
3. To back up a copy of the configuration files, enter

   copy system.cnf tftp@xxx.xxx.xxx.xxx:myname/system.cnf

   copy filter.dat tftp@xxx.xxx.xxx.xxx:myname/filter.dat

   copy dhcp.dat tftp@xxx.xxx.xxx.xxx:myname/dhcp.dat

   where **xxx.xxx.xxx.xxx** is the IP address of the TFTP server and **myname** is the router name.

Restore Configuration Files

To restore the configuration files SYSTEM.CNF, FILTER.DAT, and DHCP.DAT, enter:

copy tftp@xxx.xxx.xxx.xxx:myname/system.cnf system.cnf

copy tftp@xxx.xxx.xxx.xxx:myname/filter.dat filter.dat

copy tftp@xxx.xxx.xxx.xxx:myname/dhcp.dat dhcp.dat

sync

Flash Memory Recovery Procedures

Recovering Kernels for Routers with Configuration Switches

In the unlikely event that the flash file system should become corrupted, attempt to recover using these steps. Perform the following procedures in the order listed:

1. Try to repair the file system by issuing the **msfs** command. While logged in, issue a **sync** command followed by an **msfs** command. If the display shows that the file system is corrupted, verify that no other console (via Telnet) is currently modifying the file system with the **ps** command. Then attempt to repair the file system typing the following commands:

   msfs fix
   sync

2. If the file system is still corrupted (i.e., you cannot write a file), you will have to reformat the file system. First, attempt to save your configuration files as explained in the section *Backup and Restore Configuration Files, on page 179*. Then, while logged in, enter the following commands:

   format disk
   save
   copy tftp@xxx.xxx.xxx.xxx:kernel.f2k kernel.f2k
   sync

   The above command sequence assumes that the software presently running from RAM is correctly configured and is still functional. The **save** command re-creates all the configuration files (except the FILTER.DAT file, which you may re-create manually by typing **save filter**). The **copy** command reinstalls the operational software on the flash file system, and **sync** commits all this information to disk.

3. In the event that the software running from RAM is not sufficiently functional to perform those steps, you will have to boot from the network using a TFTP server, as explained in the section *Software Kernel Upgrades, on page 176*.

   If you cannot issue the **format** command (as explained in the previous step), you will have to erase the flash file system from the boot code.

   a. Flip configuration switch 6 to the down position and reboot the router (by powering down and up again, for example).
   b. At the manual boot menu, select **5**. "Configure boot system", and enter the "magical" number 98. Then, move switch 6 back to its up position.
   c. Reboot from the network following the steps described in the Software Upgrade Procedure. You will notice error messages indicating that the file system is not formatted. Then log in and enter:

      format disk

   d. Recreate the configuration files either by re-entering the information or by restoring them from a TFTP server. Re-install the operational software by entering the command:

      copy tftp@xxx.xxx.xxx.xxx:kernel.f2k kernel.f2k

      This assumes that TCP/IP routing is enabled and that an IP address has been assigned to the Ethernet interface.

Recovering Kernels for Routers with a Reset Button

The reset button (if your router has one) is recessed in an unlabeled hole on the back panel of the router (to the right of the Ethernet hub connector). If your router has a reset button, you can use it to perform these functions:

- If the reset button is depressed during the power on sequence, the router attempts to download a kernel from a BootP server as described in Recovery Steps Using BootP, on page 181. (The BootP server must be configured to recognize the router.) The existing configuration file (SYSTEM.CNF) is written to file SYSTEM.BAK (overwriting the existing SYSTEM.BAK file). The firmware creates a new SYSTEM.CNF file that can be accessed using the default password **admin**.

  Note: Use this function only if other recovery methods have failed and you need to start over with a new kernel image.

- If the reset button is depressed after the TEST LED turns green and until after all the LEDs flash, it bypasses the loading of the FPGA (Field Programmable Gate Array) file that provides the logic that customizes the router hardware. The actual file name is ASIC.AIC. This function is useful if a mismatch occurs between the hardware, the kernel, and the FPGA file because it forces a error that allows you to continue and replace the ASIC.AIC file.

- If the reset button is depressed while running the kernel, the router enters a ten-minute recovery period. During this period, the router password is the router serial number and DHCP is temporarily changed to the default subnet 192.168.254.0. (This reset function is described in Recovering Passwords and IP Addresses, on page 182.)

Recovery Steps Using BootP

A router that fails to boot may be an indication that the kernel has been corrupted. To recover, you need a kernel for your particular router model. If you installed the DSL Tools and successfully connected to the router, an automatic backup process was started that saved a copy of the kernel and other files to the PC in a subdirectory under DSL Tools called Sxxxxxx, where xxxxxx is the serial number of the unit. The file needed for this recovery is called KERNEL.F2K.

Before you proceed with the recovery steps described below, make sure that the router has a good Ethernet connection to the PC. If a console cable is available, you may want to connect it and start a terminal emulator session so you can see the router's console messages. You can also check the LEDs' blinking patterns (on the front panel of the router) to help you identify the state of the router.

1. Make sure that the PC path and directory information to a valid kernel are correct.
2. Start the Configuration Manager or Quick Start application (refer to your *User Reference Guide*).
3. Select Tools and BootP.
4. In the BootP dialog box, enter the following information:

   - The path to the kernel file
   - The serial number of the router
   - The IP address to be used for the boot

   Note: This IP address needs to belong to the same subnet as your PC and not be used by another device. For a simple configuration, the IP address 192.168.254.254 will work if your PC already received an IP address from the router when it was still functioning.

5. In the **BootP Setting** dialog box, click **OK**. Configuration Manager writes the above settings to a file called BOOTDBASE.TXT and calls the Bootp server.

6. Power off the router.

7. Insert a a small pen or pointed object into the small reset switch (unlabeled hole) on the back panel of the router (to the right of the Ethernet hub connector). With the object still inserted in the reset switch, power up the router. Wait until all the LED lights flash (about 10 seconds).

8. Once this is accomplished, the BootP server should receive a BootP request and start the TFTPD server. The TFTPD server will send the kernel to the router.

9. Restart Configuration Manager and try to connect to the router. Using the following instructions, you will attempt to write a new kernel to the flash system.

10. From the Configuration Manager's Main Menu, select **Tools** and **Upgrade/Backup**.

11. Click **Firmware** and the **Upgrade** button.

12. Select a kernel file and click **OK**.

13. Wait until the file is copied, and click **Yes** to reboot the rooter.

Recovering Passwords and IP Addresses

Routers with Configuration Switches

Recover a password: Set switches 5 and 6 in the down position after the router has booted. With this step, the system password is overridden, thus allowing a forgotten password to be re-entered**.**

Recover an IP address: Connect to the console terminal and type the **eth list** command to find out what the router's IP address is.

Routers with a Reset Button

The following step will assist you in recovering the router's administrative password or IP address, should you forget them.

Push the reset button and hold it for 3 seconds while the router is running. With this step, the following features are enabled for a period of 10 minutes:

- The system password can be overridden by using the router's serial number as the password.
- A DHCP client address is enabled or created, so that a connected PC can obtain an IP address from the router.

Batch File Command Execution

This feature is used to load batch files of configuration commands into the router. This allows the user to customize and simplify installation of the router. A script file can contain commands, comments (lines introduced by the # or ; characters), and blank lines.

There are two kinds of script files:

- A one-time script that is executed on startup (only once).
- A group of commands that can be executed at any time from the command line by entering the **execute** <*filename*> command.

One-time scripts are useful to execute the complete configuration process from a default (unconfigured) state.

The following steps describe how to proceed in order to create and execute a one-time script from the Quick Start application.

- Create the script on your PC using Notepad or another text editor. The command syntax can be found in the Command Line Reference manual or enter **?** on the router command line (assuming you have access to the command line with the console or with Telnet).
- Select the **Tools | Execute Script** menu item and choose the script file you just prepared. When you click **OK**, the script file is loaded to the router (under the name AUTOEXEC.BAT) and the router is restarted, thus executing the script.

Alternatively, you can manually transfer the script file from your PC to the router using the following method:

- Start the TFTP server on your PC and set the root directory where the script file is located.
- Use the following command to copy the script file to the router file system:
  copy tftp@ <*PC_IP_address*>:<*PC_file*> <*router_file*>
- To process the commands in the script file, you can either reboot your router (if the script file was copied under the name AUTOEXEC.BAT onto the router) or use the command **execute** <*filename*>.

Note: If present, the file AUTOEXEC.BAT is renamed AUTOEXEC.OLD before it is executed, so that it is only run once. If you clear the router configuration with the **Reset Defaults** button of the **Upgrade/Backup** tool or the **reboot default** command, the AUTOEXEC.OLD is renamed back to AUTOEXEC.BAT and re-run after the boot up, thus restoring your configuration.

Caution: The only limitation on commands in a script file is as follows:

> Do not include the commands **rename autoexec.old autoexec.bat** *and* **reboot** in the *same* one-time script file (copied to the router under the name AUTOEXEC.BAT). This will result in an endless loop of starting the router, executing the script, restarting the router, re-executing the script.
>
> The **rename autoexec.old autoexec.bat** command is useful if you need the script to execute on every startup.
>
> The **reboot** command is useful to apply changes and have them take effect (almost) immediately.

Chapter 7. Troubleshooting

Software problems usually occur when the router's software configuration contains incomplete or incorrect information. This chapter discusses:

- Diagnostic tools that are available to help identify and solve problems that may occur with your router
- Symptoms of software configuration problems
- Actions for you to take
- System messages

Diagnostic Tools

This section describes three diagnostic tools available to you:

- The LEDs on the front panel of your router.
- The History Log that lists the router's activity.
- The **ping** command that can verify IP connectivity.

Using LEDs

The specific pattern of LEDs on your router model are described in the *User Reference Guide* that came with the router. Certain hardware problems can be diagnosed and solved by checking the LEDs.

For the LED patterns that indicate fatal boot errors, see Identifying Fatal Boot Failures, on page 173.

LED Startup Sequence

The normal LED startup sequence involves the LEDs labeled PWR (power), TEST (self-test indicator), and LINK (modem link).

Note: On some models, the LINK LED is labeled LAN or RX0/TX0.

If the Power (PWR) LED is off:

- Check that the power cord is firmly plugged into the back panel of the router and the other end into an active AC wall or power-strip outlet.
- Check that the power switch is turned on.

The following table summarizes the *normal* LED sequence in the left column (five consecutive states) from **Power On** to **Ready State**. The right column suggests problems reflected by an "abnormal" LED state (no progression to the next state).

State	Normal LED Sequence	State Length	Problem If the LED sequence stops at this stage:
State 1 Power ON	PWR - green TEST - amber LINK or WAN - off	5 sec	A hardware problem has been detected. Contact Technical Support.
State 2	All lights flash	1 sec	
State 3	PWR - green TEST - green LINK or WAN - off	5 sec	1. Check that the DIP switches are all up. 2. Check that the correct software was loaded.
State 4	PWR - green TEST - green LINK or WAN - amber (no signal), blinking amber (signal), blinking green (training)	5 to 10 sec	1. Check your DSL cable. 2. Check the physical connection from your router to the DSLAM (Central Office). 3. Possible problem with DSLAM card.
State 5	PWR - green TEST - green LINK or WAN - green	Ready State	

LEDs in Ready State

Once the router is in **Ready State**, the LEDs may blink as follows:

- The TEST LED blinks every two seconds to show that the router remains ready and active.
- The LINK or WAN LED blinks to indicate that the WAN is transmitting.
- If present, the LANT LED blinks to indicate that the Ethernet LAN is transmitting.
- If present, the LANR LED blinks to indicate that the Ethernet LAN is receiving.

If the normal “heartbeat” of the TEST LED stops, it indicates that the router is locked up and you need to cycle power to reset it.

To read about SDSL router LEDS, see SDSL Line Activation, on page 342.

History Log

The **History Log** utility is a troubleshooting tool which displays the router’s activity. It can be accessed from a terminal emulation session (including Configuration Manager) or from Telnet.

To see message explanations, refer to the System Messages section, page 199.

Accessing History Log through Telnet

1. Click **Connect** and then **Remote System.**
2. Enter the router's IP address.
3. Click **Connect.**

Accessing History Log through Configuration Manager

1. Select **Tools** and **Terminal Window** (the console cable is required).
2. Log in with your administration password into the router (e.g. **"admin"**).
3. Use the command **system history** to view the buffer contents.

Other Logging Commands

- If you wish to monitor your router activity at all times, use the command **system log start** to view a continuous log, using Telnet. (This command will not work in a Terminal Window session; it only works from Telnet.)
- The command **system log status** is used to find out if other users, including yourself, are using this utility.
- To discontinue the log at the console, use the command **system log stop.**

When you exit Telnet, you automatically stop any logging programs running in that session.

Note: History Log is preserved across reboots, but not across power outages or power down.

Ping Command

You can verify IP connectivity to the router by running a **ping** command. You will probably find a ping utility bundled with your TCP/IP stack. In Microsoft Windows, the command is called `PING.EXE` and can be found in your Windows directory. The **ping** command provided with the Command Line Inerface is discussed on page 217.

Note: Before using the **ping** command to troubleshoot, make sure that the PWR, TEST, and LINK lights are green, indicating the ready state.

Instructions for Windows 95

1. Start a DOS window.

a. Select **Start** from the Windows 95 taskbar.

b. Select **Programs**.

c. Select **MS-DOS Prompt**.

2. Issue the ping command.

In the DOS window, type the command:

ping <*IP address*>

Example: `ping 192.168.254.254`

Interpretation and Troubleshooting

To isolate a problem with the TCP/IP protocol, perform the following three tests:

1. Try to **ping** the IP address of your PC. If you get a response, proceed directly with step 2. If you don't get a response, check that:
   - The network adapter card is installed.
   - The TCP/IP protocol is installed.
   - The TCP/IP protocol is bound to the network adapter.
2. Try to **ping** the IP address of your router. If you get a response, proceed directly to step 3. If you don't get a response, the problem lies between your PC and router:
   - Check the cables.
   - Check the hub.
   - Make sure that your PC and the local router are in the same IP subnetwork.
3. Try to **ping** the DNS server. Write down the results and call your Network Service Provider.

Investigating Hardware Installation Problems

When investigating a hardware installation problem, first **check the LEDs** on the front panel of the router. Many common hardware problems can be easily diagnosed by the LED indicators. For more information, refer to this chapter's section entitled *Diagnostic Tools*, *Using LEDs*, page 184.

If the terminal window display has a problem:

- Ensure your console is plugged in and turned on.
- Verify that you are on the right communications port (Com1, Com2).
- Check the configuration parameters for speed, parity, etc. Make sure the console is not in an XOFF state. Try entering a **`"ctrl q"`**.
- Verify that the RS232 device attached to the console is configured as a DTE. If not, a crossover or null modem adapter is required.

If the **factory configuration** has a problem:

- Compare the router configuration with your router order.
- Verify that the model number is correct (the number is displayed during the boot procedure). The model number and serial number are also displayed on the main window of Configuration Manager.

Investigating Software Configuration Problems

This section suggests what to do if you cannot:

- connect to the router.
- log in.
- access the remote network.
- access the router via Telnet.
- download software.

It then gives trouble-shooting advice for:

- Telephony services (if you have a VoDSL router)
- L2TP tunnels
- Dial Backup

Connection Problems

If you cannot connect your PC to the target router for configuration:

- For a LAN connection, verify that the router's IP address matches the IP address previously stored into the router's configuration. You must have previously set the router's Ethernet LAN IP address and subnet mask, saved the Ethernet configuration changes, and rebooted the router for the new IP address to take effect.
- Check that your LAN cable is pinned correctly and each pin end is securely plugged in.
  Note: If you are using a straight-through cable, the colors for pins 1, 2, 3, and 6 should match on both connectors. If you are using a crossover cable, the colors for pins 1, 2, 3, and 6 on one connector should match respectively 3, 6, 1, and 2 on the other connector.
- Make sure the PC and target router are on the same IP subnetwork or the target router is reachable through a router on your LAN. They can, however, be on different networks if IP routing is *off*.
- Check Network TCP/IP properties under Windows 95 and the control panel of the TCP/IP driver installed under Windows 3.1.
- Check if the LAN LED on the router's front panel blinks when "pinged".
- Check your Ethernet board IRQ settings: the PC's table may have become "confused". If so, reboot your PC.

Login Password Problems

You have been prompted for the login password and received the following message: Login Password is invalid.

- Type the correct password and press enter. Remember that the password is case-sensitive. If the password is **admin**, check that you are entering it in lowercase and that the Caps key is not active.
- If you have forgotten the password, you must reset the login password. Refer to Recovering Passwords and IP Addresses, on page 182. If the router has configuration switches, perform the following procedure:

1. Move switches **5** and **6** down.
2. Type **login** <*newpasswd*>**.** Password checking is overridden.
3. Move switches **5** and **6** up**.**
4. Complete any configuration update that caused the prompt for login.
5. Change your login password to a new password.
6. Store the configuration and reboot the router.

Note: If you do not reset switches **5** and **6** to the up position and then reboot, the router is placed in maintenance mode. Set switches **5** and **6** up and turn the power off and then on again.

Remote Network Access Problems

Bridging

- Make sure to reboot if you have made any bridging destination or control changes.
- All IP addresses must be in the same IP subnetwork (IP is being bridged).
- Check that a bridging default destination has been configured and is enabled.
- Be sure to reboot if the bridging destination or status has been changed.
- Check that bridging is enabled locally (use the **remote listBridge** command, page 305).
- Verify that bridging is enabled by the remote router (use the **remote list** command, page 304).
- Verify that the authentication passwords are correct.
- Reboot your PC if you have Windows for WorkGroups.
- In Windows 95, do not forget to declare shared disk directories. Check the sharing properties on your **C: drive .**
- In the Terminal Window, check that calls are answered from the remote router.
- Check also for any PAP/CHAP errors for the remote router.

TCP/IP Routing

- Check that Ethernet LAN TCP/IP Routing has been enabled (**eth list** command, page 280).
- The IP addresses of the local and remote networks belong to different IP subnetworks.
- Make sure that there is an existing route to the remote network.
- Make sure that there is a route back from the remote network.
- There must be a source WAN IP address defined if you are using NAT.
- Check that, if required, the source and remote WAN IP addresses are on the same subnetwork
- Reboot if you have made any IP address or control or protocol option changes.

- Check that the IP address of the station/network connected to the LAN beyond the remote router is correct, as well as the associated subnet mask.
- If the remote router WAN IP address and subnet mask are required, check that they have been specified correctly.
- Check that a default route has been specified, if needed.
- Be sure to reboot if IP addresses or control or protocol option changes have been made.
- Check that you are using an Ethernet cable.
- Check that IP routing is enabled at both ends.
- The IP address must be within the valid range for the subnet.
- Verify that the IP and gateway addresses are correct on the PC.
- Windows 95 may remember MAC addresses: if you have changed MAC addresses, reboot the router and the PC.
- In Windows 3.1., check that the TCP driver is installed correctly. Ping (**ping** command) your PC's IP address from the PC. Successful "pinging" results let you know that the TCP driver is working properly.
- If you have changed an IP address to map to a different MAC device, and **ping** or IP fails, reboot your PC.
- Use the **iproutes** command (page 215) to verify which router's name is the default gateway (this cannot be 0.0.0.0).

IPX Routing

- Check that IPX routing has been enabled and that the remote end is enabled for IPX routing.
- Validate that the IPX WAN network number matches the remote router's WAN network number.
- Check that IPX SAPs correctly identify the servers and applications on the remote network and have valid network numbers, node numbers, etc.
- Check that every SAP has a router to its internal network.
- Check that the IPX routes (network numbers, hops, and ticks) seeded into the routing table for network segments and servers beyond the remote router are correct.
- Validate that the IPX WAN network number matches the remote router's WAN network number.
- Check that the IPX routes (network numbers, hops, and ticks) seeded into the routing table for network segments and servers beyond the remote router are correct.
- Check that IPX SAPs correctly identify the servers and applications on the remote network and have valid network numbers, node numbers, etc.
- Be sure to reboot if IPX addresses, routes, SAPs or control has been changed.
- If the router fails to negotiate IPX:
    - Make sure that at least one WAN number is not equal to zero at one end of the link.
    - The server must have an IPX route to the remote LAN.
    - The Novell server needs to have **burst mode** turned on.

    - Large Internet packets have to be turned on.
- For Novell 3.12 and later:
    - Client needs VLM.EXE, net.cfg: large Internet packets=ON, Pburst=5
- If you can't see the server SAPs:
    - Check the frame types using the **eth list** command (page 280) and ensure that they are the same on both routers.
    - Check that the Ethernet cable is correctly plugged in.
    - Make sure that the Novell server is up.

Incorrect VPI/VCI (ATM Routers)

If you are given an incorrect VCI/VPI number or none at all to use for the remote, and you need to determine what the possible value might be, use the **atom findpvc** command (see ATM Debug Commands, on page 204).

Telnet Access Problems

- Ensure that the router has a valid IP address.
- Check that the Ethernet cable is plugged in.

Software Download Problems

- Ensure that a TFTP server is properly set up to locate the router software.
- Verify that the router is loading from the network and not from FLASH memory.

Voice Routing (VoDSL) Troubleshooting

After the router WAN link activates (the WAN or LINK LED is green), you should get a dial tone. The dial tone should be received even if you have not yet configured your IP and bridge network settings.

If you do not get a dial tone, check the following:

- Does the router have power?
- Is the local phone cord plugged in?
- Is the voice PVC set correctly in the router? (See the following debug commands.)
- Is the WAN link down? (The WAN or LINK LED should be solid green.)
- Is the DSLAM provisioned for the second PVC?
- Is the voice gateway connected and provisioned? (If Coppercom or ATM Standards-based gateway is down or not communicating with the IAD, you hear dead air.)
- Is the ATM network down between the DSLAM and the voice gateway?
- Is provisioning for loop start or ground start correct? For ground start, tip and ring may be reversed in the RJ11 cable.

If you get a call treatment tone (tritone or 3-stage tone, Voice LED is amber), check the following:

- Voice PVC is not set in the router or is incorrect.
- WAN link is down (WAN or LINK LED should be solid green when link is up).
- DSLAM is not provisioned for the second PVC.
- Voice gateway is not connected or provisioned (Jetstream and Tollbridge gateways).
- ATM network is down between the DSLAM and voice gateway.

If you hear clicking during heavy data downloads, check that the DSLAM supports quality of service (QoS) and that the ATM switch has the voice PVC provisioned for vRT and the data at a lower priority. You may also be able to reduce or eliminate clicking by adjusting the jitter buffer (see Adjusting the Jitter Buffer, on page 193.)

The Port Monitor GUI program can show you the voice PVC and the last event message. Use the Web GUI to verify the VPI/VCI or DLCI numbers for the data and voice connections. Also check loop start (standard phone set) or ground start. These values must match your Network Service Provider's values.

Voice Router Debug Commands

The following debug commands may be helpful.

ifs Shows whether the data and voice PVC's are configured and percent loading.

dsp <NOEC | ECON> Turns echo canceller on (NOEC) or off (ECON).

dsp provision <*x*> Sets loop or ground start signalling.

dsp tritone < on | off > Turns tritone on or off. When the DSL link is down and a phone goes off-hook, the DSP provides tritone to indicate 'no service'. Turning off tritone allows testing of DSP Ploop without a DSL link.

dsp vpinfo <*x*> Displays port status.

For standalone phone verification. (This is for lab or bench verification only.)

dsp init noabort Starts DSP for this test.

dsp cas *x* Connects and rings port *x*.

dsp ploop *x*-*y* Connects port *x* to port *y*.

dsp init Reinitializes after testing.

For example, to connect port 1 to port 2, use this command sequence:

```
dsp init noabort
dsp cas 1
dsp cas 2
dsp ploop 1-2
```

To test a **7461 router**, (4-port IAD over ADSL) and disable the failover pots interface type, enter these commands:

```
dmt to 3600
dsp failover 1
dsp ploop 2-3
dsp tritone off
dsp ring 2
```

The first command sets the timeout timer to the maximum (see ADSL DMT Router Debug Commands, on page 206). The other commands disable failover, connect ports 2 and 3, disable the gateway down message so you can hear loopbacked voice, and ring port 3. After port 2 and 3 are connected, you can pass audio between the phones. Enter **reboot** to reset everything after the test.

For ATM routers:

atom voicepvc — Displays the voice PVC. (0*39 is the default.)

atom voicepvc <*x***y*> — Changes the voice PVC to the specified *x*y*.

remote setpvc <*x***y*> <*remote*> — Changes the PVC for data (usually 0*38).

For Frame Relay routers:

frame voice — Displays the voice DLCI.

frame voice <*x*> — Changes the voice DLCI to the specified number *x*.

frame stats — Shows LMI statistics. (For a **frame stats** example, see page 334.)

For a Tollbridge gateway:

voice ip cpe — Displays the local IP address, as set by the gateway.

voice ip gateway — Displays the voice gateway IP address, as set by the gateway.

For a Jetstream gateway:

voice l2stats — Shows AAL2 statistics for control messages.

voice l2clear — Resets the AAL2 statistic counters to 0.

For an ATM standards-based gateway:

voice profile — See Changing Your ATM Standard Voice Profile, on page 23.

The following commands allow you to trace all signaling cells sent and received and all encoding changes for voice ports.

voice lestrace 1 — Enables trace messages to the console.

voice lestrace 0 — Disables trace messages to the console.

voice lestrace — Displays trace messages.

To see the CRC and line errors for SDSL, enter:

sdsl stats — For an **sdsl stats** example, see page 345.

Adjusting the Jitter Buffer

The jitter buffer shapes data to overcome the problem of latency, that is, the time delay between packets of voice data that can cause gaps in or loss of traffic in a voice call. The default is 15 milliseconds. A command is available that allows you to adjust the size of the jitter buffer. The command is as follows:

dsp jitter [<*milliseconds*>]

milliseconds — Length of the jitter buffer in milliseconds (0 - 60).

To display the current jitter buffer, enter the **dsp jitter** command without its parameter. For example:

```
# dsp jitter
Jitter Buffer: 15 ms
usage: dsp jitter <milliseconds 0-60>
```

Note: Before changing the jitter buffer size, hang up any active phones and close all data transfers.

Use this command if you hear clicks or distortion. Increase the buffer size until the problem is corrected. However, do not set the buffer unnecessarily large because that would introduce unnecessary latency and voice delay.

The jitter buffer should be set to the best estimate of the effective worst-case jitter in the voice-packet arrival time from the voice gateway. Be aware of the granularity of the setting: for G.711 voice compression, only integer multiples of 5.5 ms can be realized; for G.726, only integer multiples of 11 ms can be realized. Thus, the following table illustrates the difference between the value you specify and the actual holding time for G.711 and G.726. The numbers in parentheses are the number of voice frames held in the jitter buffer.

Value Specified	Actual (G.711)	Actual (G.726)
0-5	5.5 (1)	11 (1)
6-11	11 (2)	11(1)
12-16	16.5 (3)	22 (2)
17-22	22 (4)	22 (2)
23-28	27.5 (5)	33 (3)
29-33	33 (6)	33 (3)
34-39	38.5 (7)	44 (4)
40-44	44 (8)	44 (4)
45-50	49.5 (9)	55 (5)
51-55	55 (10)	55 (5)
56-60	60.5 (11)	66 (6)

L2TP Tunnel Troubleshooting

If you have problems setting up an L2TP tunnel, use the sample L2TP CLI file, **l2_lac.txt**, on the installation CD as your model and edit it to fit your situation.

Enter these commands at the client end (remote telecommuter):

```
# Define a remote named lnsserver

remote del lnsserver

remote add lnsserver

remote disauthen lnsserver

remote setoursysname lacclient lnsserver

remote setourpasswd clientpassword lnsserver

remote setLNS tunnelAtWork lnsserver

remote addiproute 192.168.100.0 255.255.255.0 1 lnsserver

# Set up a tunnel named tunnelAtWork
```

```
l2tp add tunnelAtWork

l2tp set chapsecret tunnelsecret tunnelAtWork

l2tp set ourtunnelname tunnelAtHome tunnelAtWork

l2tp set address 192.168.110.1 tunnelAtWork
```

Enter these commands at the LNS end (corporate site) for each teleworker:

```
# Define a remote named lacclient for the tunnel

remote del lacclient

remote add lacclient

remote setpass clientpassword lacclient

remote setLAC tunnelAtHome lacclient

remote setauthen chap lacclient

remote addiproute 192.168.101.0 255.255.255.0 1 lacclient

# Define a tunnel named tunnelAtHome.

l2tp del tunnelAtHome

l2tp add tunnelAtHome

l2tp set chapsecret tunnelsecret tunnelAtHome

l2tp set ourtunnelname tunnelAtWork tunnelAtHome
```

Troubleshooting from the Client (Remote) End

1. Ping the public port of the LNS. For example:

   ```
   ping 192.168.110.1
   ```

2. If this fails, enter the command **traceroute** to display the route and then fix the problem. It could be that your service provider or a firewall blocks the ping (port 15xx and 15xx need to be open). Or your company router might need a route defined back to the LNS-defined network.

3. Have someone monitor the LNS router to see if your tunnel call is coming in. Look for password errors or lack of "call from" messages. For example:

   ```
   4/04/2001-07:48:06:PPP: call from <Chuck> accepted via CHAP on L2TP/2001
   ```

4. To bring up a tunnel, use the command **l2tp call** *tunnelname*, or ping an address on the tunneled network. For the example above, you would enter:

   ```
   ping 192.168.101.1
   ```

5. If the tunnel starts, but you see password errors, fix them and then either **restart** the remote or **reboot** the router.

Troubleshooting from the LNS Router

1. Open a Telnet connection to the LNS router and enter the command **system log start** to see the console messages.

2. After a tunneling attempt, look for console messages like:

   ```
   04/04/2001-07:48:06:PPP: call from <Chuck> accepted via CHAP on L2TP/2001

   04/04/2001-07:48:06:DOD: link to Chuck over L2TP/2001 is now up

   04/04/2001-07:48:36:L2TP: Closing tunnel-2 to <Chucks_Tunnel> - NORMAL CLOSE/0
   ```

   If messages like these are not present, a firewall may be blocking the call.

3. Check your routing table on the LNS.

```
# iproutes
    IP route   /  Mask   --> Gateway         Interface    Hops Flags
0.0.0.0        /00000000 --> 172.17.1.200    ETHERNET/0   1 NW FW PRM RP1 RP2
172.17.1.0     /ffffff00 --> 0.0.0.0         ETHERNET/0   1 NW FW DIR PRM RP2
172.17.1.5     /ffffffff --> 0.0.0.0         ETHERNET/0   0 ME
172.17.19.2    /ffffffff --> Chuck           MLPPORT/50   1 FW PRM DOD PRV
172.17.19.7    /ffffffff --> jeff            [down]       1 FW PRM DOD PRV
224.0.0.18     /ffffffff --> 0.0.0.0         [none]       0 ME
224.0.0.122    /ffffffff --> 0.0.0.0         [none]       0 ME
255.255.255.255/ffffffff --> 0.0.0.0         [none]       0 NW PRM
```

   Each L2TP client should have a line in this table. In this example, the L2TP clients are jeff and Chuck. Note that Chuck's tunnel is up and jeff is down.

4. If the route table appears correct, ping the client L2TP address. For example:

   ```
   ping 172.17.19.7
   ```

   You should see messages like the following:

   ```
   04/06/2001-14:08:24:DOD: connecting to jeff over L2TP/2001
   04/06/2001-14:08:24:PPP: using bi-directional authentication with remote <jeff>
   04/06/2001-14:08:25:DOD: link to jeff over L2TP/2001 is now up
   ping: 172.17.19.7 - no response
   ping: 172.17.19.7 - no response
   ping: 172.17.19.7 - no response
   ping: 172.17.19.7 - no response
   ping: 172.17.19.7 - no response
   ping: packets sent 5, packets received 0
   ```

5. You can call the client router with the **l2tp call** command, but you need to set an IP address for the client first in the LNS using the command **l2tp set address** <*ipAddr*> <*TunnelName*>. For example:

   ```
   l2tp set address 192.168.53.225 jeffs_tunnel
   remote restart jeff
   ```

   Use the **iproutes** command to check the new entry in the route table:

   ```
   192.168.53.225    /ffffffff --> 172.17.1.200    ETHERNET/0   1 FW PRM PRV
   ```

Now issue the **l2tp call** command to see if there are password errors. For example:

```
# l2tp call jeffs_tunnel
04/06/2001-14:07:05:L2TP: tunnel-1 to <jeffs_Tunnel> opened
```

6. Use a **traceroute** command to the client WAN address to check that your company routers can access the public address of the client router.

```
# traceroute 192.168.53.225
         1: 172.17.1.200
         2: 172.17.1.100
         3: 12.39.98.101
         4: 12.124.40.65
         5: 12.123.13.170
         6: 12.122.5.150
         7: 12.123.13.65
         8: 12.123.221.2
         9: 207.88.240.113
        10: 64.220.0.17
        11: 64.0.0.98
        12: 198.68.76.55
        13: 205.158.11.26
        14: reply from 192.168.53.225: bytes=56 (data), time=54 ms
traceroute: packets sent 14, packets received 14
```

7. If you have another tunnel, ping that address to check that the company LAN is ok.

8. This worked so, something is wrong with Jeff's configuration. Telnet to the box to check his settings. Do a save and reboot on all routers to be saved.

9. Enter the command **l2tp list**. The following shows the display for an active tunnel:

```
INFORMATION FOR <jeffs_Tunnel>
  type................................ L2TPClient (LAC-will not dial
out)/LNS
  All Incoming Calls Tunneled here..... no
  CHAP challenge issued................ yes
  hidden AVPs used..................... yes
  sequencing/pacing.................... sequencing
    sequencing/pacing is............... required
    sequencing/pacing negotiation...... rfc compliant
    window size for sequencing/pacing.. 20
  ip address........................... 192.168.53.225
  tunnel timeout in seconds............ default (60)
  Our host name........................ efficient
  Our tunnel name...................... jeffs_Tunnel

  ACTIVE TUNNEL........................ jeffs_Tunnel
```

```
current state..................... ESTABLISHED
LOCAL TUNNEL ID................... 2
REMOTE TUNNEL ID.................. 2
remote vendor..................... Efficient 5871 IDSL Router(120-5871-001/2)
remote firmware................... 0
remote ip address................. 192.168.53.225
LAC SESSION serial number......... 1
  current state................... ESTABLISHED
  LOCAL CALL ID................... 1
    local window size.............. 20
    sequencing/pacing.............. SEQUENCING
      sequencing/pacing is......... required
  REMOTE CALL ID.................. 1
    remote window size............. 0
```

Dial Backup Troubleshooting

The Dial Backup feature is described in the section Dial Backup, on page 109. If you have Dial Backup problems, the following additional information may be helpful.

Sample Init String Settings

Use Hyperterminal directly connected to the modem to check the modem **init** string before connecting the modem to the router. The following are some example **init** strings.

Default (for USR Sportster 28.8k):

```
system modem init ATS0=0Q0V1&C1&K1X4&H1&I0S12=20
```

For Supra Express 56k:

```
system modem init ATS0=0Q0V1&C1X4L3S12=20
```

For Zoom 56k:

```
system modem init ATS0=0Q0V1&C1&D0X4L3S12=20
```

For ISDN TA Motorola Bitsurfer:

```
system modem init ATS0=0Q0V1&C2&D0X2S12=20
```

For ISDN TA 3Com Office connect:

```
system modem init ATS0=0Q0V1&C1&D0%C0X2s71=1s84=0
```

Operational Stability

If pings are failing, lower the success rate. For example, the following command lowers the success rate to 25%:

system backup successrate 25

Or, eliminate pinging as a failure criteria; use the DSL physical layer is the only failure criteria. To do so, enter this command:

system backup delete all all

Note that a Dial Backup session on the modem should time out after the PPP timer expires. When the Dial Backup **retry** timer expires the modem is disconnected even if there is traffic on the modem.

Debugging Procedures

When Dial Backup is enabled, the console port cannot be used to view log messages. So, to see messages, Telnet to the unit and enter the command:

system log start

Use the Windows GUI Port Monitor to display the line status. Other useful commands for monitoring Dial Backup status include:

ifs Shows status of all interfaces.

ipRoutes Shows current routes in IP routing table.

system list Shows ping attempts and success rates.

To temporarily stop Dial Backup, use these commands:

remote disable <*remoteName*> Stops modem dialing (specify the Dial Backup remote entry).

system backup disable Turns off Dial Backup.

System Messages

System messages are displayed on the terminal and sent to a log file (if you have opened one). The messages listed in this section are time-stamped informational and error messages. The messages are in the following format:

dd+hh:mm:ss:nn sysfunc: message

dd	date in xx/xx/xx format as specified during router initialization
hh:mm:ss:nn	time in military format (hours:minutes:seconds:hundredths of seconds)
sysfunc	software function
message	message

The following are examples of messages:

```
12/05/1997-16:31:17:ADSL: Startup initiated
12/05/1997-16:36:26:ADSL: Startup handshake in progress
```

Time-Stamped Messages

<router/user> didn't negotiate our IP address correctly

Explanation: The remote router did not negotiate the IP address options as was expected by the local router.

<router/user> terminated IPCP prematurely

Explanation: IP failed to negotiate. Try to change the remote or the source WAN IP address.

Far Avg SQ #: <2-digit number> dB [4-digit number]

Explanation: Message about the average signal quality for the remote router. This information appears during modem startup and should be ignored unless requested by Technical Support.

Authorization failed

Explanation: PAP cannot be negotiated.

Can't agree with <router/user> on what their IP address should be

Explanation: The IP address entry for the remote router in the remote router database does not match what the local router expects.

Can't obtain an IP address from <router/user>: one is needed in single user mode

Informative message.

Can't supply an IP address to <router/user>

Explanation: The remote end requests an IP address from the local end, which cannot supply it.

Cannot remove SYSTEM.CNF

Informative message.

Connecting to <router/user> @ <number> over <link/number>

Explanation: The local router is trying to connect to the specified remote destination.

Data Mode

Explanation: The connection is established and operational.

Duplicate IPX route to <router/user>

Explanation: There exist two routes to the same IPX destination. Remove one of the routes.

Duplicate IPX SAP <SAP number> to <router/user>

Explanation: There exist two IPX SAPs for the same IPX destination. Remove one of the SAPs.

Duplicate route <IP route> found on remote <router/user>

Explanation: There exist two IP routes to the same IP destination. One route needs to be removed.

Idle

Explanation: Data is not being transmitted.

IP is configured for numbered mode with <router/user>, but no address for it

Explanation: On one end of the connection, remote entries have been configured for numbered mode. On the other end, remote entries have been configured for unnumbered mode. Neither end cab communicate with the other.

No Signal Detected -- Check WAN Cable!

Explanation: (SDSL-specific error message) Your SDSL router cannot establish connectivity. Check your physical line.

No system name known - using defaults

Explanation: The router does not have a system name. For PAP/CHAP negotiation, the router will use a default name and password.

Note: IPX is misconfigured for <router/user> - no IPX WAN network

Explanation: IPX WAN address is wrong or missing.

Note: There is no IPX route statically defined for <router/user>

Informational message.

PPP: Peer not negotiating <IP | BNCP | IPX | CCP> right now

Explanation: One end of the network is not negotiating the same protocol as the other end.

Remote <router/user> didn't accept our CHAP password

Informational message.

Remote <router/user> does not respond to LPC echo. Link closed

The connection was terminated.

Remote <router/user> on <channel> didn't authenticate in time

Explanation: PPP authentication protocol did not succeed.

Remote <router/user> refuses to authenticate

Informational message.

Remote <router/user> tried to use PAP when CHAP was expected

Explanation: The remote end negotiated PAP while its minimum security level in the remote database was set to CHAP.

Remote <router/user> used wrong password <CHAP | PAP>

Explanation: The remote end has used an invalid password during CHAP or PAP security authentication.

Remote didn't accept our CHAP password

Explanation: The router attempted CHAP security authentication but the remote end rejected the password.

Remote on <interface> didn't authenticate in time

Informational message.

Remote on <interface> rejected our password with PAP

Informational message.

Remote on <interface> refuses to authenticate with us

Explanation: The remote destination refused to participate in the PAP/CHAP authentication process.

Startup failed

Explanation: The ATM modem could not synchronize with the remote end. Call Technical Support.

Startup failed: failure code = <number>, Status [code]

Explanation: The ATM modem could not synchronize with the remote end. Call Technical Support

TelnetD

Explanation: Connection accepted. A remote configuration session has been established.

User <router/user> is disabled in remote database

Informative message.

User <router/user> not found in remote database <PAP | CHAP>

Explanation: The authentication is coming from an unknown remote router.

Debugging Commands

The following commands may be available for debugging purposes. Please use them with caution because they are not fully supported.

General Debug Commands

ifs

Shows which interfaces are configured or active. For an example of its output, see page 214.

mlp debug <LCP | NCP | BNCP | IPCP | IPXCP | CCP | ECP | MLP | AUTH | NCPSTATES> [<0>]

BNCP is for bridging, CCP is for Compression Control Protocol, ECP for encryption, and NCPSTATES for state table changes.

To turn off the trace, enter the command with the optional **0** at the end.

ipdebug icmp 1

ipdebug nat 1

These commands show data received. The **ipdebug icmp 1** command is useful for showing the router can receive cells ok.

dod whycall 80

Prints out the packet that is causing the link to come up. This is useful when **system onewan on** is set. (This command makes PVC's look like dial-up links, that is, the link comes up only if user traffic exists and the link times out on inactivity.) For more information, see SYSTEM ONEWANDIALUP, on page 251.

dod debug <1 | 0>

Shows trace of when we bring up the link or time out link on inactivity. Specify 1 to turn on the trace; specify 0 to turn off the trace.

ping [-c *count*] [-i *wait*] [-s | -l *size*] [-I *sourceipaddr*] <*ipaddr*> | <*domainname*>

Sends an echo message to the specified IP address or domain name. You cannot ping your own LAN address; you can ping your own WAN address.

You can set the length of user data down to 0 bytes (**-s 0** or **-l 0**) so in routing mode it fits in one ATM cell. (See page 217.)

traceroute [-c *count*] [-i *wait*] [-s | -l *size*] [-I *sourceipaddr*] [-n] <*ipaddr*> | <*domainname*>

Traces the route taken by packets sent from the local router to the specified IP address or domain name. A packet is sent for each hop in the route. The output lists the IP addresses of the hops that returned packets. (See page 224.)

system log [start | stop | status]

Starts event logging when logged in via Telnet. Otherwise, you don't see any event messages. It is not needed if you are using a console cable. (See page 249.)

system supporttrace

Dumps all tables. If you capture and send this output to Technical Support, it can be useful in debugging problems. For more information, see SYSTEM SUPPORTTRACE, on page 253.

The information dumped includes the history log and information about the version, memory, processes, the file system, general system information, Ethernet, DHCP, Voice, remote database, interfaces, bridging, the ARP table, IP routes, IPX routes, IPX SAPs, L2TP tunnels, and IP filters.

ATM Debug Commands

atm reset

Re-initializes the ATM-25 link.

atom findPVC <on | off>

Shows VPI*VCI of cells received. This command is normally used to find the ATM VPI*VCI number necessary for configuring a remote when the Service Provider either has supplied the wrong value or simply is not able to supply one.This command should only be used when there are *no* remotes defined or when the remote entries are disabled.

The command output is directed to the console. If Telnet is used to log into the router, then issue the **system log start** command to direct the console output to the Telnet session.

Example:

```
# atom findPVC on
No remote entry found with PVC (VPI*VCI) 1*2
```

In this case, an ATM VPI*VCI is found for which there is no remote defined. 1 is the number of the VPI as found in the ATM stream. 2 is the number of the VCI as found in the ATM stream. The discovered number may be used as the VPI*VCI value in the remote, for determining whether communications are possible.

atom echoPVC <vpi number>*<vci number>

Enables an echo PVC (use **atom echo 0*21**). This is configured automatically and can be disabled with **atom echo 0*0**. The echoPVC will echo back any ATM cell received on the PVC exactly as received. This is useful when an administrative service wishes to ensure ATM connectivity but cannot use ATM OAM F5 cells to achieve this function.

atom dumpunknowncells [on | off]

Without its parameter, the command indicates whether unknown cell tracing is on or off. Set to on, the trace looks at the content of an ATM cell. It will not affect normal operation performance.

atom pls <on | off>

Changes payload scrambling.

atom empty <ATMF | ITU>

Changes type of ATM empty cell sent or expected. It is useful if ATM sync delineation errors when combined with **atom stats** command.

atom nma

States the non-matching address count.

Web GUI Debug Commands

If you point your web browser to **http://192.168.254.254/tools/index.html**, you can display an index to special pages in the web GUI. These pages include:

dump.html State variable dump (for debugging purposes)

access.html Control router administrative access.

editor.html Edit files in the router file system.

routing.html Edit the static routing table for an interface.

features.html Display and modify feature list.

password.html Change administrative password.

newpass.html Password redirection page

strings.html String table for the tools module

time.html Set router clock.

reboot.html Reboot the router.

default.html Reboot the router, restoring to defaults.

factory.html Reboot the router, erasing all configuration information.

SDSL Debug Commands

sdsl * Displays all available SDSL commands.

sdsl btstat * Displays available SDSL status commands.

sdsl btstat Displays available status values. For example:

```
# sdsl btstat
Available status:
   SLM ........................... Input Signal Level
   DC_METER ...................... Input DC Offset
   FELM .................. Far-End Signal Attenuation (Cal'd at 1168 Kbs)
   NMR ........................... Noise Margin
   TIMING_RECOVERY_CONTROL ....... Timing Recovery Control
   STARTUP_STATUS ................ Bit-Pump Status
   BIT_PUMP_PRESENT .............. Bit-Pump Present
   SELF_TEST ..................... Self Test
   REGISTER ...................... Read Register
   CONFIGURATION ................. Big-Pump Configuration
   STAGE_NUMBER .................. Stage Number
   AAGC_VALUE .................... AAGC
```

```
READ_TX ..................... Read Tx Gain
BER_METER_STATUS ............. BER Meter Status
```

sdsl bts felm Displays Far-End Signal Attenuation. It gives an estimate of the length of the loop.

Output example: `SDSL: FELM: 63 [0x3f]`

sdsl bts nmr Displays noise margin. Large values are symptoms of a bad or excessively lengthy loop.

Output example: `SDSL: NMR: 224 [0xe0]`

sdsl states trace [<all>]

Turns on trace of line changes. To turn off the trace, append **all** to the command.

Example:

```
# sdsl states trace
SDSL State Trace [00000001]: states => s
# sdsl states trace all
SDSL State Trace [00000000]: off
```

sdsl huh Dumps various registers.

Example:

```
# sdsl huh
SDSL:
    Bitpump: 8973
    CPE -- ACTIVATING
    Line Rate: [AUTO] 192 Kb/s  [3072 KHz]
    Activation Interval: 99 [AUTO:20]  [symbol_rate: 24]
    AutoSpeed:
      FastSearchAttemptsPerPass: 2
      FastSearchPasses.........: 2
      SlowSearchAttemptsPerPass: 5
      SaveDelayInSeconds.......: 45
    Two Symbol Time: 23 uS
    FW: V4.3    CS 5: BR = 80000401  OR = fffff8f66
    Ints -- On : 1228462  Mask: 0b00  IRQ: 02
    BP Status Reads: 0
    BT assumed on other end!
    BT - Self Test will run
    SDSL CONFIGURATION: 0x03f9  20 LOST: 10 [0x0a] Sym Rate: 24 [0x18]
```

ADSL DMT Router Debug Commands

A command sequence to disable the failover pots interface of the 7461 ADSL router is shown under Voice Routing (VoDSL) Troubleshooting, on page 191.

dmt * Displays the available DMT commands.

dmt link <DEFAULT | T1_413 | G_DMT | G_LITE | MULTIMODE>

Sets the link type. It is used to force the CPE into ANSI (T1.413), G_DMT, or G_LITE mode. DEFAULT and MULTIMODE are the same. The link type survives reboots.

dmt log Prints the log file.

dmt ms Shows the modem status.

dmt speed Displays the speed of the link.

dmt vers Displays the code version of line driver. The following is an output example :

```
Version:
     FW: dmt-nt.bin -- 28 May 100 10:05  [249176]  3.6.70
     ATU-R: 255 [0xff]    ATU-C: Not Available
```

ADSL modem timer commands:

The timer is started when the modem tries to activate and is stopped after a successful activation, or when it expires, whichever comes first. This “stuck” condition increments the retry counter. If this "stuck" condition occurs the allowed number of retries (consecutive, or not), the modem is reset (and the retry count reset to 0.)

dmt retries <*n*> Sets number of activation failures before the modem is reset (1 - 10000). The default is 10.

dmt to <sec> Sets timeout timer (30 to 3600 seconds). The default is 45 seconds.

The timeout changes take effect immediately and are not saved to flash memory. Save your changes if you want to keep them after the next power cycle.

Frame Relay Debug Commands

frame stats Displays statistics. Although it is not an end-to-end loopback test, it does show counters for data sent and received as well as LMI events. For an output example, see page 334.

ATM Tracing Commands

atom print Shows count of good and bad atm cells and frames.

atom rx <on | off> Shows AAL5 frames received.

atom promisc on Turns on promiscuous mode (rx ATM cells no matter what VPI*VCI).

atom cellrx <on | off>Traces ATM cells received.

atom tx <on | off> Traces ATM cells sent.

atom stats <*n*> Prints the ATM statistics every *n* seconds. It shows good and bad cells and frames.

IP Filtering Debug Commands

The following commands can start and stop an IP filter watch. For more information about IP filter watch, see the command descriptions on page 270 and page 300.

eth ip filter watch <on | off>

remote ipfilter watch <on | off>

Prints a message to the console if a packet to or from this remote is dropped or rejected.

IKE Debug Commands

If packets are not being processed correctly across an IPSec tunnel, enter this command so that the commit bit is set:

ike commit on

Setting the commit bit makes sure that no IPSec traffic arrives at the router before the router is ready for it.

The following commands allow you to start and stop an IPSec policy.

ike start <*PolicyName*>

ike stop <*PolicyName*>

Before Contacting Technical Support

Before you contact Technical Support, please have the following information ready:

- Router model number
- Router software version
- Date of purchase
- Type of operating system (Windows 95, 98, NT, or Windows for Workgroups)
- Description of the problem
- List of other equipment such as personal computers, modems, etc. and third-party software you are using, including revision levels.

To determine how to contact Technical Support, see the *User Reference Guide* and *Customer Release Notes* that came with your router or refer to the web site www.efficient.com**.**

Chapter 8. Command Reference

This chapter lists the formats of the commands you can enter on the router command line.

To see a specific command description, use the command index at the end of the manual (page 411).

The commands are organized alphabetically, in the following sections:

- Status commands
- File system commands
- Local router commands (**system** and **eth**)
- Remote router commands:

remote	**dual-ethernet**	**idsl**
adsl	**frame**	**sdsl**
atm	**hdsl**	**shdsl**
dmt		

- DHCP commands
- L2TP commands
- Bridge filtering commands
- PPPoE commands
- IPSec commands (**ike** and **ipsec**)

Command Conventions

The Command Line Interface (CLI) follows these conventions:

- Command line length may be up to 120 characters long.
- The Command Line Interface is not case-sensitive except for passwords and router names.
- All parameters are positional; i.e., each keyword/parameter must be entered in the correct order, as shown in the command format in this manual.

The command formats shown in this manual follow these conventions:

- Items that appear in **bold** type must be typed exactly as they appear.However, commands can be shortened to just those characters necessary to make the command unique.
- Items that appear in *italics* are placeholders representing specific information that you supply.
- Parameters enclosed in the characters < and > must be entered.
- Parameters enclosed in the characters [and] are optional.

Sample command responses are shown in this chapter. In many cases, only the command prompt **#** is returned. If you have not entered the correct parameters, the syntax of the command is displayed.

? OR HELP

To see the available top-level commands, enter **?** or **help**. To see the subcommands for a top-level command, enter the top-level command followed by a **?**. To see the syntax of a subcommand, enter the subcommand followed by a **?**.

Note: If the first parameter for a command is a character string, the **?** will be taken as the character string if entered in that position.

? or **help**

Examples:

```
# ?
Top-level commands:
?                   help              version
filter              logout            exit
reboot              mem               ps
copy                dir               delete
rename              execute           format
sync                msfs              ifs
date                time              ipifs
iproutes            arp               ipxroutes
pxsaps              bi                system
eth                 save              erase
key                 remote            call
ping                traceroute        tcp
dhcp                l2tp              pppoe
ipsec               ike               atom
dmt

# time ?
Usage: time HH:MM:SS
```

Status Commands

The commands in this section are online action and status commands. They allow you to perform the following functions:

- log into and log out of configuration update mode
- display the router’s configuration, the version and level numbers
- list running tasks, memory, and communication interfaces
- connect to a remote router to test the line
- list IP routes, IPX routes and SAPs, and root bridge
- save the new configuration image
- reboot the system

ARP DELETE

Deletes the IP address of the entry in the ARP table.

arp delete <*ipaddr*> | all

ipaddr IP address in the format of 4 decimals separated by periods.

all Deletes all existing arp table entries

Example: `arp delete 128.1.2.0`

ARP LIST

Lists Address Resolution Protocol (ARP) table entries in an IP routing environment. ARP is a tool used to find the appropriate MAC addresses of devices based on the destination IP addresses.

arp list <*ipaddr*> <*InterfaceName*> <*InterfaceUnit*>

ipaddr IP address associated with a MAC address for a device on the local interface in the format of 4 decimals separated by periods.

InterfaceName MAC address on the local network

InterfaceUnit For an Ethernet interface, this can be a 1 or 0. For a DSL interface, this is a VPN number.

Example: `arp list`

Response:

```
IP Addr              Mac Address              Interface
192.84.210.148       00:05:02:00:80:A8        ETHERNET/0
```

BI

Lists the root bridge, and indicates whether the router is learning, listening, or forwarding.

bi

Response:

```
# bi
GROUP  0Our  ID=8000+00206f0249fc  Root  ID=8000+00206f0249fc
Port ETHERNET/0              00+00 FORWARDING
```

BI LIST

Lists the contents of the bridge table.

Each MAC address in the table is listed with its corresponding bridge port as learned by the bridge function. The line also shows the number of seconds elapsed since the last packet was received by the MAC address followed by flags. Possible flags include:

P	Permanent (This entry is not aged out of the table.)
FLD	Flood
US	This entry is for the target router.
A	Accept
FWD	Forward
BC	Broadcast
MC	Multicast

.

bi list

Example:

```
# bi list
BRIDGE GROUP 0:
00206F024C34:                    P     US    SD A
0180C2000000:                    P              A         MC
FFFFFFFFFFFF:                    P FLD          A          BC MC
02206F02E70D: ETHERNET/0      325                  FWD
00C04F2E1AEB: ETHERNET/0      143                  FWD
0060081BD761: ETHERNET/0       95                  FWD
```

CALL

Dials a remote router. This command can be used to test the ISDN link or L2TP secession and the configuration settings for the remote router.

call <*remoteName*>

Response:

```
# Request Queued
```

DATE

Displays or changes the current date on the router's clock. To change the current time, use the command **time** (page 224).

Automatic SNTP requests are generated if the system needs to get the time. You can specify an SNTP server using the command **sntp server** (page 223) and a UTC offset with the command **sntp offset** (page 221).

To see the current date and time on the router clock, enter **date** with no parameters.

date <*mm/dd/yy*>

mm Month (1 - 12).

dd Day of the month (1 - 31).

yy Year (1-4 digits, indicating a year from 1968 through 2034). Thus, 1/1/4 is January 1, 2004, 1/1/33 is January 1, 2033, and 1/1/78 is January 1, 1978.

Example:

```
# date
BootTime:      5/1/2001 at 15:42:42
Current time:  5/1/2001 at 15:52:49

# date 5/2/1
Time set to UTC-420, 5/2/2001 at 15:52:49
Time adjusted for (-) 0 days 11 hours 49 minutes 34 seconds
```

ERASE

The **erase** command erases the entire router's configuration or parts of it from FLASH memory.

You will need to completely reconfigure any part of the configuration that you erase.

Note: An **erase** command does not take effect until after a **reboot** *without* a **save** command

Note: There is a time lag between the response issued by the **erase** command and the time that the data is actually deleted from FLASH memory. Issue a **sync** command after an **erase** command before powering off the router. This commits the changes to FLASH memory.

erase all | keys | dod | sys | eth | filter | ipsec | ike | atom | sdsl | idsl | frame | dhcp | atm25 | l2tp | sntp

Examples:

`erase`	Same as **erase all**.
`erase all`	Erases the entire router configuration from FLASH memory, including settings for the system, Ethernet LAN, DSL line, DHCP, and remote router database.
`erase atom`	Erases the ATM configuration settings.
`erase dhcp`	Erases the DHCP configuration settings from FLASH memory. To clear all DHCP information without erasing FLASH memory, use the command **dhcp clear all records** (page 354).
`erase dod`	Erases the current state of the remote router database.
`erase eth`	Erases the configuration settings for the Ethernet LAN from FLASH memory.
`erase filter`	Erases the current bridging filtering database from FLASH memory.When you issue this command you must reboot (*without* a **save**).
`erase keys`	Erases the software option keys from FLASH memory.
`erase sys`	Erases the name, message, and authentication password system settings from FLASH memory.

EXIT

Has the same function as **logout**, but will disconnect you from a Telnet session.

exit

IFS

Lists the communication interfaces installed in the router and the status of the interfaces.

ifs

Example:

```
ifs
Interface   Speed       In %     Out % Protocol     State    Connection
ETHERNET/0 10.0mb      0%/0%     0%/0% (Ethernet)   OPENED
SHDSL/0      384kb    50%/50%   50%/50% (ATM)        OFF
ATM-VOICE/1  384kb    45%/45%     0%/0% (ATM)        OFF
BACKUP/0      57kb     0%/0%     0%/0% (AHDLC/PPP)  OPENED   to backup
CONSOLE/0  9600 b      0%/0%     0%/0% (TTY)        OFF
VOX-STRM/0    0 b                    (CLEAR)      OFF
```

Additional interfaces on other routers could include:

```
FR/3        144kb      0%/0%     0%/0% (HDLC/FR)    OPENED
FR-VC/1     144kb     0%/12%     0%/2% (FR)         OPENED   to internet
DMT/0         0 b                    (ATM)        OFF
ATM-VC/1      0 b                    (ATM)        OFF
```

`Interface`	`ETHERNET`	LAN

	`SHDSL`
	`DMT`
	`FR`
		WAN physical layer

	`ATM-VC` `FR`	WAN layer 2 virtual circuit
	`BACKUP`	Dial Backup modem
	`ATM-VOICE`	Voice over DSL
	`CONSOLE`	Serial port
	`VOX-STRM`	Streaming voice control channel

`In% Out%` Downstream and upstream percentages. The first percentage is an instantaneous value taken every second. The second percentage is the weighted average over 5 seconds using the formula:

current avg = (4 * old average + instant value)/5

`Protocol` Protocol in use, such as frame relay (FR), asynchronous PPP (AHDLC/PPP), and serial (TTY).

`State`	Current state of the interface.	
	OFF	Down
	STANDBY	Being negotiated.
	OPENED	Physical interface operational.
	CONNECTED	Logical interface operational.

IPIFS

Lists the IP interface.

ipifs

`Response:`

```
ATM_VC/1            192.168.254.1 (FFFFFF00) dest 192.168.254.2 sub 192.168.254.0
                    net 192.168.254.0 (FFFFFF00) P-2-P
ETHERNET/0          192.84.210.12 (FFFFFF00) dest 0.0.0.0 sub 192.84.210.0
                    net 192.84.210.0 (FFFFFF00) BROADCAST mtu 1500
```

IPROUTES

Lists the current entries in the IP routing table.

iproutes

`Response:`

```
# iproutes
IP route          /  Mask     -->  Gateway    Interface    Hops Flags

0.0.0.0           /ffffffff   -->  0.0.0.0    [none]       0 NW PRIV
192.84.210.0      /ffffff00   -->  0.0.0.0    ETHERNET/0   1 NW FW DIR PERM
192.84.210.12     /ffffffff   -->  0.0.0.0    ETHERNET/0   0 ME
192.168.254.0     /ffffff00   -->  0.0.0.0    [none]       0 NW PRIV
192.168.254.1     /ffffffff   -->  HQ         ATM_VC/1     0 ME
192.168.254.2     /ffffffff   -->  HQ         ATM_VC/1     1 FW DIR PRIV
224.0.0.9         /ffffffff   -->  0.0.0.0    [none]       0 ME
255.255.255.255   /ffffffff   -->  0.0.0.0    [none]       0 NW PERM
```

Where:	NW	Network
	PERM	Permanent (static)
	DOD	Initiate link dial-up
	FW	Forward
	DIR	Direct
	ME	This router

IPXROUTES

Lists the current entries in the IPX routing table.

ipxroutes

Response:

```
# ipxroutes
Network       Gateway     Interface       Hops  Ticks  Flags
00001001:     HQ          [down]          1      4      STATIC FORWARD DOD
00000456:     (DIRECT)    ETHERNET/0      0      1      FORWARD
```

where:	STATIC	Static route
	DOD	Initiate link dial-up
	FORWARD	
	DIRECT	

IPXSAPS

Lists the current services in the IPX SAPs table.

ipxsaps

Response:

```
# ipxsaps
Service Name     Type  Node number Network   Skt    Hops
SERV312_FP         4   000000000001:00001001:045    1
```

LOGOUT

Logs out to reinstate administrative security after you have completed changing the router's configuration.

logout

MEM

The mem command report the amount of ram installed in the router.

mem

Response:

```
# mem
Small buffers used.......18         (7% of 256 used)
Large buffers used.......41         (16% of 256 used)
Buffer descriptors used..59         (7% of 768 used)
Number of waiters s/l....0/0

Table memory allocation statistics:
Sizes  16     32     64     128    256    512    1024   2048
Used   34     18     12     3      8      9      8      7
```

```
Free   3      1      4      0      1      1      1      1

Sizes  4096   8192
Used   3      1
Free   1      0
Total in use: 51936, total free: 857368 (8272 + 849096)
```

MLP SUMMARY

Lists the status of the protocols negotiated for an active remote connection. The following are the most common protocols:

- MLP (Multilink Procedure)
- IPNCP (IP routing Network Protocol)
- CCP (Compression Control Protocol)
- BNCP (Bridging Network Protocol)
- IPXCP (IPX Network Protocol)

Open indicates that the protocol is in ready state.
Stopped means that the protocol is defined, but did not successfully negotiate with the remote end.
No message means that the link is not active.

mlp summary

Example: `mlp summary`

PING

Sends an echo message, available within the TCP/IP protocol suite. The echo message is sent to a remote node and returned; the echo tests connectivity to the remote node. It is particularly useful for locating connection problems on a network.

The remote node can be specified by IP address or by domain name. If a domain name is specified, the address of the domain is requested from the domain name server (DNS).

A status message is issued for each echo message sent.

Note: You cannot ping your own LAN address; you *can* ping your own WAN address.

To fit the echo message into one ATM cell in routing mode, set the length of user data down to 0 bytes (**-s 0** or **-l 0**).

Note: To terminate the ping before it ends, press control-c.

ping [**-c** *count*] [**-i** <*wait*>] [**- s** | **-l** <*size*>)] [**-I <***srceaddr*>] <*ipaddr*> | <*domainname*>

-c *count* Number of packets sent (from 1 to 2000000000). The default is 5 packets.

-i *wait* Wait period between packets in seconds (from 1 to 10). The default is 1 second.

-s *size* Packet data length in bytes (from 0 to 1648). The default is 56 bytes.

-l *size* Same as -s *size.*

-I *srcaddr* Source IP address contained in the echo message (4 decimals separated by periods). Use this option to force packets into a tunnel or to force use of the management address as the source address.

ipaddr *domainname* Remote node to which the echo message is sent. It can be specified by its domain name or by its IP address (4 decimals separated by periods).

Examples:

The following command pings the domain name www.yahoo.com.

```
# ping www.yahoo.com
```

The command attempts a DNS (domain name server) lookup to find the address of the domain. If the DNS server address is not known, it returns the following message:

```
ping: unknown host www.yahoo.com
```

If the DNS lookup is successful, the ping sends five packets, one second apart, with a packet length of 56 bytes.

```
ping: reply from 216.32.74.52: bytes=56 (data), icmp_seq=1, time=86 ms
ping: reply from 216.32.74.52: bytes=56 (data), icmp_seq=2, time=81 ms
ping: reply from 216.32.74.52: bytes=56 (data), icmp_seq=3, time=82 ms
ping: reply from 216.32.74.52: bytes=56 (data), icmp_seq=4, time=84 ms
ping: reply from 216.32.74.52: bytes=56 (data), icmp_seq=5, time=82 ms
ping: packets sent 5, packets received 5
```

The following command requests 2 echo messages sent 7 seconds apart with a packet length of 34 bytes. The messages are sent to IP address 192.168.254.2.

```
# ping -c 2  -i 7 -s 34 192.168.254.2
ping: reply from 192.168.254.2: bytes=34 (data), time<5 ms
ping: reply from 192.168.254.2: bytes=34 (data), time<5 ms
ping: packets sent 2, packets received 2
```

The following command sends packets with the source IP address 192.168.254.254 to the IP address 192.4.210.122. Default values are used for the other options.

```
ping -I 192.168.254.254 192.4.210.122
```

The following command uses management address 192.168.1.2 as the source address when pinging destination address 192.168.100.100.

```
ping -I 192.168.1.2 192.168.100.100
```

PS

Lists all of the tasks (processes) running in the system and the status of the tasks.

ps

Response:

```
# ps
```

```
TID:              NAME     FL  P   BOTTOM    CURRENT SIZE
1:IDLE                     02  7   1208f0    121008   2032
3:MSFS_SYNC                03  6   1224a0    122ba8   2032
4:SYSTEM LOGGER            03  5   122cd0    1233d8   2032
5:LL_PPP                   03  5   126750    126e58   2032
6:NL_IP                    03  5   126fe0    1272e0   1000
7:TL_IP_UDP                03  3   127460    127768   1000|
8:TL_IP_TCP                03  3   1278c0    127fd0   2032
9:IP_RIP                   03  4   128120    128420   1000
10:TELNETD                 03  5   128550    128838   1000
11:DUM                     03  5   12b580    12bc88   2032
12:ATM25                   03  1   12c0a0    12c790   2032
13:SNMPD                   03  5   124b60    125a70   4080
14:BOOTP                   03  5   12e3d0    12e6c0   1000
15:CMD                     01  6   12cba0    12d9f8   4080
```

TID:	task ID field
NAME:	name of the task
FL:	flag field
P:	number from 1 to 7 with the highest priority equal to 1
BOTTOM:	address of the task stack
CURRENT:	current stack pointer
SIZE:	stack size in bytes

REBOOT

This command causes a reboot of the system.

Caution: A reboot erases any configuration changes that have not been saved. Remember to enter a **save** command before the **reboot** command.

Certain configuration settings *require* a **reboot** before the setting becomes effective, including:

- A change from IP routing to bridging or the reverse.
- The addition of IKE filters
- IPX changes

Other configuration changes become effective following either a **reboot** or a **restart** of the Ethernet or remote interface. These changes include:

- System settings
- Ethernet IP address
- TCP/IP routing
- Remote router default bridging destination
- TCP/IP route addresses
- SAPs and bridging
- Adding a new remote entry to the remote database.

A reboot also ensures that all file system updates are completed. There is a time lag between the entry of a **save** command and the safe storage of the data in FLASH memory. If the power goes off before the data is stored in memory, the data can be lost. *Always* reboot before powering off the router. Or, use the **sync** command (page 229) to commit file changes to memory.

reboot [<*option*>]

option If no option is specified, the router is rebooted using the existing configuration file.

default This option deletes the system configuration file and restores the router to its original defaults (before any configuration was entered).

Note: The word *default* must be fully *spelled out*.

factory This option deletes all files except AUTOEXEC.OLD if it exists. AUTOEXEC.OLD is renamed AUTOEXEC.BAT; it is re-executed by the reboot. This option also resets the non-volatile RAM; thus deleting the IP address of the router and the TFTP server during the boot process and also forcing the router to boot from FLASH instead of from the network.

SAVE

The **save** command saves the entire router's configuration or parts of it to FLASH memory. The keyword in the command determines what is saved.

Note: There is a time lag between the response issued by the **save** command and the time when the data is actually stored in FLASH memory. Issue a **sync** command after a **save** command before powering off the router. This commits the changes to FLASH memory.

save all | keys | dod | sys | eth | filter | ipsec | ike | atom | sdsl | idsl | frame | dhcp | atm25 | l2tp | sntp

Examples:

`save` Same as **save all**.

`save all` Saves the configuration settings for the system, Ethernet LAN, DSL line, and remote router database into FLASH memory.

`save atom` Saves the ATM configuration settings.

`save dhcp` Saves the DHCP configuration settings into FLASH memory.

`save dod` Saves the current state of the remote router database.

`save eth` Saves the configuration settings for the Ethernet LAN into FLASH memory.

`save filter` Saves the bridging filtering database to FLASH memory. A reboot *must* be executed to load the database for active use.

`save sys` Saves the name, message, and authentication password system settings into FLASH memory.

SNTP ACTIVE

Displays the *active* SNTP server, that is, the server that last responded to an SNTP request.

To see a list of SNTP servers or change the servers in the list, use the command **sntp server**. To display or change the *preferred* SNTP server, use the command **sntp prefServer**.

sntp active

Example:

```
# sntp active
Active SNTP server is 1 (192.6.38.127)
```

SNTP DISABLE

Disables SNTP requests.

To enable SNTP requests, use the command **sntp enable** (page 221).

sntp disable

Example:

```
# sntp enable
Current offset from UTC is 0 minutes
Use <system sntp offset> to set time zone
```

SNTP ENABLE

Enables SNTP requests.

To disable SNTP requests, use the command **sntp disable** (page 221).

sntp enable

Example:

```
# sntp enable
Current offset from UTC is 0 minutes
Use <system sntp offset> to set time zone
```

SNTP OFFSET

Specifies the SNTP offset from the Universal Time Coordinate (UTC).

The offset is specified in minutes. A *positive* offset is an offset to the *east* of the Greenwich meridian; a *negative* offset is to the *west* of the Greenwich meridian.

To see the current offset, specify **sntp offset** with no parameter.

Note: To make a change permanent, you must **save** the change before you **restart** or **reboot**.

sntp offset [<*minutes*>]

minutes Number of minutes east or west of the Greenwich meridian. A positive number is east; a negative number is west.

Example:

```
# sntp offset
Current offset from UTC is 0 minutes
usage: sntp offset <Minutes from UTC>
       (offset is negative for west, positive for east of Greenwich meridian)

# sntp offset -360
```

SNTP PREFSERVER

Displays or changes the *preferred* SNTP server. (The preferred server is the server that should be attempted first when a request is made.)

To specify a server preference, specify the number of the preferred server within the SNTP server list. To see the SNTP server list, enter **sntp server**.

To see the current *preferred* SNTP server, enter **sntp prefServer** with no parameter.

To see the *active* SNTP server (that is, the server that last responded to an SNTP request), use the command **sntp active**.

Note: To make a change permanent, you must **save** the change before you **reboot**.

sntp prefServer [<*number*>]

number Number of a server within the SNTP server list. To see the server numbers, enter **sntp server**.

Example:

```
# sntp server
Current server (1) IP addr: 192.6.38.127
Current server (2) IP addr: 192.5.41.40
Current server (3) IP addr: 192.6.38.127
Current server (4) IP addr: 209.81.9.7
Current server (5) IP addr: 129.7.1.66
Usage: sntp server <IP address>|default [server number]

# sntp prefserver
The preferred SNTP server is 1 (192.6.38.127)
# sntp prefserver 3
Preferred SNTP server is set to 3 (192.6.38.127)
```

SNTP REQUEST

Requests the time from an SNTP server. (SNTP is the Simple Network Time Protocol defined by RFC 1769.)

Note: A request is performed only if SNTP is enabled (see **sntp enable**, page 221).

To specify the IP address of an SNTP server, use the command **sntp server**.

sntp request

Example:

```
# sntp request
SNTP is currently disabled
# sntp enable
# sntp request
Time server IP address not set, use "sntp server w.x.y.z"
# sntp server
Current server (1) IP addr = 0.0.0.0
usage: sntp server <IP address>|default [server number]

# sntp server 12.101.4.1
# sntp request
Time set to UTC-480, 5/7/2001 at 17:29:25.245
Time adjusted for (-) 0 days 1 hours 0 minutes 0 seconds
```

SNTP SERVER

Displays or changes the SNTP server list.

- To *see* the current SNTP server list, specify **sntp server** with no parameter.
- To specify the *default* server list, specify **sntp server default**.
- To *add* a server to the list, specify **sntp server** with the server IP address and a new number for the entry.
- To *change* the address of a server, specify **sntp server** with the server IP address and the existing entry number.
- To *remove* a server from the list, specify **sntp server 0.0.0.0** and the number of the server to be removed.

Note: To make a change permanent, you must **save** the change before you **reboot**.

sntp server <*IPaddress*> | default [<*number*>]

IPaddress	IP address of an SNTP server (4 decimals separated by periods). (To remove a server, specify 0.0.0.0 as the IP address.)
default	Requests the default server list (see the example below).
number	Number of the server in the list. If that server number is already in the list, the IP address is changed; otherwise, a new entry is added to the list. If you omit a number, the IP address of the *active* server is changed.

Example:

```
# sntp server default
Current server (1) IP addr: 192.5.41.40
Current server (2) IP addr: 192.6.38.127
Current server (3) IP addr: 209.81.9.7
Current server (4) IP addr: 129.7.1.66
Current server (5) IP addr: 192.168.254.2
```

```
# sntp server 172.17.20.69 6
```

TCP STATS

Displays the TCP statistics and open connections.

tcp stats

Example: `tcp stats`

TIME

Displays or changes the current time on the router's clock. To change the current date, use the command **date** (page 212).

Automatic SNTP requests are generated if the system needs to get the time. You can specify an SNTP server using the command **sntp server** (page 223) and a UTC offset with the command **sntp offset** (page 221).

To see the current date and time on the router clock, enter **time** with no parameters.

time <*hh:mm:ss*>

hh Hour (0 - 23).

mm Minute (0 - 59).

ss Second (0 - 59).

Example:

```
# time
BootTime:      5/18/2001 at 11:57:12
Current time:  5/18/2001 at 12:00:01

# time 1:01:01
Time set to UTC-420, 5/18/2001 at 1:01:01.074
Time adjusted for (-) 0 days 11 hours 49 minutes 34 seconds
```

TRACEROUTE

Traces the route taken by packets sent from the local router to the specified IP address or domain name. A packet is sent for each hop in the route. The output lists the IP addresses of the hops that returned packets.

Unless the **-n** option is specified, **traceroute** also attempts to look up the name of each gateway in the route. If the DNS lookup is successful, the name is included in the output message.

Note: To terminate the traceroute before it completes, press **control-c**.

traceroute [**-c** *count*] [**-i** <*wait*>] [**- s**	**-l** <*size*>)] [**-I** <*srceaddr*>] [**-n**] <*ipaddr*>	<*domainname*>

-c *count* Number of packets sent (from 1 to 255). The default is 30 packets.

-i *wait* Wait period between packets in seconds (from 1 to 2000000000). The default is 1 second.

-s *size* Packet data length in bytes (from 0 to 1648). The default is 56 bytes.

-l *size* Same as **-s** *size*.

-I *srcaddr* Source IP address contained in the echo message (4 decimals separated by periods). Use this option to force packets into a tunnel or to force use of the management address as the source address.

-n Eliminates the DNS lookup for each hop. Only the IP address of the hop is listed in the output message.

ipaddr
domainname The end of the route, specified by a domain name or IP address (4 decimals separated by periods).

Examples:

The following two commands trace the same route. The first specifies the domain name; the second specifies the IP address.

```
# traceroute www.yahoo.com
# traceroute 204.71.200.68
```

Both commands send up to thirty packets with a wait period of one second and a packet length of 56 bytes. The following is an example of the command output:

```
          1: 172.17.20.122       l2tp-router.flowpoint.com
          2: 172.17.20.1         checkpoint.flowpoint.com
          3: 12.39.98.136        csco2.efficient.com
          4: 12.124.40.65
          5: 12.123.13.166       gbr5-p56.sffca.ip.att.net
          6: 12.122.5.142        gbr3-p100.sffca.ip.att.net
          7: 12.122.5.253        gbr2-p60.sffca.ip.att.net
          8: 12.123.13.61        gar1-p370.sffca.ip.att.net
         10: 206.132.150.250
         11: 206.132.254.37      ge0-0-1000M.hr8.SNV.gblx.net
         12: 206.178.103.62      baslr-ge3-0-hr8.snv.yahoo.com
         13: reply from 204.71.200.68: bytes=56 (data), time=18 ms
traceroute: packets set 13, packets received 12
```

For a faster route trace, specify the **-n** option to eliminate the domain name lookup.

```
# traceroute -n  204.71.200.68
          1: 172.17.20.122
          2: 172.17.20.1
          3: 12.39.98.136
          4: 12.124.40.65
          5: 12.123.13.166
          6: 12.122.5.142
          7: 12.122.5.253
          8: 12.123.13.61
         10: 206.132.150.250
         11: 206.132.254.37
         12: 206.178.103.62
         13: reply from 204.71.200.68: bytes=56 (data), time=79 ms
```

```
traceroute: packets sent 13, packets received 12
```

VERS

Displays the software version level, source, software options, and amount of elapsed time that the router has been running.

All software options are listed. If the option has a + prefix, the option was enabled using a key. If the option has a ~ prefix, the option is disabled in this router. For more information, see Software Option Keys, on page 124.

vers

Example:

```
# vers
Efficient 5851 SDSL [ATM] Router
FlowPoint-2000 BOOT/POST V5.3.0 (19-Mar-99 15:25)
Software version v4.0.0 built Mon Apr 17 09:30:26 PDT 2000
Maximum users: unlimited
Options: SDSL, RFC1483, IP ROUTING, IP FILTERING, WEB, +IPSEC, +3DES, L2TP,
ENCRYPT, BRIDGE, IPX
Up for 79 days 19 hours 57 minutes (started 9/8/2000 at 17:11)
```

File System Commands

The file system commands allow you to perform maintenance and recovery on the router. These commands allow you to:

- Format the file system
- List the contents of the file system
- Copy, rename, and delete files

The router file system is DOS-compatible, and the file system commands are similar to the DOS commands of the same name.

COPY

Copies a file from the source to the destination. This command allows you to update the router software level or to write configuration files to a TFTP server

Issue a **sync** command after a **copy** command to commit the changes to FLASH memory.

Caution: No warning message is issued if you copy over an existing file.

copy <*srcfile*> <*dstfile*>

srcfile Filename of the source file to be copied.It can be either the name of a local file or a file accessed remotely via a TFTP server.

A local filename is in the format: *name.ext*.

A remotely accessed filename is specified as: **tftp@***serveraddr:filename.ext*. The TFTP server address is optional. If the TFTP server address is not specified, the address used is either the one from which the router booted or the one permanently configured in the boot system.

To force use of a specific source address when copying a file from a TFTP server, use this format: **tftp@***serveraddr-sourceaddr***:***filename***.***ext*

dstfile Destination filename to which the file is copied.

Examples:

The following command copies the file KERNELNW on TFTP server 128.1.210.66 to the local file KERNEL.F2K.

```
# copy tftp@128.1.210.66:kernelnw kernel.f2k
Copying...
421888 bytes copied
```

The following command uses the source address 192.168.1.2 when copying the file KERNELNW on TFTP server 192.168.100.100 to the local file KERNEL.F2K.

```
copy tftp@192.168.100.100-192.168.1.2:kernelnw kernel.f2k
```

DELETE

Removes a file from the file system.

delete <*filename*>

filename Name of the file to be deleted. The filename is in the format xxxxxxxx.xxx.

Example: `delete kernel.f2k`

Response:

```
kernel.f2k deleted.
```

DIR

Displays the directory of the file system. The size of each file is listed in bytes.

dir

Example: `dir`

EXECUTE

This command loads batch files of configuration commands into the router. This allows for customization and simpler installation of the router. A script file can contain commands, comments (lines introduced by the # or ; characters), and blank lines.

There are two kinds of script files:

- A one-time script that is executed on startup (only once).
- A group of commands that can be executed at any time from the Command Line Interface with the **execute** <*filename*> command.

One-time scripts are useful to execute the complete configuration process from a default (unconfigured) state.

execute <*filename*>

filename Name of the file to be executed.

Example: `execute script1`

FORMAT DISK

Erases and reformats the router file system. This command should *only* be used when the file system is unusable. If the router does not execute the POST test and software boot successfully, and the result of the **dir** command indicates the file system is corrupted, you may wish to reformat the disk, reboot the router, and recopy the router software.

format disk

Example: `format disk`

Response:

```
NEWFS: erasing disk...
NEWFS: fs is 381k and will have 762 sectors
NEWFS: 128 directory slots in 8 sectors
NEWFS: 747 fat entries in 3 sectors
NEWFS: writing boot block...done.
NEWFS: writing fat tables...done.
NEWFS: writing directory...done.
Filesystem formatted!
```

MSFS

Checks the structure of the file system. This command performs a function similar to the DOS **chkdsk** command. The router analyzes the File Allocation Table (FAT) and produces a file system status report.

Warning: When you specify **fix**, make sure that no other operation is being performed on the configuration files at the same time by Configuration Manager or by another user.

msfs [fix]

fix If fix is specified, errors are corrected in the FAT. This option should *only* be used when an **msfs** command results in a recommendation to apply the **fix** option.

Example: `msfs`

Response:

```
Filesystem 0, size=825k:
Checking filesystem...
Checking file entries...
       SYSTEM     CNF ... 2304      bytes .. ok.
       ATM25      DAT ... 20        bytes .. ok.
       DHCP       DAT ... 1536      bytes .. ok.
       KERNEL     F2K ... 257014    bytes .. ok.
       IDL_7      AIC ... 14828     bytes .. ok.
       ASIC       AIC ... 14828     bytes .. ok.
       FILTER     DAT ... 1284      bytes .. ok.
       1097 fat(s) used, 0 fat(s) unused, 0 fat(s) unref, 534 fat(s) free
       561664 bytes used by files, 9728 bytes by tables, 273408 bytes free
```

RENAME

Renames a file in the file system.

rename <*oldName*> <*newName*>

oldName Existing name of the file. The filename is in the format xxxxxxxx.xxx.

newName New name of the file. The filename is in the format xxxxxxxx.xxx.

Example: `rename ether.dat oldeth.dat`

Response:

```
'ether.dat' renamed to 'oldeth.dat'
```

SYNC

Commits the changes made to the file system to FLASH memory.

sync

Example: `sync`

Response:

```
Syncing  file systems...done.
```

Warning: Syncing is not complete until you see the message "done".

SYSTEM Commands

All commands in this section begin with the word **system**. The commands set basic router configuration information, such as the following:

- name of the router
- optional system message
- authentication password
- security authentication protocol
- management security
- system administration password
- IP address translation
- NAT configuration
- host mapping
- WAN-to-WAN forwarding
- filters
- Dial Backup configuration
- SNTP parameters

SYSTEM ?

Lists the supported keywords. To see the syntax for a command, enter the command followed by a **?**.

system ?

Example:

```
# sys ?
System commands:
?                   msg                 name
passwd              authen              community
list                admin               history
log                 addHostMapping      delHostMapping
addServer           delServer           bootpServer
supportTrace        telnetport          snmpport
httpport            syslogport          addTelnetFilter
delTelnetFilter     addSNMPFilter       delSNMPFilter
addHTTPFilter       delHTTPFilter       addSyslogFilter
delSyslogFilter     wan2wanforwarding   OneWANdialup
blockNetBIOSDefault addUDPrelay         delUDPrelay
securityTimer       addIPRoutingTable   delIPRoutingTable
moveIPRoutingTable
```

SYSTEM ADDBOOTPSERVER

Adds an address to the BootP server list. (The BootP server list is also the DHCP relay list.)

While the BootP server list has at least one address, the router disables its own DHCP server and, instead, forwards all DHCP/BootP requests to all servers in the list. It forwards every reply received from any of the servers in the list to the appropriate LAN. To read about BootP service, see page 167.

Addresses can also be added to the list using the **dhcp addrelay** command (page 352). To remove an address from the list, use the **dhcp delrelay** command (page 355).

To see the current BootP server address, enter the command **dhcp addrelay** or **system addBootPServer** with no parameters. To remove a BootPserver address, use the command **dhcp delrelay** or **system delBootPServer** (page 242).

system addbootpServer <*ipaddr*>

ipaddr IP address of the server (4 decimals separated by periods).

Example:

```
# system addbootpServer 128.1.210.64
# system addbootpServer
BOOTP/DHCP Server address: 128.1.210.64
```

SYSTEM ADDHOSTMAPPING

This command is used to remap a range of local-LAN IP addresses to a range of public IP addresses on a *system-wide basis*. These local addresses are mapped one-to-one to the public addresses.

Note: The range of public IP addresses is defined by <*first public addr*> only. The rest of the range is computed automatically (from <*first public addr*> to <*first public addr*> + number of addresses remapped - 1) inclusive.

system addHostMapping <*first private addr*> <*second private addr*> <*first public addr*>

first private addr First IP address in the range of IP addresses to be remapped, in the format of 4 decimals separated by periods.

second private addr Last address in the range of IP addresses to be remapped, in the format of 4 decimals separated by periods.

first public addr Defines the range of public IP addresses, in the format of 4 decimals separated by periods.

The rest of the range is computed automatically.

Example: `system addHostMapping 192.168.207.40 192.168.207.49 10.1.1.7`

SYSTEM ADDHTTPFILTER

This command can block all devices except those within the defined IP address range from using the HTTP protocol (for example, to browse the Web). This command can block devices on the WAN from accessing the Web browser. This validation feature is *off* by default.

Note 1: This command does *not* require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password). To delete addresses from the HTTP filter, use the command **system delHTTPfilter** (page 243).

For more information, see Controlling Remote Management, on page 107.

system addHTTPFilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr First IP address of the range.

last ip addr Last IP address of the range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example: `system addHTTPFilter 192.168.1.5 192.168.1.12`

SYSTEM ADDIPROUTINGTABLE

Defines a new virtual routing table. Once defined, you can add routes to the table using the commands **eth ip bindRoute** (page 266) and **remote bindIPVirtualRoute** (page 293).

The command specifies the name of the new routing table and the range of IP addresses that reference the table for their routing. When the router receives a packet, the source address of the packet determines which routing table is used. For example, if the range of addresses for the virtual routing table ROSA includes address 192.168.25.25, then every packet with the source address 192.168.25.25 is routed using virtual routing table ROSA.

If the source address of a packet is not within the address ranges for any virtual routing table, the default routing table is referenced to route the packet.

For more information, see Virtual Routing Tables, on page 80.

If an IP routing table has been defined, you can see its range of addresses using the command **system list**.

system addIPRoutingTable <*first ip addr*> [<*last ip addr*>] <*tablename*>

first ip addr First IP address of the range (4 decimals separated by periods).

last ip addr Last IP address of the range (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address. The specified address range may not overlap the address range defined for any other virtual routing table.

tablename Name of the virtual routing table to which the addresses are assigned (character string).

Example:

The following command defines a virtual routing table named ROSA (if it does not already exist) and assigns it the IP address range 192.168.1.5 through 192.168.1.12.

```
system addIPRoutingTable 192.168.1.5 192.168.1.12 ROSA
```

After routing table ROSA has been defined, the following line appears in the output for the command **system list**:

```
192.168.1.5 through 192.168.1.12 uses IP Routing Table <ROSA>
```

SYSTEM ADDSERVER

This Network Address Translation (NAT) command is used to configure a local IP address as the selected server on the LAN (FTP, SMTP, etc.) for the global configuration. To learn more, see Network Address Translation (NAT), on page 95.

Multiple **system addserver**, **remote addserver** (page 293), and **eth ip addserver** (page 265) commands can designate different servers for different protocols, ports, and interfaces. When a request is received, the router searches the server list for the appropriate server. The order of search for a server is discussed in Server Request Hierarchy, on page 98.

To delete a server designation, use the command **system delserver** (page 244).

system addServer <*action*> <*protocol*> <*first port*> [<*last port*> [<*first private port*>]]

action One of the following command actions:

ipaddr	Selects the host with this IP address as server (4 decimals separated by periods).
discard	Discards the incoming server request.
me	Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.

protocolid	Numeric protocol ID.
tcp	TCP only.
udp	UDP only.
all	All protocols.

first port First or only port as seen by the remote end. Port used by the selected server

portid	Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
dns	DNS port (Domain Name Server).
ftp	FTP port (File Transfer Protocol).
h323	H.323 port.
http	HTTP port (Hypertext Transfer Protocol used on the Internet).
login	rlogin port (port 513).
rsh	Remote Shell port.
smtp	SMTP port (Simple Mail Transfer Protocol).
snmp	SNMP port (Simple Network Management Protocol).
t120	T.120 port.
telnet	Telnet port.
tftp	TFTP port (Trivial File Transfer Protocol).
all	All ports.

last port Optional last port in the range of ports as seen by the remote end for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the remote end.

Example:

```
system addServer 192.168.1.5 tcp smtp
```

SYSTEM ADDSNMPFILTER

This command is used to validate SNMP clients by defining a range of IP addresses that are allowed to access the router via SNMP. This validation feature is *off* by default.

Note 1: This command does *not* require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password). To delete addresses from the SNMP filter, use the command **system delSNMPfilter** (page 244).

For more information, see Controlling Remote Management, on page 107.

system addSNMPFilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr First IP address of the client range.

last ip addr Last IP address of the client range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example: `system addSNMPFilter 192.168.1.5 192.168.1.12`

SYSTEM ADDSYSLOGFILTER

Limits the Syslog server addresses that may be returned by DHCP. By default, this validation feature is *off*.

The Syslog filter can comprise one or more ranges of IP addresses that DHCP may return for Syslog servers. To delete addresses from the Syslog filter, use the command **system delsyslogfilter** (page 245).

This command does not affect the Syslog server addresses that you specify explicitly. For more information on the router as Syslog client, see page 168.

Note: This command does not require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password).

system addSyslogFilter <*firstipaddr*> [<*last ip addr*>] | LAN

first ip addr First IP address of the valid server range.

last ip addr Last IP address of the valid server range. May be omitted if the range contains only one IP address.

LAN Limits the valid Syslog servers to those on the local Ethernet LAN.

Example:

```
system addSyslogFilter 192.168.1.5 192.168.1.12
```

SYSTEM ADDSYSLOGSERVER

Adds an address to the list of Syslog servers. The router sends system event messages to all Syslog servers in the list, unless the Syslog port has been disabled. For more information about the router as Syslog client, see page 168.

To see the server addresses, use the command **system list**. To remove a Syslog server address from the list, use the command **system delSyslogServer** (page 245).

Note: The new server address becomes effective after you **save** and **reboot**.

system addSyslogServer <*ipaddr*>

ipaddr IP address to be added to the Syslog server address list (4 decimals separated by periods).

Example:

```
system addSyslogServer 192.168.1.5
```

SYSTEM ADDTELNETFILTER

This command is used to validate Telnet clients by defining a range of IP addresses that are allowed to access the router via Telnet. This validation feature is *off* by default. For more information, see Controlling Remote Management, on page 107.

Note 1: This command does *not* require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (log in with password). To delete addresses from the Telnet filter, use the command **system delTelnetfilter** (page 245).

system addTelnetFilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr First IP address of the client range.

last ip addr Last IP address of the client range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example:

```
system addTelnetFilter 192.168.1.5 192.168.1.12
```

SYSTEM ADDUDPRELAY

This command is used to create a UDP port range for packet forwarding. You can specify a port range from 0 to 65535; however, 137 to 139 are reserved for NetBIOS ports. Overlap of UDP ports is not allowed.

system addUDPrelay <*ipaddr*> <*first port*>|all [<*last port*>]

ipaddr IP address of the server to which the UDP packet will be forwarded.

first port First port in the UDP port range to be created.

all Incorporates all the available UDP ports in the new range.

last port Last port in the UDP port range to be created.

Example: `system addUDPrelay 192.168.1.5 all`

SYSTEM ADMIN

Sets the administration password that is used to control write access to the target router configuration.

system admin <*password*>

password Write-enable login password**.**

Example: `system admin adx1lp`

SYSTEM AUTHEN

Forces the target router authentication protocol that is used for security negotiation with the remote routers when the local side authentication is set. You should not need to issue this command as the best security possible is provided with the **none** default.

To see the current authentication override (**none**, **pap,** or **chap**), enter the command **system authen** with no parameters. To read about PAP/CHAP authentication, see page 25.

system authen none | pap | chap

none The authentication protocol is negotiated, with the *minimum* best security level as defined for each remote router in the database.

pap Negotiation begins with PAP (instead of CHAP) for those entries that have PAP in the remote database and only when the call is initiated locally.

chap Overrides all the remote database entries with CHAP, that is, only CHAP is performed.

Example:

```
# system authen chap
```

```
# system authen
Authentication needed.................CHAP
```

SYSTEM BACKUP ADD

Adds an IP address to the list of addresses to be pinged for the Dial Backup option. The command can specify an explicit address, or it can request that the router determine the gateway or DNS address and add that address to the list.

For more information about Dial Backup, see page 109.

system backup add <*ipaddr*> | GW | DNS [<*group*>]

ipaddr IP address to be added to the list (four decimals separated by periods).

GW Gateway address. The router determines the actual gateway address.

DNS Domain Name Server address. The router determines the actual DNS address.

group Optional number of a group to which the address is assigned (integer, 0 through 65535). The default is group 0.

Examples:

The following command adds the address 192.168.1.5 to group 0 of the addresses to be pinged.

```
system backup add 192.168.1.5
```

The following command adds the gateway address to group 1 of the addresses to be pinged.

```
system backup add  GW  1
```

SYSTEM BACKUP DELETE

Deletes an IP address from the list of addresses to be pinged for the Dial Backup option. The command can:

- Specify an explicit address to be deleted.
- Request that the router delete the gateway or DNS address from the list.
- Delete all addresses in a group.
- Clear all addresses from the list.

To see the addresses in the current list, use the command **system list**. For more information about Dial Backup, see page 109.

system backup delete <*ipaddr*> | GW | DNS | all [<*group*> | all]

ipaddr IP address to be deleted from the list (four decimals separated by periods).

GW Gateway address. The router determines the actual gateway address and deletes it.

DNS Domain Name Server address. The router determines the actual DNS address and deletes it.

all Requests deletion of all addresses in the group.

group Optional number of a group from which the specified address or all addresses are deleted (integer, 0 through 65535). The default is group 0.

all Specifies all groups, including group 0.

Examples:

The following command deletes the address 192.168.1.5 from group 0.

```
system backup delete 192.168.1.5
```

The following command deletes the gateway address from group 1.

```
system backup delete  GW  1
```

The following command deletes all addresses from group 2.

```
system backup delete  all  2
```

The following command clears all addresses from the list.

```
system backup delete  all  all
```

SYSTEM BACKUP DISABLE

Disables the Dial Backup option in the router.

Note: Because Dial Backup uses the console port, you cannot access the command line via the console port while Dial Backup is enabled. You must use the Web GUI interface or a Telnet session to disable Dial Backup.

Note: If you do not use the **save** command to save this change, Dial Backup is only temporarily disabled and it is re-enabled at the next reboot. Temporarily disabling Dial Backup stops Dial Backup, but it does not change the use of the console port.

To disable Dial Backup *across reboots* and change the use of the console port, enter the following commands:

```
system backup disable
save
reboot
```

To re-enable the Dial Backup option, use the **system backup enable** command.

For more information about Dial Backup, see page 109.

system backup disable

SYSTEM BACKUP ENABLE

Turns on the enable switch for the Dial Backup option in the router. Use this command to re-enable Dial Backup after it has been disabled, as follows:

- If Dial Backup has been *temporarily* disabled, this command restarts its use.
- If Dial Backup has been disabled *across one or more reboots*, it can be re-enabled by the command sequence:

```
system backup enable
save
reboot
```

Note: Dial Backup cannot be enabled unless the remote containing its dialup parameters is also enabled. (Check this using the command **remote list**).

To see the current setting of the Dial Backup switch, use the **system list** command. To disable Dial Backup, use the **system backup disable** command.

For more information about Dial Backup, see page 109.

system backup enable

SYSTEM BACKUP PINGINTERVAL

Changes the ping interval for a group, that is, the number of seconds between pings during a test of the addresses in the group.

Note: If you change the ping interval to **0**, you disable the group of addresses.

To see the current ping intervals, use the **system list** command. For more information about the ping interval and Dial Backup, see Ping Interval, Number of Samples, and Success Rate, on page 114.

system backup pinginterval <*seconds*> [<*group*>]

seconds Number of seconds in the ping interval for the group (integer). The default is 5 seconds.

group Optional number of a group (integer, 0 thru 65535). The default is group 0.

Examples:

The following command changes the ping interval to 10 seconds for group 0.

```
system backup pinginterval 10
```

The following command disables the pinging of addresses in group 1.

```
system backup pinginterval  0  1
```

SYSTEM BACKUP PINGSAMPLES

Changes the number of ping samples for a group, that is, the number of pings performed for each address in the group.

Note: If you change the ping samples value to **0**, you disable pinging for that group of addresses.

To see the current ping sample values, use the **system list** command. For more information about ping samples and Dial Backup, see Ping Interval, Number of Samples, and Success Rate, on page 114.

system backup pingsamples <*samples*> [<*group*>]

samples Number of times the addresses in the group are pinged (integer). The default is 6.

group Optional number of a group (integer, 0 through 65535). The default is group 0.

Examples:

The following command changes the number of ping samples to 10 for addresses in group 0.

```
system backup pingsamples  10
```

The following command disables the pinging of addresses in group 1.

```
system backup pingsamples  0  1
```

SYSTEM BACKUP RETRY

Changes the Dial Backup retry period. The retry period determines how often the router attempts to restore the DSL link. For more information about the Dial Backup retry period, see DSL Restoration Retry Period, on page 112.

The default retry period is thirty minutes. The minimum retry period is two minutes. To see the current retry value, use the **system list** command

Note: When the Dial Backup **retry** timer expires, the modem is disconnected even if there is traffic on the modem.

system backup retry <*minutes*>

minutes Number of minutes in the retry period (integer). The default is 30; the minimum is 2.

Example:

The following command changes the retry period to 60 minutes.

```
system backup retry 60
```

The following command changes the retry period to 2 minutes because the minimum is 2 minutes.

```
system backup retry 1
```

SYSTEM BACKUP STABILITY

Changes the Dial Backup stability period. The stability period guards against frequent switching back and forth between the DSL link and the backup port. For more information about the Dial Backup stability period, see Stability Period, on page 112.

The default stability period is three minutes. The minimum stability period is one minute.

To see the current stability value, use the **system list** command.

system backup stability <*minutes*>

minutes Number of minutes in the stability period (integer). The default is 3; the minimum is 1.

Example:

The following command changes the stability period to 5 minutes.

```
system backup stability 5
```

SYSTEM BACKUP SUCCESSRATE

Changes the minimum success rate required for a group of pinged addresses. If the success rate is less than the minimum, the DSL link is assumed to have failed and a switchover to the backup is performed.

Note: If you change the success rate to **0**, you disable pinging for that group of addresses.

Note: A minimum success rate of 100% is not recommended; this would require a reply from every ping sent.

To see the current success rate values, use the **system list** command. For more information about success rates and Dial Backup, see Ping Interval, Number of Samples, and Success Rate, on page 114.

system backup successrate <*percentage*> [<*group*>]

percentage Minimum success rate required during a ping test of the addresses in the group (integer, 0 thru 99). The default is 50.

group Optional number of a group (integer, 0 thru 65535). The default is group 0.

Examples:

The following command changes the success rate to 75% for addresses in group 0.

```
system backup successrate 75
```

The following command disables the pinging of addresses in group 1.

```
system backup successrate  0  1
```

SYSTEM BLOCKNETBIOSDEFAULT

The router can block all NetBIOS and NetBUI requests from being sent over the WAN. This command sets the default value used when a remote router entry is defined.

The command **remote blockNetBIOS** (page 294) can change the NetBIOS setting for a specific remote router. To see the current NetBIOS default, use the command **system list**.

system blockNetBIOSDefault yes | no

yes Sets the default to block all NetBIOS and NetBUI requests.

no Sets the default to *not* block NetBIOS and NetBUI requests.

Example:

```
system blockNetBIOSdefault yes
```

SYSTEM COMMUNITY

Enhances SNMP security by allowing the user to change the SNMP community name from its default value of "public" to a different value. Refer to Controlling Remote Management, on page 107.

Note: The command **system community** (with no value) will display the current community name.

system community [<*SNMP community name*>]

SNMP community name String of up to 40 characters.

Example 1: `system community fred`

Example 2: `system community`

SYSTEM DEFAULTMODEM

Lists the default modem settings. The modem settings are for the backup V.90 modem connected to the console port.

To change the modem settings, use the **system modem** command (page 249). For more information on the Dial Backup option, see page 164.

system defaultmodem

SYSTEM DELBOOTPSERVER

Removes an address from the BootP server list. (The BootP server list is also the DHCP relay list.)

To remove all addresses from the list, use **system delbootpserver all**.

Addresses can also be removed from the list using the **dhcp delrelay** command (page 355). To add an address to the list, use the **dhcp addrelay** command (page 352).

system delbootpServer <*ipaddr*> | all

ipaddr IP address of the server (4 decimals separated by periods).

all Removes all addresses from the BootP server list.

Examples:

```
system delbootpServer 128.1.210.64
system delbootpServer all
```

SYSTEM DELHOSTMAPPING

Undoes an IP address/host translation (remapping) range that was previously established with the command **remote addHostMapping** on a *per-systemwide basis*.

system delHostMapping <*first private addr*> <*second private addr*> <*first public addr*>

first private addr First IP address in the range of IP address, in the format of 4 decimals separated by periods.

second private addr Last address in the range of IP address, in the format of 4 decimals separated by periods.

first public addr Defines the range of public IP addresses, in the format of 4 decimals separated by periods.

The rest of the range is computed automatically.

Example: `system delHostMapping 192.168.207.40 192.168.207.49 10.1.1.7`

SYSTEM DELHTTPFILTER

Deletes an address filter created by the **system addHTTPFilter** command. To see the address range of the filter, use the command **system list**.

system delHTTPFilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr First IP address of the range.

last ip addr Last IP address of the range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example:

```
system delHTTPFilter 192.168.1.5 192.168.1.12
```

SYSTEM DELIPROUTINGTABLE

Deletes a range of addresses that reference a virtual routing table or deletes the entire virtual routing table.

To list the virtual routing tables, use the **iproutes** command (page 215).

For more information, see Virtual Routing Tables, on page 80.

system delIPRoutingTable ALL | <*first ip addr*> [<*last ip addr*>] <*tablename*>

ALL Deletes the virtual routing table. Both the table definition and all routes in the table are deleted.

first ip addr First IP address of the range to be deleted (4 decimals separated by periods).

last ip addr Last IP address of the range to be deleted (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address.

tablename Name of the virtual routing table (character string).

Examples:

Deletes two IP addresses from the address range that references routing table ROSA:

```
system delIPRoutingTable 192.168.1.5 192.168.1.6 ROSA
```

Deletes the virtual routing table ROSA:

```
system delIPRoutingTable all ROSA
```

SYSTEM DELSERVER

Deletes an entry created by the **system addServer** command (page 233).

system delServer *<action> <protocol> <first port>* [*<last port>* [*<first private port>*]]

action One of the following command actions:
- *ipaddr* Selects the host with this IP address as server (4 decimals separated by periods).
- **discard** Discards the incoming server request.
- **me** Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.
- *protocolid* Numeric protocol ID.
- **tcp** TCP only.
- **udp** UDP only.
- **all** All protocols.

first port First or only port as seen by the remote end. Port used by the selected server
- *portid* Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
- **dns** DNS port (Domain Name Server).
- **ftp** FTP port (File Transfer Protocol).
- **h323** H.323 port.
- **http** HTTP port (Hypertext Transfer Protocol used on the Internet).
- **login** rlogin port (port 513).
- **rsh** Remote Shell port.
- **smtp** SMTP port (Simple Mail Transfer Protocol).
- **snmp** SNMP port (Simple Network Management Protocol).
- **t120** T.120 port.
- **telnet** Telnet port.
- **tftp** TFTP port (Trivial File Transfer Protocol).
- **all** All ports.

last port Optional last port in the range of ports as seen by the remote end for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the remote end.

Example:

```
system delServer 192.168.1.5 tcp smtp
```

SYSTEM DELSNMPFILTER

Deletes the client range previously defined by the command **system addsnmpfilter**.

Note 1: This command does *not* require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password).

system delSNMPFilter *<first ip addr>* [*<last ip addr>*] | LAN

first ip addr First IP address of the client range.

last ip addr Last IP address of the client range*;* may be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example: `system delsnmpfilter 192.168.1.5 192.168.1.12`

SYSTEM DELSYSLOGFILTER

Deletes the Syslog address filter.To see the address range of the filter, use the command **system list**. To define a new Syslog address filter, use the command **system addSyslogFilter** (page 234).

Note: This command does not require a reboot; it takes effect immediately.

system delSyslogFilter <*firstipaddr*> <*lastipaddr*> | LAN

first ip addr First IP address of the range.

last ip addr Last IP address of the range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example:

```
system delSyslogFilter 192.168.1.5 192.168.1.12
```

SYSTEM DELSYSLOGSERVER

Removes an address from the list of Syslog servers. To see the server addresses, use the command **system list**. To specify a new Syslog server address, use the command **system addSyslogServer** (page 235).

Note: This command does not require a reboot; it takes effect immediately.

system delSyslogServer <*ipaddr*>

ipaddr IP address to be removed from the Syslog server address list.

Example:

```
system delSyslogServer 192.168.1.5
```

SYSTEM DELTELNETFILTER

Deletes the client range previously defined by the command **system addTelnetFilter**.

Note 1: This command does *not* require a reboot and is effective immediately.

Note 2: To list the range of allowed clients, use the command system list when logged in with read and write permission (be sure to log in with password).

system delTelnetFilter <*first ip addr*> [<*last ip addr*>] | LAN

first ip addr First IP address in the client range.

last ip addr Last IP address in the client range; may be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

Example: `system deltelnetfilter 192.168.1.5 192.168.1.12`

SYSTEM DELUDPRELAY

Deletes the port range that was previously enabled by the command **system addUDPrelay.**

system delUDPrelay <*ipaddr*> <*first port*>| *all [<last port>]*

ipaddr IP address of the server.

first port First port in the UDP port range to be deleted.

all Deletes all existing UDP ports.

last port Last port in the UDP port range to be deleted.

Example: `system delUDPrelay 192.168.1.5 all`

SYSTEM HISTORY

Displays the router's most recent console log.

system history

Example:

```
# system history
Begin System History.

POST summary: successful

Initializing the system RAM ..... done

Hardware "IDSL" successfully initialized -- ID: 3000
Today is Tuesday May 15, 2001; the time is 10:40:30
My MAC address is: 00:20:6F:0B:67:A1
Reason for this reset: power up

Trying to boot from flash memory

loading ..............................................................don
e.
Verifying CRC (77D79D92).....................................done.
```

```
Efficient Networks, Inc. SS5871 (P/N 120-5871-001), Rev 34-06 (S/N 747425)
Now 2769k free before buffers
Interfaces detected
   LAN: Ethernet (10BASET HUB)   WAN: IDSL


SpeedStream 5871 IDSL Router (120-5871-001/2) v5.0.0
Copyright (c) 1999-2000 Efficient Networks, Inc.
All Rights Reserved

INIT: buffer pool is 1371632 bytes
ETHERNET/0 interface started, MAC=00:20:6F:0B:67:A1
05/15/2001-10:40:38:ETH: Obtaining an IP address for ETHERNET/0:3 with DHCP

SpeedStream 5871 IDSL Router (120-5871-001/2) v5.0.0 Ready
Login:
Login: *****
Logged in successfully!
# system history

End System History.
#
```

SYSTEM HTTPPORT

This command manages HTTP port access. It can:

- Disable HTTP for this router (sets the HTTP port to 0).
- Request the default HTTP port (80). This re-enables HTTP after it is disabled.
- Redefine the HTTP port.

Note: This command requires a **save** and **reboot** to take effect.

To see the current setting, use the command **system list**. For more information, see Controlling Remote Management, on page 107.

system httpPort default | disabled | <*port*>

default	Restores the port value to the default value **80** and re-enables the port.
disabled	Disables the HTTP port.
port	Defines a new HTTP port number. Use this option to restrict remote access.

Examples:

```
system httpport default

system httpport disabled

system httpport 3333
```

SYSTEM LIST

Lists the system settings for the target router.

To change the listed settings, use these commands:

`GENERAL INFORMATION FOR <>`	**system name** (page 250)
`Authentication override`	**system authen** (page 236)
`WAN to WAN Forwarding`	**system wan2wanforwarding** (page 261)
`Block NetBIOS Default`	**system blockNetBIOSdefault** (page 241)
`BOOTP/DHCP Server address`	**dhcp addrelay** (page 352)
`Telnet Port`	**system telnetPort** (page 260)
`Telnet Clients`	**system addtelnetfilter** (page 235)
`SNMP Port`	**system SNMPPort** (page 252)
`SNMP Clients`	**system addSNMPfilter** (page 234)
`HTTP Port`	**system httpPort** (page 247)
`HTTP Clients`	**system addHTTPfilter** (page 232)
`Syslog Port`	**system SyslogPort** (page 259)
`Allowed Syslog Servers`	**system addSyslogFilter** (page 234)
`Default Syslog Servers`	**system addSyslogServer** (page 235)
`System message:`	**system msg** (page 250)
`Security timer`	**system securityTimer** (page 252)
`One WAN Dial Up`	**system oneWANdialup** (page 251)
`Backup`	**system backup** (page 251)
`Retry Interval`	**system backup retry** (page 240)
`Stability Interval`	**system backup stability** (page 240)

system list

Example:

```
# system list
GENERAL INFORMATION FOR <SOHO>
  System started on..................... 9/8/2000 at 13:29
  Authentication override.............. none
  WAN to WAN Forwarding................ no
  Block NetBIOS Default................ no
  BOOTP/DHCP Server address........... none
  Telnet Port.......................... default (23)
  Telnet Clients....................... all
  SNMP Port............................ default (161)
  SNMP Clients......................... all
  HTTP Port............................ default (80)
  HTTP Clients......................... all
  Syslog Port.......................... default (514)
```

```
Allowed Syslog Servers.............. all
Default Syslog Servers.............. none
System message:
Security timer..................... 30 minutes
One WAN Dial Up.................... no
Backup............................. no (no valid remote profile is enabled)
  Retry Interval In Minutes......... 30
  Stability Interval In Minutes..... 3
```

SYSTEM LOG

Allows logging of the router's activity in a Telnet session.

system log start | stop | status

start — Used to monitor router activity at all times.

Example: `system log start`

stop — Used to discontinue the logging utility at the console.

Example: `system log stop`

status — Used to find out if other users (yourself included) are using this utility.

Example: `system log status`

SYSTEM MODEM

Changes the selected modem setting. The modem settings are for the backup asynchronous modem connected to the console port.

For more information on the Dial Backup option, see page 164.

system modem reset | escape | init | offhook | dial | answer | hangup <*string*>

dial — The two possible strings for the **dial** setting are **ATDT** for tone dialing or **ATDP** for pulse dialing. The default is tone dialing.

<*string*> — New setting for the option selected by the first parameter

Examples:

The following command changes the string for the init setting:

```
system modem init  ATS0=0Q0V1&C2&D3&K1X4&H1&I0S12=20
```

The following command selects pulse dialing:

```
system modem dial ATDP
```

SYSTEM MOVEIPROUTINGTABLE

Moves a range of IP addresses to another virtual routing table. The command first looks at the address ranges defined for other virtual routing tables, searching for the addresses to be moved. If it finds addresses to be moved, it deletes them from the address ranges for the other virtual routing tables. The command then adds the specified address range to the virtual routing table named on the command.

To list the routes in the virtual routing tables, use the **iproutes** command (page 215) or the **remote listiproutes** command (page 305).

For more information, see Virtual Routing Tables, on page 80.

system moveIPRoutingTable <*first ip addr*> [<*last ip addr*>] <*tablename*>

first ip addr First IP address of the range to be moved (4 decimals separated by periods).

last ip addr Last IP address of the range to be moved (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address.

tablename Name of the virtual routing table to be assigned the address range (character string). The virtual routing table may be new or it may already exist.

Example:

Suppose you want all packets with source addresses in the range 192.168.254.11 through 192.168.254.20 to be routed using virtual routing table MIGUEL. Addresses in that range may already be assigned to other virtual routing tables. Therefore, to delete the addresses from any other virtual routing tables and assign the address range to MIGUEL, you enter this command:

```
system moveIPRoutingTable 192.168.254.11 192.168.254.20 MIGUEL
```

SYSTEM MSG

Sets or changes the message saved in the local router you are configuring. To see the current message, enter **system msg** with no parameters or use the command **system list**.

system msg <*message*>

message Message (up to 255 characters). Space characters are not allowed; use underscore characters instead. If you do not enter a message, the current message is displayed.

Example:

```
# system msg Configured _on_10/21/98
# system msg
  System message: Configured _on_10/21/98
```

SYSTEM NAME

Sets or changes the name of the local router being configured. To see the current router name, enter **system name** with no parameters.

You must assign a name to the local router. This name is sent to a remote router during PAP/CHAP authentication.

system name [<*name*>]

name Name of the target router (character string).

Note: The system name is case-sensitive and may be no more than 50 characters.

Space characters are not allowed within the name; you may use underscore characters instead. (The system name is a "word" when exchanged with PAP/CHAP.)

If you do not enter a name, the current name of the router is displayed. If you type anything after **system name**, the characters will be taken as the new name.

Example:

```
# system name Router1
# system name
System name: <Router1>
```

SYSTEM ONEWANDIALUP

This command can force the router to have no more than *one* remote connection active at a time. (Multiple links to the same remote are allowed.) To see the current setting, use the command **system list** and check the **One WAN Dial Up** line.

This command is useful when security concerns dictate that the router have only one connection active at a time. For example, if set to *on*, the router cannot connect to both the Internet and another location (such as your company) at the same time.

A connection is only generated when data is forwarded to the remote router (dial-on-demand); Permanent links cannot be automatically generated.

The command allows multiple connections to the SAME location and supports the PPP Multi-Link protocol. To do so, at system startup time, the router examines each remote entry. If if finds only one remote enabled, it leaves the remote enabled. If it finds more than one remote enabled, it disables every entry that does not have a protocol of PPP or PPPLLC. It sets the minimum number of active links (**remote minLink**) to 0 (zero) on the enabled entries; if the command did not perform this function, connections to multiple destinations would not be possible (since the link to the destination with **minLink=**non-zero would be active).

This **system oneWANdialup** command complements the **system wan2wanforwarding** command (page 261). That command allows multiple connections to different locations to be active at the same time but stops traffic from passing from one WAN connection to another.

system oneWANdialup on | off

on Enables only one active connection at a time to a remote entry.

off Disables **system oneWANdialup**, allowing WAN connections to multiple locations**.**

Example:

```
system oneWANdialup on
```

SYSTEM PASSWD

Sets the system authentication password for the target router that is used when the router connects to other routers or is challenged by them. This password is a default password used for all remote sites unless a unique password is explicitly defined for connecting to a remote router with the **remote setOurPasswd** command.

system passwd <*password*>

password Authentication password of the target router.

Note: The password is case-sensitive and should be no more than 40 characters.

Example: `system passwd chwgn1`

SYSTEM SECURITYTIMER

This command allows the user to change the 10-minute default security timer to another value. The router automatically logs out a Telnet or console user out of privileged mode when no typing has occurred for the length of time set for the security timer.

Note: To disable the security timer, set its value to 0.

To see the current security timer value, use the command **system list**.

system securityTimer <*minutes*>

minutes Timer length in minutes. To disable the automatic logout, set the value to 0.

Example: `system securityTimer 15`

SYSTEM SNMPPORT

This command manages SNMP port access. It can:

- Disable SNMP for this router (sets the SNMP port to 0).
- Request the default SNMP port (161). This re-enables SNMP after it is disabled.
- Redefine the SNMP port.

Note: This command requires a **save** and **reboot** to take effect.

To see the current setting, use the command **system list**. For more information, see Controlling Remote Management, on page 107.

system snmpport default | disabled | <*port*>

default Restores the port value to the default value **161** and re-enables the port.

disabled Disables remote SNMP management.

port Defines a new SNMP port number. Use this option to restrict remote access.

Examples:

```
system snmpport default

system snmpport disabled

system snmpport 3333
```

SYSTEM SUPPORTTRACE

Lets you capture to a file all the configuration data that Technical Support may need to investigate configuration problems. This exhaustive list command incorporates the following commands:

- **system history**
- **vers**
- **mem**
- **system list**
- **eth list**
- **dhcp list (if DHCP is enabled)**
- **remote list**
- **ifs**
- **bi (if bridging is enabled)**
- **ipifs**
- **iproutes**
- **ipxroutes**

system supporttrace

Example:

```
# system supporttrace

=== HISTORY ===

End System History.

=== VERSION ===
Efficient 7851 SDSL [CM/FR] (120-7851-034) Router
Efficient-5000 BOOT/POST V7.0.101 (19-Apr-01 16:57)
Software version v5.X.Y(irislin).0 built Mon May 7 17:42:01 PDT 2001
Maximum users: unlimited
Options: FRAME RELAY, ASYNC, SDSL, VOICE-TOLLBRIDGE, RFC1483, IP ROUTING,
         IP FILTERING, WEB, HW-DES, IPSEC, 3DES, L2TP, ENCRYPT, BRIDGE, IPX,

         CMMGMT, DIAL-BACKUP, VRRP
```

```
Up for 0 days 20 hours 53 minutes (started 5/17/2001 at 17:49)
=== MEMORY ===
Amount of RAM installed..  4096 Kbytes
Small buffers used.......  25 (3% of 656 used)
Large buffers used....... 161 (23% of 700 used)
Buffer descriptors used.. 186 (10% of 1695 used)
Number of waiters s/l....   0/0

Table memory allocation statistics:
Sizes      8    16    32    64   128   256   512  1024
Used       7   132    28    90     2    13     7     5
Free       1     1     2     2     1     2     1     1

Sizes   2048  4096  8192
Used      19     9     1
Free       0     1     0

Total in use: 105080, total free: 968952 (6488 + 962464)

=== PROCESSES ===
TID:          NAME               FL P BOTTOM CURRENT SIZE
  1:IDLE                         02 7 2f6974  2f7880 4080
 24:SENDSIG                      04 3 30ec84  30f368 2032
  3:MSFS_SYNC                    03 6 2f8a04  2f9100 2032
  4:SYSTEM LOGGER                03 5 2fc874  2fcf70 2032
  5:LL_PPP                       03 5 2fb844  2fc738 4080
  6:NL_IP                        03 5 2fddf4  2fe4f0 2032
  7:TL_IP_UDP                    03 3 2fe674  2fed78 2032
  8:TL_IP_TCP                    03 3 2feed4  2ff5d8 2032
  9:TELNETD                      03 5 2ff734  2ffe18 2032
 10:IKE                          03 4 301504  301be8 2000
 11:BOOTP                        03 5 303fd4  3046c0 2032
 12:DUM                          03 5 302964  303850 4080
 13:SDSL                         03 5 304d34  3053d8 2032
 14:CALLCTRL                     03 3 306624  306d18 2032
 15:DSP                          03 3 306e34  307520 2032
 16:SNMPD                       03 5 3055a4  3064a8 4080
 17:CAS                          03 3 3076d4  307dc0 2032
 18:HAPI                         04 2 307ff4  308ed8 4096
 19:HTTPD                        03 5 3090a4  309f58 4080
 20:DNS                          03 5 30a204  30b0b0 4000
 21:SNTP                         03 4 30e454  30eb38 2000
 22:CMD                          01 6 30cf54  30db58 4080
 25:IP_RIP                       03 4 310a94  311190 2032


=== FILE SYSTEM ===
Filesystem 0, size=1714k :
Checking filesystem...
Checking file entries...
  KERNEL   IRI ... 684629 bytes .. ok.
  ASIC     AIC ...  50847 bytes .. ok.
  KEYFILE  DAT ...    768 bytes .. ok.
  SYSTEM   CNF ...   2304 bytes .. ok.
  FRAME    DAT ...      0 bytes .. ok.
  ATOM     DAT ...      0 bytes .. ok.
  DHCP     DAT ...   1280 bytes .. ok.
  SDSL     DAT ...     28 bytes .. ok.
```

```
  41DB833E GAN ...    192 bytes .. ok.
  2BC5A0B4 GAN ...    192 bytes .. ok.
  2BC5A0B4 DHV ...    960 bytes .. ok.
  DSP      DAT ...     28 bytes .. ok.
  USER     BAT ...    462 bytes .. ok.
  41DB833E DHV ...    960 bytes .. ok.
  EF2E6B8F GAN ...    192 bytes .. ok.
  35B2A0B5 GAN ...    192 bytes .. ok.
  35B2A0B5 DHV ...    960 bytes .. ok.
  EF2E6B8F DHV ...    960 bytes .. ok.
  2D4E5524 GAN ...    192 bytes .. ok.
  2D4E5524 DHV ...    960 bytes .. ok.
  FILTER   DAT ...   1284 bytes .. ok.
  KERNEL   F2K ... 684629 bytes .. ok.
  2807 fat(s) used, 590 fat(s) free
  0 fat(s) unused, 0 fat(s) unreferenced, 2 fat(s) reserved
  1437184 bytes used by files, 14848 bytes by tables, 302080 bytes free


=== SYSTEM ===
GENERAL INFORMATION FOR <>
  System started on..................... 5/17/2001 at 17:49
  Authentication override.............. none
  WAN to WAN Forwarding................ yes
  Block NetBIOS Default................ no
  BOOTP/DHCP Server address............ none
  Telnet Port.......................... default (23)
  Telnet Clients....................... all
  SNMP Port............................ default (161)
  SNMP Clients......................... all
  HTTP Port............................ default (80)
  HTTP Clients......................... all
  Syslog Port.......................... default (514)
  Allowed Syslog Servers............... all
  Default Syslog Servers............... none
  System message:
  Security timer....................... 10 minutes
  One WAN Dial Up...................... no
  Backup............................... no (no valid remote profile is
enabled)
    Retry Interval In Minutes.......... 30
    Stability Interval In Minutes...... 3

MODEM STRINGS:
  Reset:    ATZ
  Escape:   +++
  Init:     ATS0=0Q0V1&C1&D0X4S12=20
  Off-Hook: ATH1
  Dial:     ATDT
  Answer:   ATA
  Hangup:   ATH0


=== ETHERNET ===
GLOBAL BRIDGING/ROUTING SETTINGS:
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
    Bridge only PPPoE with dest........ no
  IP Routing enabled................... yes
```

```
    Multicast forwarding enabled....... no
    Firewall filter enabled ........... yes
    Directed Broadcasts Allowed........ no
    RIP Multicast address.............. default
    VRRP Multicast address............. default
  IPX Routing enabled.................. no

ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:09:0C:25
  Send IP RIP to the LAN............... rip-1 compatible
    Advertise me as default router..... yes
  Process IP RIP packets received...... rip-1 compatible
    Receive default route by RIP....... yes
  IP address translation............... no
  IP filters defined................... yes
  IP address/subnet mask............... 192.168.254.254/255.255.255.0
  Management IP address/subnet mask.... 0.0.0.0/0.0.0.0
  Static Ethernet routes defined....... none
 Virtual Ethernet routes defined...... none
  IPX External network number.......... 00000000
  IPX Frame type....................... 802.2
  MTU.................................. default


=== DHCP ===
       BOOTP/DHCP Relay address ..... none
       bootp tftpserver ............. none
       bootp file ................... n/a

Subnet 192.168.254.0, enabled
       When DHCP servers are active . stop
       Mask ......................... 255.255.255.0
       first ip address ............. 192.168.254.2
       last ip address .............. 192.168.254.20
       lease ........................ default
       bootp ........................ not allowed
       bootp server ................. none
       bootp file ................... n/a

       Client IP          State     Host Name           Expires
       192.168.254.2      enabled   QA-LABPC            Jun 24 2001
17:50:53

=== VOICE ===
VOICE DLCI is 22
Port  Pkts from Network/Dsp  VoiceRate   CallState              ChannelID
  1            0/         0  G711 uLaw   Inactive                    0
  2            0/         0  G711 uLaw   Inactive                    0
  3            0/         0  G711 uLaw   Inactive                    0
  4            0/         0  G711 uLaw   Inactive                    0
  5            0/         0  G711 uLaw   Inactive                    0
  6            0/         0  G711 uLaw   Inactive                    0
  7          198/       570  G711 uLaw   Inactive                    0
  8            0/         0  G711 uLaw   Inactive                    0

=== REMOTE DATABASE ===
INFORMATION FOR <configuredForCMPPlay>
  Status............................... enabled
  Interface in use..................... FR
```

```
  Protocol in use..................... RFC1483 (SNAP) - MAC Encapsulated
Routing
  Data Link Connection Id (DLCI)....... 528
  IP address translation............... on
  IP filters defined................... yes
  Send/Receive Multicast............... off
  Block NetBIOS Packets................ off
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Management IP address/subnet mask.... 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           0.0.0.0/0.0.0.0/1
  IPX network number................... 00000000
  Use IPX RIP/SAP (negotiate with PPP): yes
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
    Bridge only PPPoE with dest........ no
  mtu.................................. 1500


=== INTERFACES ===
Interface     Speed        In %     Out % Protocol     State
Connection
ETHERNET/0   10.0mb       0%/0%     0%/0% (Ethernet)   OPENED
FR/0          784kb       0%/0%     0%/0% (HDLC/FR)    OPENED
FR-VOICE/1    784kb       0%/0%     0%/0% (CLEAR)      OPENED
CONSOLE/0      57kb       0%/0%     0%/0% (TTY)        OPENED
FR-VC/2       784kb       0%/0%     0%/0% (FR)         OPENED       to
configuredForCMPPlay


=== PPP ===

=== BRIDGING ===
Bridging is disabled
Bridging is disabled

=== ARP TABLE ===
IP Addr              Mac Address             Interface
224.0.0.9            01:00:5E:00:00:09       ETHERNET/0
172.17.32.1          02:20:6F:09:0C:25       FR-VC/2

=== IP ROUTES ===
    IP route   /  Mask   --> Gateway         Interface    Hops Flags

0.0.0.0        /00000000 --> configuredForCMPPlay FR-VC/2      1 NW FW PRM
RP1 RP2
172.17.32.0    /ffffff00 --> configuredForCMPPlay FR-VC/2      1 NW FW DIR
PRM PRV
172.17.32.132  /ffffffff --> configuredForCMPPlay FR-VC/2      0 ME
192.168.254.0  /ffffff00 --> 0.0.0.0         ETHERNET/0   1 NW FW DIR PRM
RP1 RP2
```

```
192.168.254.254/ffffffff --> 0.0.0.0         ETHERNET/0   0 ME
224.0.0.9      /ffffffff --> 0.0.0.0         [none]       0 ME
224.0.0.18     /ffffffff --> 0.0.0.0         [none]       0 ME
255.255.255.255/ffffffff --> 0.0.0.0         [none]       0 NW PRM

=== IP IFS ===
FR-VC/2       172.17.32.132 (FFFFFF00) dest 0.0.0.0 sub 172.17.32.0
              net 172.17.0.0 (FFFF0000) BROADCAST  mtu 1500 mru 4096
              MAC address in use 02:20:6F:09:0C:25
              DHCP - lease good until Jul 24 2137 0:17:23
ETHERNET/0    192.168.254.254 (FFFFFF00) dest 0.0.0.0 sub 192.168.254.0
              net 192.168.254.0 (FFFFFF00) BROADCAST  mtu 1500 mru 1500
              MAC address in use 00:20:6F:09:0C:25

=== IPX ROUTES ===
No IPX sessions are active.

=== IPX SAPS ===
No IPX sessions are active.

=== L2TP TUNNELS ===

=== IP FILTERS ===

Begin IPFilters for configuredForCMPPlay
# watching for dropped/rejected packets is OFF

# Begin rules for input list
remote ipfilter flush  input configuredForCMPPlay
remote ipfilter insert 0 input accept -c 0 -p 50 -da 172.17.32.132 (IKE
Global Filter) configuredForCMPPlay
remote ipfilter insert 1 input accept -c 0 -p 51 -da 172.17.32.132 (IKE
Global Filter) configuredForCMPPlay
remote ipfilter insert 2 input accept -c 0 -p udp -sp 500 -da 172.17.32.132
-dp 500 (IKE Global Filter) configuredForCMPPlay
# End rules for input list

# Begin rules for receive list
remote ipfilter flush  receive configuredForCMPPlay
# End rules for receive list

# Begin rules for transmit list
remote ipfilter flush  transmit configuredForCMPPlay
remote ipfilter insert 0 transmit accept -c 0 -p udp -sa 172.17.32.132 -sp
500 -dp 500 (IKE Global Filter) configuredForCMPPlay
remote ipfilter insert 1 transmit accept -c 0 -p 50 -sa 172.17.32.132 (IKE
Global Filter) configuredForCMPPlay
remote ipfilter insert 2 transmit accept -c 0 -p 51 -sa 172.17.32.132 (IKE
Global Filter) configuredForCMPPlay
# End rules for transmit list

# Begin rules for output list
remote ipfilter flush  output configuredForCMPPlay
remote ipfilter insert 0 output accept -c 0 -p udp -sa 172.17.32.132 -sp 500
-dp 500 (IKE Global Filter) configuredForCMPPlay
# End rules for output list

End IPFilters for configuredForCMPPlay
```

```
Begin IPFilters for (ETHERNET/0)
# watching for dropped/rejected packets is OFF

# Begin rules for input list
eth ip filter flush  input 0
eth ip filter insert 0 input accept -c 0 -p 50 -da 192.168.254.254 (IKE
Global Filter) 0
eth ip filter insert 1 input accept -c 0 -p 51 -da 192.168.254.254 (IKE
Global Filter) 0
eth ip filter insert 2 input accept -c 0 -p udp -sp 500 -da 192.168.254.254
-dp 500 (IKE Global Filter) 0
# End rules for input list

# Begin rules for receive list
eth ip filter flush  receive 0
# End rules for receive list

# Begin rules for transmit list
eth ip filter flush  transmit 0
eth ip filter insert 0 transmit accept -c 0 -p udp -sa 192.168.254.254 -sp
500 -dp 500 (IKE Global Filter) 0
eth ip filter insert 1 transmit accept -c 0 -p 50 -sa 192.168.254.254 (IKE
Global Filter) 0
eth ip filter insert 2 transmit accept -c 0 -p 51 -sa 192.168.254.254 (IKE
Global Filter) 0
# End rules for transmit list

# Begin rules for output list
eth ip filter flush  output 0
eth ip filter insert 0 output accept -c 0 -p udp -sa 192.168.254.254 -sp 500
-dp 500 (IKE Global Filter) 0
# End rules for output list

End IPFilters for (ETHERNET/0)



=== IPSEC ===
There are no security associations.


=== IKE ===
There are no IKE peers.
There are no IKE proposals.
There are no IKE IPSec Proposals.
There are no IKE IPSec Policies.


=== END OF TECH SUPPORT DATA
```

SYSTEM SYSLOGPORT

This command manages Syslog port access. It can:

- Disable Syslog for this router (sets the Syslog port to 0).
- Request the default Syslog port (514). This re-enables Syslog after it is disabled.

- Redefine the Syslog port.

Note: This command requires a **save** and **reboot** to take effect.

To see the current setting, use the command **system list**. For more information on configuring the router as a Syslog client, see page 168. For more information on restricting port access, see Controlling Remote Management, on page 107.

system syslogport default | disabled | <*port*>

default	Restores the port value to the default value **514** and re-enables the port.
disabled	Disables the Syslog port.
port	Defines a new Syslog port number. Use this option to restrict remote access.

Examples:

```
system syslogport default

system syslogport disabled

system syslogport 3333
```

SYSTEM TELNETPORT

The router has a built-in Telnet server. This command can:

- Disable the Telnet server (sets the TCP port to 0).
- Request the default TCP port (23). This re-enables the Telnet server after it is disabled.
- Specify which router's TCP port is to receive a Telnet connection.

Note: This command requires a **save** and **reboot** to take effect.

To see the current setting, use the command **system list**.

system telnetport default | disabled | <*port*>

default	Requests the default port value (23).
disabled	Disables the Telnet server. The router will not accept any incoming TCP request.
port	Port number of the Ethernet LAN. It is recommended that this number be > 2048 if not 0 (disabled) or 23 (default).

Examples:

```
system telnetport default

system telnetport disabled

system telnetport 3333
```

SYSTEM WAN2WANFORWARDING

Allows the user to manage WAN-to-WAN forwarding of data from one WAN link to another.

For example, an employee uses the router at home to access both a company network and the Internet at the same time. To prevent the passing of company information to the Internet, WAN-to-WAN forwarding should be disabled.

To see the current setting for WAN to WAN forwarding, use the command **system list**.

This **system wan2wanforwarding** command complements the **system oneWANdialup** command (page 251). That command allows you to limit WAN connections to just one remote location at a time.

system wan2wanforwarding on | off

on	Allows data to be forwarded from one WAN link to another WAN link.
off	Stops data from being forwarded from one WAN link to another WAN link.

Example: `system wan2wanforwarding on`

Ethernet Interface Commands

The commands in this section begin with the word **eth**. The commands configure the Ethernet interfaces in your router. You can:

- Set the Ethernet LAN IP address
- Define logical interfaces to provide service to multiple IP subnets
- Manage the contents of the default routing table and any virtual routing tables
- Enable and disable IP routing
- List the current configuration settings

Note: In general, these commands require a **save** and **reboot** before they take effect. However, changes made to IP filters and to virtual routing tables take effect immediately; the changes are lost, though, if they are not saved before the next **reboot**.

ETH ?

Lists the supported keywords.

eth ?

Example:

```
# eth ?
Ethernet commands:

?           add           delete
br          ip            ipx
list        mtu

# eth ip ?
eth ip sub-commands
?           addr          ripmulticast
options     enable        disable
firewall    directedBcast addroute
delroute    defgateway    bindRoute
unbindRoute filter
```

ETH ADD

Adds a logical interface onto an Ethernet port so that the router can provide service to multiple IP subnets. The **eth add** command defines the port number and logical interface number. You should then use an **eth ip addr** command to define the IP subnet that uses the logical interface. For more information, see IP Subnets, on page 79.

A logical interface 0 always exists for Ethernet port 0 (and for port 1 in a dual-port router); logical interface 0 cannot be deleted.

Once defined, routes and filters can be created for the new logical interface using the other **eth** commands in this section. To list the currently defined logical interfaces, use the **eth list** command (page 280). To remove a logical interface, use an **eth delete** command (page 263).

Note: This command requires a **save** and **reboot** before it takes effect.

eth add *<port#>:<logical#>*

port# Ethernet interface (0 for a single-port router; 0 or 1 for a dual-port router).

logical# New logical interface number. It cannot be 0 because logical interface 0 always exists.

Example: `eth add 0:1`

ETH DELETE

Deletes a logical interface from an Ethernet port. For more information, see IP Subnets, on page 79

When you delete a logical interface, all information defined for that interface, such as routes and filters, is deleted automatically.

To list all currently defined logical interfaces, use the **eth list** command (page 280).

Note: Logical interface 0 cannot be deleted.

Note: This command takes effect immediately; however, if the change is not saved before the next **reboot**, the deletion is lost and the deleted interface reappears after the reboot.

eth delete *<port#>:<logical#>*

port# Ethernet interface (0 for a single-port router; 0 or 1 for a dual-port router).

logical# Logical interface number. (It cannot be 0.)

Example: `eth delete 0:1`

ETH IP ADDHOSTMAPPING

Remaps a range of local LAN IP addresses to a range of public IP addresses on a *per-interface basis*. These local addresses are mapped one-to-one to the public addresses. For more information, see Host Remapping, on page 99.

Note: The range of public IP addresses is defined by *<first public addr>* only. The rest of the range is computed automatically (from *<first public addr>* to *<first public addr>* + number of addresses remapped - 1) inclusive.

eth ip addHostMapping *<first private addr> <second private addr> <first public addr> <interface>*

first private addr First IP address in the range of IP address (4 decimals separated by periods).

second private addr Last address in the range of IP address (4 decimals separated by periods).

first public addr Defines the range of public IP addresses (4 decimals separated by periods). The rest of the range is computed automatically.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
eth ip addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 1
```

ETH IP ADDR

Defines the IP address and subnet mask for an Ethernet port or logical interface.

eth ip addr <*ipaddr*> <*ipnetmask*> [<*interface*>]

ipaddr Ethernet LAN IP address (4 decimals separated by periods.)

ipnetmask IP network mask (4 decimals separated by periods.)

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Examples:

The following command sets the IP address and subnet mask for the default Ethernet interface (0:0).

```
eth ip addr 192.168.1.254 255.255.255.0
```

The following command sets the IP address and subnet mask for logical interface 1 on Ethernet port 0.

```
eth ip addr 10.0.27.1 255.255.255.0 0:1
```

ETH IP ADDROUTE

Adds a route to the default routing table for the Ethernet interface.

This command is needed only if the system does not support RIP (see RIP Controls, on page 83 and the **eth ip options** command, on page 276).

Note: This command requires a **save** and **reboot** before it takes effect.

eth ip addRoute <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*interface*>]

ipaddr Ethernet LAN IP address (4 decimals separated by periods).

ipnetmask IP network mask (4 decimals separated by periods).

gateway IP address (4 decimals separated by periods).

hops Number of routers through which the packet must go to get to its destination.

interface Ethernet interface through which the packet is sent out. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Examples:

The following command adds a route to the default routing table for the default Ethernet interface (0:0).

```
eth ip addRoute 10.1.2.0 255.255.255.0 192.168.1.17 1
```

The following command adds a route to the default routing table for logical interface 1 on Ethernet port 0.

```
eth ip addRoute 10.1.3.0 255.255.255.0 10.0.27.20 1 0:1
```

ETH IP ADDSERVER

This Network Address Translation (NAT) command adds a server's IP address (on the LAN) associated with this interface for a particular protocol. For more information, see Network Address Translation (NAT), on page 95.

To delete a server designation, use the command **eth ip delserver** (page 268).

eth ip addServer <*action*> <*protocol*> <*first port*> [<*last port*> [<*first private port*>]] <*interface*>

action One of the following command actions:

ipaddr	Selects the host with this IP address as server (4 decimals separated by periods).
discard	Discards the incoming server request.
me	Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.

protocolid	Numeric protocol ID.
tcp	TCP only.
udp	UDP only.
all	All protocols.

first port First or only port as seen by the Ethernet interface. Port used by the selected server

portid	Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
ftp	FTP port.
h323	H.323 port.
http	HTTP port.
smtp	SMTP port.
sntp	SNTP port.
t120	T.120 port
telnet	Telnet port.
tftp	TFTP port.

all All ports.

last port Optional last port in the range of ports as seen by the Ethernet interface for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the Ethernet interface.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
eth ip addServer 192.168.1.5 tcp smtp 1

eth ip addServer 192.168.1.10 tcp 9000 9000 telnet 0
```

ETH IP BINDROUTE

Adds an Ethernet route to the named IP virtual routing table.

Duplicate routes are not allowed within a routing table. However, identical routes may be added to different routing tables. For example, the same route may be added to a virtual routing table and to the default routing table.

To list the routes, use the **iproutes** command, page 215. To remove an Ethernet route from a virtual routing table, use the **eth ip unbindRoute** command, page 277.

Note: A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

eth ip bindRoute <*ipaddr*> <*ipnetmask*> <*hops*> [<*gateway*>] <*tablename*> [<*interface*>]

ipaddr Ethernet LAN IP address (4 decimals separated by periods).

ipnetmask IP network mask (4 decimals separated by periods).

hops Number of routers through which the packet must go to get to its destination.

gateway IP address of the gateway (4 decimals separated by periods).

tablename IP virtual routing table to which the route is added.

interface Ethernet interface through which the packet is sent out. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

The following commands add a route for IP address 10.1.2.0/255.255.255.0 to four routing tables: ROSA, MIGUEL, FRANCISCO, and the default routing table. The first two routes are for Ethernet interface 0:1 and use gateway 192.168.252.9; the second two are for the default Ethernet interface (0:0) and, therefore, specify another gateway (192.168.252.7).

```
eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.9 ROSA 0:1

eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.9 MIGUEL 0:1

eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.7 FRANCISCO

eth ip addRoute 10.1.3.0 255.255.255.0 1 192.168.252.7
```

ETH IP DEFGATEWAY

Assigns an Ethernet default gateway for packets whose destination address does not have a route defined.

This setting is most useful when IP routing is not enabled, in which case the system acts as an IP host (i.e., an end system, as opposed to an IP router).

Note: This command requires a **save** and **reboot** before it takes effect.

Note: The following command is recommended instead of the **eth ip defgateway** command. It sends packets for all IP addresses to the specified gateway:

eth ip addRoute 0.0.0.0 255.255.255.0 <*gateway*> **1**

eth ip defgateway <*ipaddr*> *[*<*interface*>]

ipaddr	Ethernet LAN IP address (4 decimals separated by periods).
interface	Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface. If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified. To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example: `eth ip defgateway 192.168.1.1`

ETH IP DELHOSTMAPPING

Undoes an IP address/ host translation (remapping) range that was previously established with the command **eth ip addHostMapping** on a *per-interface basis (*page 263*)*. For more information, see Host Remapping, on page 99.

eth ip delHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*interface*>

first private addr	First IP address in the range of IP address (4 decimals separated by periods).
second private addr	Last address in the range of IP address (4 decimals separated by periods).
first public addr	Defines the range of public IP addresses (4 decimals separated by periods). The rest of the range is computed automatically.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
eth ip delHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 1
```

ETH IP DELROUTE

Removes a route from the default routing table that was added using the **eth ip addroute** command.

The route to be deleted is identified by its IP address and mask and its Ethernet interface. To see the remaining routes, use the **iproutes** command (page 215).

Note: This command requires a **save** and **reboot** before it takes effect.

eth ip delRoute <*ipaddr*> <*ipnetmask*> [<*interface*>]

ipaddr Ethernet LAN IP address (4 decimals separated by periods).

ipnetmask IP network mask (4 decimals separated by periods).

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Examples:

The following command deletes the route for IP address 10.9.2.0/255.255.255.0 for the default Ethernet interface (0:0).

```
eth ip delRoute 10.9.2.0 255.255.255.0
```

The following command deletes the route for IP address 10.1.3.0/255.255.255.0 for the Ethernet interface 0:1.

```
eth ip delRoute 10.1.3.0 255.255.255.0 0:1
```

ETH IP DELSERVER

Deletes an entry created by the **eth ip addServer** command (page 265).

eth ip delServer <*action*> <*protocol*> <*first port*> [<*last port*> [<*first private port*>]] <*interface*>

action One of the following command actions:

ipaddr	Selects the host with this IP address as server (4 decimals separated by periods).
discard	Discards the incoming server request.
me	Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.

protocolid	Numeric protocol ID.
tcp	TCP only.
udp	UDP only.
all	All protocols.

first port First or only port as seen by the Ethernet interface. Port used by the selected server

portid	Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
ftp	FTP port.
h323	H.323 port.
http	HTTP port.
smtp	SMTP port.
sntp	SNTP port.
t120	T.120 port
telnet	Telnet port.
tftp	TFTP port.
all	All ports.

last port Optional last port in the range of ports as seen by the Ethernet interface for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the Ethernet interface.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
eth ip delServer 192.168.1.5 tcp ftp 0
```

ETH IP DIRECTEDBCAST

Enables or disables the forwarding of broadcast packets directed to a specific network prefix. When forwarding is disabled, the router silently discards all packets broadcast to a subnet. The default is **off**; thus, by default, all network prefix-directed broadcast packets are discarded. This applies to all broadcast interfaces, including all Ethernet interfaces.

A network prefix-directed broadcast address is the broadcast address to a particular network. For example, if a network's IP address is 192.168.254.254 and its mask is 255.255.255.0, its network prefix-directed broadcast addresses are 192.168.254.0 and 192.168.254.255.

This feature is independent of the IP firewall and IP filtering features. However, it does require that IP routing be enabled (see **eth ip enable**, page 270). To see the current settings for IP routing and directed broadcasts, use the command **eth list**, page 280.

eth ip directedBcast on | off

on Enables the forwarding of packets broadcast to a subnet.

off Disables the forwarding of packets broadcast to a subnet. The default setting is **off**.

Example: `eth ip directedBcast on`

ETH IP DISABLE

Disables IP routing across the Ethernet LAN. This commands acts as a master switch allowing you to disable all IP routing for testing or control purposes.

Note: This command requires a **save** and **reboot** before it is effective.

eth ip disable

Example: `eth ip disable`

ETH IP ENABLE

Enables IP routing across the Ethernet LAN. This command acts as a master switch allowing you to re-enable all IP routing.

Note: This command requires a **save** and **reboot** before it is effective.

eth ip enable

Example: `eth ip enable`

ETH IP FILTER

Manages the IP filters for the Ethernet interface(s). The filters are used to screen IP packets.

Each Ethernet interface can have its own set of filters. The intended interface is designated at the end of the filter command. If the router has two physical Ethernet interfaces (an Ethernet hub router), the interface is designated by its port number (0 or 1). If logical interfaces have been defined to provide service to multiple IP subnets, the logical interface number is also specified (*port #:<logical #*, for example, 0:1).

Each interface can have filter lists that are applied at up to four points in the process: Input, Receive, Transmit, and Output. For more information on how and when the filter types are applied, refer to *IP Filtering, on page 129*.

Note: IP filters take effect immediately upon entry. They can even affect the current connection that you are using to enter commands. Unlike other configuration changes, you do not need to **save** and **reboot** or **restart**.

eth ip filter *<command> <type> <action> [<parameters>] [<interface>]*

The following commands are provided for managing IP filters for an Ethernet interface:

eth ip filter append [<*line number*>] <*type*> <*action*> [<*parameters*>] *[<interface>]*

Appends a filter to the list of filters for this <*type*> and <*interface*>. The filter is specified by the <*action*> and optional <*parameters*>.

If no line number is specified, the filter is appended to the end of the list; otherwise, it is appended after the specified line. For example, “append 0” appends the filter after line 0. Filters are used in the order they appear in their list.

eth ip filter insert [<*line number*>] <*type*> <*action*> <*parameters*> *[<interface>]*

Inserts a filter in the list of filters for this <*type*> and <*interface*>. The filter is specified by the <*action*> and optional <*parameters*>.

If no line number is specified, the filter is inserted at the beginning of the list; otherwise, it is inserted before the specified line. For example, “insert 0” inserts the filter before line 0 so it is the first filter in the list. Filters are used in the order they appear in their list.

eth ip filter delete <*type*> <*action*> <*parameters*> *[<interface>]*

Deletes the first filter that matches the filter specified on the command.

eth ip filter flush [*<first line> [<last line>]*] <*type*> *[<interface>]*

Deletes a range of filters from the list for this <*type*> and <*interface*>.

If no line numbers are specified, all filters in the list are deleted. If only the first line number is specified, all filters from that line to the end are deleted. To see the current filter list, use the **eth ip filter list** command. Filters are used in the order they appear in their list.

eth ip filter clear [*<first line> [<last line>]*] [<*type*>] <*clear arg*> *[<interface>]*

Resets the counters for the specified filters. A filter has a counter if the **-c** parameter was specified when the filter was defined.

You can specify the filters whose counters are to be reset by their line number range and type (input, output, or forward). If no type is specified, the counters for all filters for the interface are reset. If no line numbers are specified, the counters for all filters for that type and interface are reset. If only the first line number is specified, all counters for filters from that line to the end of the list are reset. To see the line numbers and counters, use the **eth ip filter list** command.

eth ip filter check <*type*> <*parameters*> *[<interface>]*

Checks the action that would be taken if a packet with the specified parameters was compared with the list of filters defined for the specified type and interface. For example, the command

eth ip filter check input -p TCP 1

would check what action (accept, drop, reject, inipsec, outipsec) would be taken for a TCP packet after it was compared with the list of input filters defined for port 1.

eth ip filter list <*type*> *[<interface>]*

Lists all filters of the specified <*type*> defined for the specified <*interface*>.

eth ip filter watch <on | off> [-q | -v] *[<interface>]*

Turns on or turns off the console watch for the interface. If the watch is on, a message is printed to the console serial port when a packet is dropped or rejected. (The message is also sent to any Syslog servers; see Syslog Client, on page 168.)

However, if the parameter **-q** (quiet) was specified for a filter, no message is printed when that filter matches a packet. If the parameter **-v** (verbose) was specified for a filter, a message is printed whenever that filter matches a packet, regardless of the filter action.

To see the messages, Telnet to the router and enter **system log start**. The watch does not continue after a reboot; to resume the watch after a reboot, you must enter the **eth ip filter watch on** command again.

The filter *type* specifies at which point the filter is compared to the IP packet (see the illustration under Filters and Interfaces, on page 129):

input When the packet enters the interface, *before* any network address translation is performed.

receive When the packet enters the interface, *after* any network address translation, but before routing table processing.

transmit After routing table processing, *before* any network address translation before the packet is sent out.

output After routing *and* network address translation, just before the packet is sent out.

If the packet matches the filter, the specified *action* is performed:

accept The packet is allowed to proceed for further processing.

drop The packet is discarded, without sending an ICMP (Internet Control Management Protocol) error message.

reject The packet is discarded and an ICMP error message is returned to the sender.

inipsec The packet is passed to IPSec for decrypting. The filter is intended to match packets coming from the other IPSec gateway. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 149).

outipsec The packet is passed to IPSec so it can be encrypted and sent to the other IPSec gateway. The filter is intended to match packets coming from the local protected network. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 149).

The following *parameters* specify the characteristics that an IP packet must have in order to match the filter. A filter can require any or all of these characteristics.

-p <*protocol*> | **TCP** | **UDP** | **ICMP**
The packet must have the specified protocol. If no protocol is specified, the filter matches *every* protocol.

-sa <*first source ip addr*>[:<*last source ip addr*>]
The packet must have a source IP address within the specified address range. If only one address is specified, the packet must have that source IP address. If no source IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

-sm <*source ip mask*>
The filter uses the specified mask when comparing the <*first source ip addr*>...<*last source ip addr*> with the source IP address in the IP packet. If no source mask is specified, the mask used is 255.255.255.255.

-sp <*ICMP type*> | <*first source port*>[**:**<*last source port*>]
The packet must have a source port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that source port. If no source port is specified, the filter matches any source port in the range 0**:**0xffff.

-da <*first dest ip addr*>[:<*last dest ip addr*>]
The packet must have a destination IP address within the specified address range. If only one address is specified, the packet must have that destination IP address. If no destination IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

-dm <*dest ip mask*>
The filter uses the specified mask when comparing the <*first dest ip addr*>...<*last dest ip addr*> with the destination IP address in the IP packet. If no destination mask is specified, the mask used is 255.255.255.255.

-dp <*ICMP type*> | <*first dest port*>[**:**<*last dest port*>]
The packet must have a destination port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that destination port. If no destination port is specified, the filter matches any destination port in the range 0**:**0xffff.

-tcp syn | **ack** | **noflag** | **rst**
If the IP packet is a TCP packet, the filter matches the packet only if the packet flag settings are as specified. If no **-tcp** option is specified for the filter, flag settings are not checked.

Note: You may specify *more* than one **-tcp** option for the IP filter.

The **syn**, **ack**, and **noflag** settings work together as follows:
- Specify **-tcp syn** if the TCP SYN flag must be set.
- Specify **-tcp ack** if the TCP ACK flag must be set.
- Specify **-tcp noflag** if neither the SYN flag nor the ACK flag can be set.

For example, for the IP filter to match the initiation of a TCP connection, specify **-tcp syn**. The filter will match TCP packets that have the TCP SYN flag set but *not* the TCP ACK flag set. For the filter to match the response to initiation of a TCP connection, specify **-tcp syn** *and* **-tcp ack**. The filter will match only TCP packets with *both* the TCP SYN and TCP ACK flags set.

The **-tcp rst** setting is independent of the others; if you specify **-tcp rst** for the filter, the filter matches every TCP packet with the TCP RESET flag set, regardless of the other flag settings. For example, for the filter to match packets for “established” connections, you would specify both **-tcp rst** and **-tcp ack** so that the filter is applied to every TCP packet that has either the RESET flag or the ACK flag set.

The following *parameters* request additional filter options.

-b
This option requests that this filter be compared *twice* with each packet. The first time the source filter information is matched against the source information in the IP packet and the destination filter information is matched against the destination information in the IP packet. The second time the source filter information is matched against the destination information in the IP packet and the destination filter information is matched against the source information in the IP packet.

-c <*count of times rule used*>
This option requests a counter for this filter. If specified, a count is kept of how many IP packets have

matched this filter since the router was rebooted. To see the current count for a filter, use the **eth ip filter list** command. To clear a counter, use the **eth ip filter clear** command.

-ipsec <*IPSec record name*>

Use this option when the *action* specified is **inipsec** or **outipsec**. It specifies the IPSec Security Association that uses the filter.

-q or **-v**

Specify one of these options to determine when watch messages are sent for this filter. The messages are sent to the console serial port (and to any Syslog servers; see page 168).

If neither **-q** or **-v** are specified for the filter, and an **eth ip filter watch on** command is entered for the interface, a message is sent each time this filter causes a packet to be dropped or rejected.

If **-q** (quiet) is specified, no messages are printed for this filter, even if the filter causes a packet to be dropped or rejected.

If **-v** (verbose) is specified, a message is printed every time this filter matches a packet, regardless of the filter action.

The optional *interface* determines which Ethernet interface the filter applies to.

If the router has only one Ethernet interface, <*interface*> may be omitted.

If the router has two physical Ethernet interfaces (that is, a dual-port router), you must specify the port by its number (0 or 1).

If logical interfaces have been defined for the physical Ethernet interface, the port number and the logical interface number are specified (<*port #*>:<*logical #*>, for example, 0:1).

`Examples:`

This command clears all filters from the Input filter list for Ethernet interface 0. Use this command as the first command in a list of commands starting a new Input filter list.

```
eth ip filter flush input 0
```

These commands prevent the forwarding of all IP traffic. If you put these filters at the end of the filter lists, they will stop all packets that have not matched filters earlier in the lists.

```
eth ip filter append receive drop
eth ip filter append transmit drop
```

ETH IP FIREWALL

The router supports IP Internet Firewall Filtering to prevent unauthorized access to your system and network resources from the Internet. This filter discards packets received from the WAN that have a source IP address recognized as a local LAN address. This command sets Ethernet Firewall Filtering on or off and allows you to list the active state.

Note 1: This command requires a **save** and **reboot** before it takes effect.

Note 2: To perform Firewall Filtering, IP routing must be enabled. For more information, see Internet Firewall Filtering, on page 82.

eth ip firewall on | off | list

on — Sets firewall filtering on. IP routing *must* also be enabled for filtering to be performed.

off — Sets firewall filtering off.

list — Lists the current status of firewall filtering.

Example:

```
# eth ip firewall list
The Internet firewall filter is currently on.
0 offending packets were filtered out.
```

ETH IP MGMT

This command assigns to an Ethernet interface an IP address which is to be used for management purposes only and not for IP address translation. This management IP address is generally a private network address used solely by the ISP.

The management IP address is separate from the IP address used for IP address translation. The IP address used for address translation is generally a public IP address valid on the Internet. It is set by the **eth ip addr** command (page 264).

Note: The management address is not effective until after the next **save** and **reboot**.

Note: To use the management address as the source address for a ping, you must specify it using the **-I** option on the **ping** command (page 217). For example, to use management address 192.168.1.2 when pinging destination address 192.168.100.100, specify:

```
ping -I 192.168.1.2 192.168.100.100
```

Note: To use the management address as the source address for a copy, you must specify both the source and destination addresses on the **copy** command (page 226).

To list the current management address for the Ethernet interface, if any, use the **eth list** command (page 280). To set a management address for the WAN interface, see **remote setMgmtIpAddr** (page 314).

eth ip mgmt <*ipaddr*> <*ipnetmask*> *[<interface>]*

ipaddr — IP address (4 decimals separated by periods).

ipnetmask — IP subnet mask (4 decimals separated by periods).

interface — Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
# eth ip mgmt 10.0.0.2 255.255.255.0 0:1
# save
```

```
# reboot
```

ETH IP OPTIONS

Turns on or turns off an IP option for the specified Ethernet interface. The IP options include:

- Options to transmit or receive **RIP-1** and/or **RIP/2** packets. (See RIP Controls, on page 83.)
- Option to advertise this router as the **default router**.
- Option to enable forwarding of **IP multicast traffic**.

Note: This command requires a **save** and **reboot** before it takes effect.

eth ip options <*option*> on | off [<*interface*>]

option Must be one of the following:

rxrip	Receive and process IP RIP-1 compatible and RIP-2 broadcast packets from the Ethernet LAN. Also receive and process RIP-2 packets that are multicast as defined by the **eth ip ripmulticast** command. Set this option if the local router is to discover route information from the Ethernet LAN. The default is on.
rxrip1	Receive and process RIP-1 packets only.
rxrip2	Receive and process RIP-2 packets only.
rxdef	Receive the default route address from the Ethernet LAN. The default is on. This option is useful if you do not want to configure your router with a default route.
txrip	Transmit RIP-1 compatible broadcast packets and RIP-2 multicast packets over the Ethernet LAN. The default is on.
txrip1	Transmit broadcast RIP-1 packets only.
txrip2	Transmit multicast RIP-2 packets only.
txdef **avdfr**	Advertise this router as the default router over the Ethernet LAN (provided it has a default route). The default is on. Set this to off if another router on the local LAN is the default router.
multicast	Enables this Ethernet interface to forward IP multicast traffic. **Note:** If any remote has multicast forwarding enabled, multicast forwarding is enabled on all Ethernet interfaces automatically. However, you can disable forwarding for a specific interface using this command.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

```
eth ip options avdfr off
```

ETH IP RIPMULTICAST

Changes the multicast address for RIP-1 compatible and RIP-2 packets. The default address is 224.0.0.9.

For more information, see RIP Controls, on page 83.

eth ip ripmulticast <*ipaddr*>

ipaddr IP address of the remote network or station (4 decimals separated by periods).

Example: `eth ip ripmulticast 239.192.0.9`

ETH IP TRANSLATE

This command is used to control Network Address Translation on a *per-interface basis*. It allows several PCs to share a single IP address to the Internet. To read more about Network Address Translation (NAT), see page 95.

eth ip translate on | off <*interface*>

on | off Indicates whether Network Address Translation is on or off for this Ethernet interface.

interface Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

This command enables Network Address Translation for port 0.

```
eth ip translate on 0
```

This command disables Network Address Translation for logical interface 0:1

```
eth ip translate off 0:1
```

ETH IP UNBINDROUTE

Removes an Ethernet route from the named IP virtual routing table.

To list the routes, use the **iproutes** command, page 215. To add an Ethernet route to a virtual routing table, use the **eth ip bindRoute** command.

Note: A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

eth ip unbindRoute <*ipaddr*> <*tablename*> [<*interface*>]

ipaddr Ethernet LAN IP address (4 decimals separated by periods).

tablename IP virtual routing table from which the route is removed.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

The following commands remove Ethernet routes from virtual routing table ROSA. The first deleted route is for IP address 10.1.2.0 and the default Ethernet interface (0:0). The second deleted route is for IP address 10.1.3.0 and the logical Ethernet interface 0:1.

```
eth ip unbindRoute 10.1.2.0 ROSA

eth ip unbindRoute 10.1.3.0 ROSA 0:1
```

ETH IP VRID

Assigns a virtual router ID (VRID) to an Ethernet interface. The same VRID must be assigned to the master router and its backup routers. For more information, see VRRP Backup, on page 116.

This command designates the interface as the VRRP interface for the router. You must use another logical Ethernet interface as the management interface for the router. To create a new logical Ethernet interface, use the command **eth add** (page 262) and then assign it an IP address with an **eth ip addr** command (page 264).

Note: The assignment takes effect after you **save** the change and **restart** the interface or **reboot** the router.

After you assign the VRID, you specify its attributes with the **eth vrrp** commands (see page 282).

If you delete the VRID (**eth vrrp delete**), the VRRP interface designation is cleared. You can also clear the VRRP interface designation by entering the **eth ip vrid** command with **0** as the VRID.

eth ip vrid <*vrid*> *[*<*interface*>]

vrid Virtual router ID (integer, 1-255).

If you specify **0** as the VRID, the Ethernet interface is no longer the VRRP interface.

interface Ethernet interface. The default Ethernet interface is 0:0.

To specify a logical interface other than 0:0, specify both the port number (0 or 1) and the logical interface number using the format <*port #*>:<*logical #*> (for example, 0:1).

Examples:

This command assigns VRID 7 to the logical Ethernet interface 0:1.

```
eth ip vrid 7 0:1
```

This command clears the VRRP interface designation from interface 0:1.

```
eth ip vrid 0 0:1
```

This command assigns VRID 1 to the default logical Ethernet interface 0:0.

```
eth ip vrid 1
```

ETH IPX ADDR

Sets the IPX network number for the Ethernet LAN connection.

eth ipx addr <*ipxnet*> [*port#*]

ipxnet IPX network number represented by 8 hexadecimal characters.

port# Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

Example: `eth ipx addr 123`

ETH IPX DISABLE

Disables IPX routing across the Ethernet LAN. This acts as a master switch allowing you to disable IPX routing for testing or control purposes.

Note: This command requires a reboot.

eth ipx disable [*port#*]

port# Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

Example: `eth ipx disable`

ETH IPX ENABLE

Enables IPX routing across the Ethernet LAN. This acts as a master switch that allows you to enable IPX routing.

Note: This command requires a reboot.

eth ipx enable [*port#*]

port# Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

Example: `eth ipx enable`

ETH IPX FRAME

Sets the frame encapsulation method. The default is 802.2.

eth ipx frame <*type*>

type 802.2 (DEC standard)
802.3 (Intel standard)
dix (Xerox/Ethernet II standard)

Example: `eth ipx frame 802.3`

ETH LIST

Lists information about the Ethernet interfaces including the status of bridging and routing, IP protocol controls, and IP address and subnet mask.

eth list [<*interface*>]

interface Ethernet interface for which information is listed. If the parameter is omitted, information is listed for all Ethernet interfaces in the router.

For a dual-port router, you may specify the port number (0 or 1).

If logical interfaces are defined, you may specify a port and logical interface number (<*port #*>:<*logical interface #*>, such as 0:1).

Example:

```
# eth list
GLOBAL BRIDGING/ROUTING SETTINGS:
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
  IP Routing enabled................... yes
    Multicast forwarding enabled....... no
    Firewall filter enabled............ yes
    Directed Broadcasts Allowed........ no
    RIP Multicast address.............. default
  IPX Routing enabled.................. no

ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:98:04
  Send IP RIP to the LAN.............. no
    Advertise me as default router..... yes
  Process IP RIP packets received...... no
    Receive default route by RIP....... yes
  IP filters defined................... no
  IP address/subnet mask............... 192.168.0.101/255.255.255.0
  Static Ethernet routes defined....... 1
    IP address/subnet mask............. 0.0.0.0/0.0.0.0
    IP gateway/metric.................. 192.168.0.252/1
  Virtual Ethernet routes defined...... none
  IPX External network number.......... 00000000
  IPX Frame type....................... 802.2
```

```
MTU.............................. default
```

ETH MTU

Sets the maximum transfer unit for the Ethernet interface. The default is 1500 bytes.

You can set the MTU size to less than 1500 bytes, but you cannot set the MTU to greater than 1500 bytes, even if you specify a larger value on an **eth mtu** command. (RFC 1042 recommends 1500 bytes as the maximum MTU for an Ethernet network.)

To see the current MTU size for an interface that has IP enabled, use the **ipifs** command (page 215).

eth mtu <*size*> [<*interface*>]

size Maximum number of bytes that can be transferred as a unit.

interface Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

Example:

The following command decreases the MTU size for Ethernet interface 0:1 to 1400 bytes.

```
eth mtu 1400 0:1
```

ETH RESTART

Stops and restarts a logical Ethernet interface. To read about logical Ethernet interfaces, see page 79.

Certain configuration changes for a logical Ethernet interface become effective only after the logical interface is restarted or the router is rebooted. Remember to **save** the changes before the restart or reboot.

Note: Use **restart** instead of **reboot** whenever possible. A restart does not affect other interfaces, allowing their traffic to continue. For example, using restart, you can add an IP route without killing voice traffic.

To restart an remote interface, use **remote restart** (page 307).

eth restart <*interface*>

interface Logical Ethernet interface. Specify both the port number and the logical interface number using the format <*port #*>:<*logical #*> (for example, 0:1).

Example:

The following command restarts logical Ethernet interface 0:1.

```
eth restart 0:1
```

ETH START

Starts a stopped logical Ethernet interface. To read about logical Ethernet interfaces, see page 79.

A logical Ethernet interface is stopped using the command **eth stop** (page 282). To stop and immediately restart a logical Ethernet interface, use the command **eth restart** (page 281).

eth start <*interface*>

interface Logical Ethernet interface. Specify both the port number and the logical interface number using the format <*port #*>:<*logical #*> (for example, 0:1).

Example:

The following command starts logical Ethernet interface 0:1.

```
eth start 0:1
```

ETH STOP

Stops a logical Ethernet interface. To read about logical Ethernet interfaces, see page 79.

Note: To keep certain configuration changes, you must enter a **save** command before stopping the logical interface.

The stopped interface is disabled until it is started again. To start a logical Ethernet interface, use the command **eth start** (page 282). To stop and immediately restart a logical Ethernet interface, use the command **eth restart** (page 281).

eth stop <*interface*>

interface Logical Ethernet interface. Specify both the port number and the logical interface number using the format <*port #*>:<*logical #*> (for example, 0:1).

Example:

The following command stops logical Ethernet interface 0:1.

```
eth stop 0:1
```

ETH VRRP ADD

Defines a VRRP attribute record for the VRID (virtual router ID). Attribute records must be defined for the VRID in the master router and in each of its backup routers. For more information, see VRRP Backup, on page 116.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

To see the contents of the VRRP attribute records, use the command **eth vrrp list** (page 284). You can change the attribute values using other **eth vrrp** commands (see Defining VRRP Attributes, on page 118.)

eth vrrp add <*vrid*> *[<port#>]*

vrid Virtual router ID (integer, 1-255). It is defined by an **eth ip vrid** command (page 278).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Examples:

This command defines an attribute record for VRID 7 for the default port 0.

```
eth vrrp add 7
```

This command defines an attribute record for VRID 2 for port 1.

```
eth vrrp add 2 1
```

ETH VRRP CLEAR PASSWORD

Clears the password in a VRRP attribute record for the VRID (virtual router ID). To read more about VRRP Backup, see page 116.

Note: If the VRRP attribute record has no password, no VRRP authentication is performed.

Note: If you clear the password for one VRRP router, you must clear the password for every router for that VRID on the LAN. For example, if VRID 7 is defined in routers A, B, and C in the LAN and you clear the password for router A, you must clear the password for routers B and C as well.

To see the current password, use the command **eth vrrp list** (page 284). To set a new password, use the command **eth vrrp set password** (page 286).

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp clear password <*vrid*> [<*port#*>]

vrid Virtual router ID of the VRRP attribute record (integer, 1-255). The attribute record was created by the command **eth vrrp add** (page 282).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command clears the password for VRID 7 using default port 0.

```
eth vrrp clear password 7
```

ETH VRRP DELETE

Deletes a VRRP attribute record for the VRID (virtual router ID). It also disassociates the VRRP IP and MAC addresses from the logical interface. To read more about VRRP, see page 116.

Use this command to disable VRRP. To re-instate a deleted VRID, you need to redefine both the VRID and the VRRP attribute record.. For example, the following commands disable VRID 7 and then re-enable it for the logical interface 0:0:

```
# eth vrrp delete 7
# eth ip vrid 7
# eth vrrp add 7
# 04/16/2001-08:36:06:VRRP: VRRP 7 on Interface ETHERNET/0 now active
```

When removing a VRRP configuration from a router, you would delete both the VRRP attribute record and the extra logical interface. To do so, use the commands **eth vrrp delete** and **eth delete** (page 263).

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp delete <*vrid*> *[*<*port#*>*]*

vrid Virtual router ID (integer, 1-255).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command deletes the attribute record for VRID 7 for the default port 0.

```
eth vrrp delete 7
```

ETH VRRP LIST

Lists the VRRP attribute records for the port and shows the status of the VRRP router. To read more about VRRP, see page 116.

eth vrrp list [<*port#*>]

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command lists the attribute records for the default port 0.

```
# eth vrrp list
VRRP Records for Ethernet Port .... 0
  VRRP Record..................... VRID   8, Priority 100, Interval   1
                                   Flags: preempt, No Authentication

                                   Interface: ETHERNET/0:3
                                   Primary WAN inteface is down
```

ETH VRRP SET MULTICAST

Changes the multicast address used for VRRP router announcements. This address is used by all VRRP announcements from this router, regardless of VRID or port. To read more about VRRP Backup, see page 116.

Note: This command is not usually needed for VRRP configuration. Do not use this command unless you clearly understand its impact.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp set multicast *<ipaddr>*

ipaddr IP address that is to be the new multicast address (4 decimals, separated by periods).

Example:

This command specifies a new multicast address for VRRP.

```
eth vrrp multicast 192.168.255.255
```

ETH VRRP SET OPTION

Specifies the preemption option in a VRRP attribute record for the VRID (virtual router ID).

Note: The default for the attribute is **preempt**.

The preemption option determines what the router does when it recovers from a failure, as follows:

- If the router is the master router for the IP address (it has priority 255), it always immediately preempts the backup router and resumes its function in the network.The preemption option cannot change this.
- However, if the router is a backup router for the IP address and it determines that a router with a lower priority is currently functioning as backup, the preemption option determines whether this router immediately preempts the router with lower priority or waits for the lower priority router to go away before becoming the active VRRP router.

To read more about VRRP Backup, see page 116.

The preemption setting may differ among the backup routers for a VRID.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp set option preempt | nopreempt *<vrid>* [*<port#>*]

preempt Preempt immediately.

nopreempt Do not preempt a router with lower priority.

vrid Virtual router ID of the VRRP attribute record (integer, 1-255). The attribute record was created by the command **eth vrrp add** (page 282).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command specifies no preemption for VRID 7 using default port 0.

```
eth vrrp set option nopreempt 7
```

ETH VRRP SET PASSWORD

Specifies the password in a VRRP attribute record for the VRID (virtual router ID). The password is used to authenticate VRRP advertisement packets. It is sent as clear text on the LAN. To read more about VRRP Backup, see page 116.

Note: If you do not specify a password, no authentication is performed.

To see the current password, use the command **eth vrrp list**. To clear a password, use the command **eth vrrp clear password** (page 283).

Note: The password must be the *same* for every router in the Virtual Router, that is, for every router in the LAN with the same VRID. For example, if a VRRP interface in routers A, B, and C has the VRID 7, routers A, B, and C must all specify the same password for VRID 7.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp set password <*password*> <*vrid*> [<*port#*>]

password Password (1-8 characters). The password is case-sensitive.

vrid Virtual router ID of the VRRP attribute record (integer, 1-255). The attribute record was created by the command **eth vrrp add** (page 282).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command specifies the password “AbCdEfGh” for VRID 7 using default port 0.

```
eth vrrp set password AbCdEfGh 7
```

ETH VRRP SET PRIORITY

Specifies the priority attribute in a VRRP attribute record for the VRID (virtual router ID). The priority value determines which VRRP router in the LAN takes over when a VRRP router fails. For more information, see VRRP Backup, on page 116.

Note: If you do not specify a priority value for a VRRP attribute record, the default priority, **100**, is used.

The priority for the master router must be the maximum, 255; the priority for each backup router must be less than 255.

The priority values must *differ* for each router that uses the same VRID. For example, the master router for VRID 7 must have priority 255 while the first backup router for VRID 7 could have the default priority 100 and a second backup router for VRID 7 could have priority 50.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp set priority *<priority> <vrid> [<port#>]*

priority Priority value (integer, 1-255). The priority for the master router must be 255; the priority for each backup router must be less than 255.

vrid Virtual router ID of the VRRP attribute record (integer, 1-255). The attribute record was created by the command **eth vrrp add** (page 282).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Examples:

This command specifies the maximum priority for the master router for VRID 7 using default port 0.

```
eth vrrp set priority 255 7
```

This command defines priority 50 for a backup router for VRID 7 using port 1.

```
eth vrrp set priority 50 7 1
```

ETH VRRP SET TIMEINTERVAL

Specifies the time interval attribute in a VRRP attribute record for the VRID (virtual router ID). The time interval determines how often VRRP advertisement packets are sent, and thus, how quickly a backup router can recognize that another VRRP router is down.

Note: If you do not specify a time interval value for a VRRP attribute record, the default time interval, **1 second**, is used.

If the backup does not receive a VRRP packet from another VRRP router during the master down interval, the backup assumes the other router is down. The master down interval is calculated as follows:

Master _Down_Interval = (3 * Time_Interval) + Skew_Time

Skew_Time = (256 - Priority) / 256

Thus, the default skew time is (256 - 100) / 256, or .609375. The default master down interval is (3 * 1) + .609375, or 3.609375 seconds.

For more information, see VRRP Backup, on page 116.

Note: The time interval must be the *same* for every router in the Virtual Router, that is, for every router in the LAN with the same VRID. For example, if a VRRP interface in routers A, B, and C has the VRID 7, routers A, B, and C must all specify the same time interval for VRID 7.

Note: This command takes effect immediately, but you must **save** the change if it is to persist after you **restart** the interface or **reboot** the router.

eth vrrp set timeinterval <*seconds*> <*vrid*> [<*port#*>]

seconds Time interval value in seconds (integer).

vrid Virtual router ID of the VRRP attribute record (integer, 1-255). The attribute record was created by the command **eth vrrp add** (page 282).

port# Physical Ethernet interface (port) number (0 or 1). The default is 0; the parameter may be omitted if the router has only one port.

If the router has two ports (an Ethernet hub router), the port number (0 or 1) must be specified.

Example:

This command specifies two seconds as time interval for VRID 7 using default port 0.

```
eth vrrp set timeinterval 2 7
```

REMOTE Commands

The commands in this section begin with the word **remote**. The commands allow you to add, delete, and modify remote routers to which the target router can connect. Remote router information that can be configured includes:

- PVC numbers
- Security authentication protocols and passwords
- WAN IP/ IPX addresses
- IP routes
- IPX routes and SAPS
- Remote bridging addresses and bridging control
- Host mapping

REMOTE ?

Lists the supported keywords. (The list varies depending on the router model.)

remote ?

Example:

```
# remote ?
Sub-commands for remote:
?                   help                  add
del                 delete                list
enable              disable               start
stop                restart               setAuthen
enaAuthen           disAuthen             setPasswd
setOurPasswd        delOurPasswd          setOurSysName
delOurSysName       listPhones            setLNS
setL2TPClient       setProtocol           setPVC

setATMnsap          delATMnsap            setATMTraffic
setPhone            delPhone              setBWThresh
setBod              addCaller             delCaller
setMaxLine          setMinLine            setTimer
addHostMapping      delHostMapping        addServer
delServer           setIPTranslate        setIPslavePPP
setPPPOptions       ipfilter              blockNetBIOS
setCompression      stats                 statsclear
setRmtIpAddr        addIproute            delIproute
bindIPVirtualRoute  unbindIPVirtualRoute  setIpOptions
listIproutes        setIpxaddr            addIpxroute
delIpxroute         listIpxroutes         addIpxsap
delIpxsap           listIpxsaps           setIpxOptions
listBridge          setBrOptions          addBridge
delBridge           enaBridge             disBridge
```

```
setmtu
```

REMOTE ADD

Adds a remote router entry into the remote router database.

remote add <*remoteName*>

remoteName Name of the remote router (character string). The name is case-sensitive.

Example: `remote add HQ`

REMOTE ADDBRIDGE

Defines the remote router entry as the default bridging destination for outbound bridging. The command can define either the default bridging destination for all MAC addresses or the default bridging destination for a specific MAC address.

When you specify a MAC address on this command, a permanent entry for that address is created in the bridging table. Thereafter, packets that contain that MAC address are bridged using the specified remote router entry. (To see the entries in the bridging table, use the **bi list** command.)

Note: Bridging using the specified remote is effective only after it has been enabled using the **remote enabridge** command (page 300). To see the current bridge settings for a remote, use the **remote listbridge** command (page 305). To remove the default designation from a remote, use the **remote delbridge** command (page 295).

If IP and IPX routing are disabled, all packets, with an unknown destination, are bridged to the default bridging destination. If IP and/or IPX routing is enabled, bridging occurs only for packets that are not routed.

remote addbridge ***** | <*MAC_addr*> <*remoteName*>

***** All MAC addresses.

MAC_addr MAC address (six bytes, specified as six hexadecimals, separated by colons).

remoteName Name of the remote router (character string). The name is case-sensitive.

Example:

```
remote addbridge 01:08:03:0A:0B:0C HQ
```

REMOTE ADDHOSTMAPPING

Remaps a range of local LAN IP addresses to a range of public IP addresses on a *per-remote-router basis*. These local addresses are mapped one-to-one to the public addresses.

Note: The range of public IP addresses is defined by <*first public addr*> only. The rest of the range is computed automatically (from <*first public addr*> to <*first public addr*> + number of addresses remapped - 1) inclusive.

remote addHostMapping <*first private addr*><*second private addr*><*first public addr*><*remoteName*>

first private addr	First IP address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods.
second private addr	Last address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods.
first public addr	Defines the range of public IP addresses, in the format of 4 decimals separated by periods. The rest of the range is computed automatically.
remoteName	Name of the remote router (character string).
Example:	`remote addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 HQ`

REMOTE ADDIPROUTE

Adds an IP address route to a network or station on the LAN connected beyond the remote router. The route is added to the default routing table.

The local router's routing table must be seeded statically to access networks and stations beyond this remote router. After the connection is established, standard RIP update packets can dynamically add routes to the routing table. Setting this address is not required if the local router never connects to the remote router *and* the remote router supports RIP.

Note: Changes to the default routing table require a **save** and a **remote restart** or **reboot** before they take effect.

remote addIpRoute *<ipaddr> <ipnetmask> <hops> [<ipgateway>] <remoteName>*

ipaddr	IP address of the remote network or station (4 decimals separated by periods).
ipnetmask	IP network mask of the remote network or station (4 decimals separated by periods).
hops	Perceived cost to reach the remote network or station by this route (number between 1 and 15).
ipgateway	Address of a router on the remote LAN (4 decimals separated by periods). Enter a gateway only if you are configuring a MER interface. Check with your system administrator for details.
remoteName	Name of the remote router (character string).

Examples:

The first two addresses in the list represent subnetworks, the third is a class B network, the fourth is a host, and the fifth address is the default route. The fifth command adds the default route when the WAN interface is a point-to-point interface; the sixth command adds the default route when the WAN interface is a broadcast interface.

```
remote addIpRoute 10.1.210.64 255.255.255.192 1 HQ

remote addIpRoute 10.1.210.032 255.255.255.224 1 HQ

remote addIpRoute 172.17.0.0 255.255.0.0 2 HQ

remote addIpRoute 10.1.210.072 255.255.255.255 1 HQ

remote addIpRoute 0.0.0.0 0.0.0.0 1 HQ

remote addIproute 0.0.0.0 0.0.0.0 1 172.16.10.1 HQ
```

REMOTE ADDIPXROUTE

Adds an IPX route for a network or station on the LAN network connected beyond the remote router. The target router's routing information table must be seeded statically to access networks and stations beyond this remote router. After the connection is established, standard RIP update packets will dynamically add to the routing table. (Setting this address is not required if a target router never connects to the remote router and the remote router supports RIP.)

Note: A **reboot** command must be performed on the target router for the addition of a static route to take effect.

remote addIpxRoute <*ipxNe#*> <*metric*> <*ticks*> <*remoteName*>

ipxNe# IPX network number represented by 8 hexadecimal characters.

metric Number of routers through which the packet must go to get to the network/station.

ticks Number in 1/8 seconds which is the estimated time delay in reaching the remote network or station.

remoteName Name of the remote router (character string).

Example: `remote addIpxRoute 456 1 4 HQ`

REMOTE ADDIPXSAP

Adds an IPX SAP to the server information table for a service on the LAN network connected beyond the remote router. The target router's SAP table must be seeded statically to access services beyond this remote router. After the connection is established, standard SAP broadcast packets will dynamically add to the table.

Note: A **reboot** must be performed on the target router for the addition of a SAP to take effect.

remote addIpxSap <*servicename*> <*ipxNet* > <*ipxNode*> <*socket*> <*type*> <*hops*> <*remoteName*>

servicename Name of server.

ipxNet IPX network number represented by 8 hexadecimal characters.

ipxNode IPX node address represented by 12 hexadecimal characters.

socket Socket address of the destination process within the destination node. The processes include services such as file and print servers.

type Number representing the type of server.

hops Number of routers through which the packet must go to get to the network/station.

remoteName Name of the remote router (character string).

Example: `remote addIpxSap Fileserver 010a020b 0108030a0b0c 451 HQ`

REMOTE ADDSERVER

This Network Address Translation (NAT) command is used to add a server's IP address (on the LAN) associated with this remote router for a particular protocol. To learn more, see Network Address Translation (NAT), on page 95.

Multiple **system addserver** (page 233) and **remote addserver** commands can designate different servers for different protocols, ports, and interfaces. When a request is received, the router searches the server list for the appropriate server. The order of search for a server is discussed in Server Request Hierarchy, on page 98.

To delete a server designation, use the command **remote delserver** (page 298).

remote addServer *<action> <protocol> <first port>* [*<last port>* [*<first private port>*]] *<remoteName>*

action One of the following command actions:
- *ipaddr* Selects the host with this IP address as server (4 decimals separated by periods).
- **discard** Discards the incoming server request.
- **me** Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.
- *protocolid* Numeric protocol ID.
- **tcp** TCP only.
- **udp** UDP only.
- **all** All protocols.

first port First or only port as seen by the remote end. Port used by the selected server
- *portid* Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
- **ftp** FTP port.
- **h323** H.323 port.
- **http** HTTP port.
- **smtp** SMTP port.
- **sntp** SNTP port.
- **t120** T.120 port
- **telnet** Telnet port.
- **tftp** TFTP port.
- **all** All ports.

last port Optional last port in the range of ports as seen by the remote end for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the remote end.

remoteName Name of the remote router (character string).

Example:

```
remote addServer 192.168.1.5 tcp smtp
remote addServer 192.168.1.10 tcp 9000 9000 telnet router2
```

REMOTE BINDIPVIRTUALROUTE

Adds a remote route to the named IP virtual routing table.

To list the remote routes, use the **remote listIProutes** command, page 305. To remove a route from a virtual routing table, use the **remote unbindIPVirtualRoute** command, page 324.

Note: A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **remote restart** or **reboot**.

remote bindIPVirtualRoute <*ipaddr*> <*ipnetmask*> <*hops*> *[*<*ipgateway*>*]* <*tableName*> <*remoteName*>

ipaddr IP address of the remote network or station (4 decimals separated by periods).

ipnetmask IP network mask of the remote network or station (4 decimals separated by periods).

hops Perceived cost in reaching the remote network or station by this route (number between 1 and 15).

ipgateway Address of a router on the remote LAN (4 decimals separated by periods).
Enter a gateway only if you are configuring a MER interface.

tableName IP virtual routing table to which the route is added.

remoteName Name of the remote router (character string).

Example:

The following command adds a route to virtual routing table FRANCISCO. The route is to IP address 10.1.2.0/255.255.255.0 and goes through remote router HQ.

```
remote bindIPVirtualRoute 10.1.2.0 255.255.255.0 1 francisco HQ
```

REMOTE BLOCKNETBIOS

This command turns on or turns off a filter that blocks all NetBIOS packets over this WAN connection.

remote blockNetBIOS on|off <*remoteName*>

REMOTE DEL

Deletes a remote router entry from the remote router database.

remote del <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote del HQ`

REMOTE DELATMNSAP

This command deletes an ATM mapping set by the **remote setATMnsap** command (page 307).

remote delATMNasp ATMF | E164 partial | full <*nsap*> <*remoteName*>

REMOTE DELBRIDGE

Removes the designation of the remote router entry as the default bridging destination. (Default bridging destinations are defined using the **remote addbridge** command, page 290.) To see the bridge settings for a remote entry, use the **remote listbridge** command (page 305).

To remove a designation as the default bridging destination for a specific MAC address, specify that address on the command. The entry is then removed from the bridging table. To see the entries in the bridging table, use the **bi list** command (page 212).

remote delbridge ***** | <*MAC_addr*> <*remoteName*>

***** All MAC addresses.

MAC_addr MAC address (six bytes, specified as six hexadecimals, separated by colons).

remoteName Name of the remote router (character string). The name is case-sensitive.

Example:

```
remote delbridge 01:08:03:0A:0B:0C HQ
```

REMOTE DELENCRYPTION

Deletes encryption files associated with a remote router.

remote delEncryption <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote delEncryption HQ`

REMOTE DELHOSTMAPPING

Undoes an IP address/host translation (remapping) range that was previously established with the command **remote addhostmapping** on a *per-remote-router basis.*

remote delHostMapping <*first private addr*> <*second private addr*> <*first public addr*> <*remoteName*>

first private addr First IP address in the range of IP address, in the format of 4 decimals separated by periods.

second private addr Last address in the range of IP address, in the format of 4 decimals separated by periods.

first public addr Defines the range of public IP addresses, in the format of 4 decimals separated by periods. The rest of the range is computed automatically.

remoteName Name of the remote router (character string).

Example: `remote delHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 HQ`

REMOTE DELIPROUTE

Deletes an IP address route for a network or station on the LAN connected beyond the remote router. The route is deleted from the default routing table.

Note: Changes to the default routing table require a **save** and **remote restart** or **reboot** before they take effect.

remote delIpRoute <*ipaddr*> <*remoteName*>

ipaddr IP address of the remote network or station (4 decimals separated by periods).

remoteName Name of the remote router (character string).

Example: `remote delIpRoute 10.1.2.0 HQ`

REMOTE DELIPXROUTE

Deletes an IPX address for a network on the LAN connected beyond the remote router.

Note: The **reboot** command must be issued on the target router for a deleted static route to take effect.

remote delIpxroute <*ipxNet*> <*remoteName*>

ipxNet IPX network number represented by 8 hexadecimal characters.

remoteName Name of the remote router (character string).

Example: `remote delIpxRoute 010a020b HQ`

REMOTE DELIPXSAP

Deletes an IPX service on the LAN network connected beyond the remote router.

Note: The **reboot** command must be issued on the target router for a deleted service to take effect.

remote delIpxSap <*servicename*> <*remoteName*>

servicename Name of server

remoteName Name of the remote router (character string).

Example: `remote delIpxSap Fileserver HQ`

REMOTE DELOURPASSWD

Removes the unique CHAP or PAP authentication password entries established by the command **remote setOurPasswd.**

remote delOurPasswd <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote delOurPasswd HQ`

REMOTE DELOURSYSNAME

Removes the unique CHAP or PAP authentication system name entries established by the command **remote setOurSysName.**

remote delOurSysName <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote delOurSysName HQ`

REMOTE DELPHONE

Deletes a phone number that was specified by the command **remote setPhone** (page 316).

remote delPhone async | isdn 1 | 2 <*phone#*> <*remoteName*>

async Asynchronous connection

isdn ISDN connection

1 Primary phone number or first ISDN channel

2 Alternative phone number or second ISDN channel.

phone# Decimal number representing the exact digits to be dialed. Digits, the asterisk, and the # characters are accepted; use a comma to specify a 2-second pause.

remoteName Name of the remote entry (character string).

Example:

```
remote delphone async 1 9,3801100 backup
remote delphone async 2 9,3801101 backup
```

REMOTE DELSERVER

Deletes an entry created by the **remote addServer** command (page 293).

remote delServer <*action*> <*protocol*> <*first port*> [<*last port*> [<*first private port*>]]

action One of the following command actions:
- *ipaddr* Selects the host with this IP address as server (4 decimals separated by periods).
- **discard** Discards the incoming server request.
- **me** Sends the incoming server request to the local router, regardless of its IP address.

protocol Protocol used by the selected server.
- *protocolid* Numeric protocol ID.
- **tcp** TCP only.
- **udp** UDP only.
- **all** All protocols.

first port First or only port as seen by the remote end. Port used by the selected server
- *portid* Numeric value between 0 and 65,535. A numeric value of 0 matches any port.
- **ftp** FTP port.
- **h323** H.323 port.
- **http** HTTP port.
- **smtp** SMTP port.
- **sntp** SNTP port.
- **t120** T.120 port
- **telnet** Telnet port.
- **tftp** TFTP port.
- **all** All ports.

last port Optional last port in the range of ports as seen by the remote end for the server on the LAN.

first private port If specified, this is a port remapping of the incoming request from the remote end.

Example:

```
remote delServer 192.168.1.5 tcp ftp router1
```

REMOTE DISABLE

Disables the remote. The remote remains disabled even after a reboot. To enable the remote, you must enter the command **remote enable** (page 299).

Note: You may enter and save information and settings for a disabled remote entry. However, the remote entry cannot be used until it is enabled.

Note: If the remote is currently active when the remote is disabled, the active session is *not* stopped. To stop the active session, use the **remote stop** command (page 324).

remote disable <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote disable HQ`

REMOTE DISAUTHEN

This command is intended for situations where third-party routers cannot be authenticated; the target router will not attempt to authenticate the remote router.

remote disAuthen <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote disAuthen HQ`

REMOTE DISBRIDGE

Disables bridging from the target router to the remote router.

Note: This command requires rebooting the target system for the change to take effect.

remote disBridge <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote disBridge HQ`

REMOTE ENAAUTHEN

With this command the target router will try to negotiate authentication as defined in the remote router's database.

remote enaAuthen <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote enaAuthen HQ`

REMOTE ENABLE

Enables use of an entry in the remote router database. Although the command makes it possible to use the remote entry, it does *not* start an active session for the remote.

Note: The entry remains enabled across reboots. The entry remains enabled until it is disabled by a **remote disable** command (page 298).

remote enable <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote enable HQ`

REMOTE ENABRIDGE

Enables bridging from the target router to the remote router. This command requires rebooting the target system for the change to take effect.

remote enaBridge <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote enaBridge HQ`

REMOTE IPFILTER

This command manages the IP filters on the WAN interface. The filters screen IP packets at the interface level.

You can define filters for any entry in the remote router database. To see the names of the remote entries, use the command **remote list**.

A remote entry can have up to four lists of filters; the list types are Input, Receive, Transmit, and Output. For more information on how these filter types are applied, refer to *IP Filtering, on page 129*.

Note: IP filters take effect immediately upon entry. They can even affect the current connection that you are using to enter commands. Unlike other configuration changes, you do not need to **save** and **restart** or **reboot**.

remote ipfilter <*command*> <*type*> <*action*> <*parameters*> <*remoteName*>

The following commands are provided for managing IP filters for the WAN interface:

remote ipfilter append [<*line number*>] <*type*> <*action*> [<*parameters*>] <*remoteName*>

Appends a filter to the list of filters for this <*type*> (Input, Receive, Transmit, or Output) for this remote entry.

If no line number is specified, the filter is appended to the end of the list; otherwise, it is appended after the specified line. For example, “append 0” appends the filter after line 0. Filters are used in the order they appear in their list.

remote ipfilter insert <*type*> <*action*> <*parameters*> <*remoteName*>

Inserts a filter in the list of filters for this <*type*> (Input, Receive, Transmit, or Output) for this remote entry.

If no line number is specified, the filter is inserted at the beginning of the list; otherwise, it is inserted before the specified line. For example, “insert 0” inserts the filter before line 0 so it is the first filter in the list. Filters are used in the order they appear in their list.

remote ipfilter delete <*type*> <*action*> <*parameters*> <*remoteName*>

Deletes the first filter that matches the filter specified on the command.

remote ipfilter flush [*<first line> [<last line>]*] *<type> <remoteName>*

Deletes a range of filters of this *<type>* (Input, Receive, Transmit, or Output) for this remote entry.

If no line numbers are specified, all filters in the list are deleted. If only the first line number is specified, all filters from that line to the end are deleted. To see the current filter list, use the **remote ipfilter list** command. Filters are used in the order they appear in their list.

remote ipfilter clear [*<first line> [<last line>]*] [*<type>*] *<clear arg> <remoteName>*

Resets the counters for the specified filters. A filter has a counter if the **-c** parameter was specified for the filter.

You can specify the filters whose counters are to be reset by their line number range and type (input, receive, transmit, or output). If no type is specified, the counters for all filters for the interface are reset. If no line numbers are specified, the counters for all filters for that type and interface are reset. If only the first line number is specified, all counters for filters from that line to the end are reset. To see the filter lists and counters, use the **remote ipfilter list** command.

remote ipfilter check *<type> <parameters> <remoteName>*

Checks the action that would be taken if a packet with the specified parameters was compared with the list of filters defined for the specified type and remote entry.

For example, the command

```
remote ipfilter check input -p TCP branch1
```

would check what action (accept, drop, reject, inipsec, outipsec) would be taken for a TCP packet after it was compared with the list of input filters defined for remote entry **branch1**.

remote ipfilter list *<type> <remoteName>*

Lists all filters of the specified *<type>* (input, receive, transmit, or output) for this remote entry.

remote ipfilter watch <on | off> [-q | -v] *<remoteName>*

Turns on or turns off the console watch for this remote router entry. If the watch is on, a message is printed to the console serial port when a packet is dropped or rejected. (The message is also sent to any Syslog servers; see Syslog Client, on page 168.)

However, if the parameter **-q** (quiet) was specified for a filter, no message is printed when that filter matches a packet. If the parameter **-v** (verbose) was specified for a filter, a message is printed whenever that filter matches a packet, regardless of the filter action.

To see the messages, Telnet to the router and enter **system log start**. The watch does not continue after a restart or reboot; to resume the watch, you must enter the **remote ipfilter watch on** command again.

The filter *type* specifies at which point the filter is compared to the IP packet (see the illustration under Filters and Interfaces, on page 129):

input When the packet enters the interface, *before* any network address translation is performed.

receive When the packet enters the interface, *after* any network address translation, but before routing table processing.

transmit After routing table processing, *before* any network address translation before the packet is sent out.

output After routing *and* network address translation, just before the packet is sent out.

If the packet matches the filter, the specified *action* is performed:

accept The packet is allowed to proceed for further processing.

drop The packet is discarded, without sending an ICMP (Internet Control Management Protocol) error message.

reject The packet is discarded and an ICMP error message is returned to the sender.

inipsec The packet is passed to IPSec for decrypting. The filter is intended to match packets coming from the other IPSec gateway. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE, rather than your own filters, to manage your IP Security (see IPSec (Internet Protocol Security), on page 149).

outipsec The packet is passed to IPSec so it can be encrypted and sent to the other IPSec gateway. The filter is intended to match packets coming from the local protected network. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 149).

The following *parameters* specify the characteristics that an IP packet must have in order to match the filter. A filter can require any or all of these characteristics.

-p <*protocol*> | **TCP** | **UDP** | **ICMP**
The packet must have the specified protocol. If no protocol is specified, the filter matches *every* protocol.

-sa <*first source ip addr*>[:<*last source ip addr*>]
The packet must have a source IP address within the specified address range. If only one address is specified, the packet must have that source IP address. If no source IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

-sm <*source ip mask*>
The filter uses the specified mask when comparing the <*first source ip addr*>...<*last source ip addr*> with the source IP address in the IP packet. If no source mask is specified, the mask used is 255.255.255.255.

-sp <*ICMP type*> | <*first source port*>[**:**<*last source port*>]
The packet must have a source port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that source port. If no source port is specified, the filter matches any source port in the range 0**:**0xffff.

-da <*first dest ip addr*>[:<*last dest ip addr*>]
The packet must have a destination IP address within the specified address range. If only one address is specified, the packet must have that destination IP address. If no destination IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

-dm <*dest ip mask*>
The filter uses the specified mask when comparing the <*first dest ip addr*>...<*last dest ip addr*> with the destination IP address in the IP packet. If no destination mask is specified, the mask used is 255.255.255.255.

-dp <*ICMP type*> | <*first dest port*>[**:**<*last dest port*>]
The packet must have a destination port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that destination port. If no destination port is specified, the filter matches any destination port in the range 0**:**0xffff.

-tcp syn|ack|noflag
If the IP packet is a TCP packet, the filter matches the packet only if the packet flag settings are as specified. If no **-tcp** option is specified for the filter, flag settings are not checked.

Note: You may specify *more* than one **-tcp** option for the IP filter.

The **syn**, **ack**, and **noflag** settings work together as follows:
- Specify **-tcp syn** if the TCP SYN flag must be set.
- Specify **-tcp ack** if the TCP ACK flag must be set.
- Specify **-tcp noflag** if neither the SYN flag nor the ACK flag can be set.

For example, for the IP filter to match the initiation of a TCP connection, specify **-tcp syn**. The filter will match TCP packets that have the TCP SYN flag set but *not* the TCP ACK flag set. For the filter to match the response to initiation of a TCP connection, specify **-tcp syn** *and* **-tcp ack**. The filter will match only TCP packets with *both* the TCP SYN and TCP ACK flags set.

The **-tcp rst** setting is independent of the others; if you specify **-tcp rst** for the filter, the filter matches every TCP packet with the TCP RESET flag set, regardless of the other flag settings. For example, for the filter to match packets for "established" connections, you would specify both **-tcp rst** and **-tcp ack** so that the filter is applied to every TCP packet that has either the RESET flag or the ACK flag set.

The following *parameters* request additional filter options.

-b
This option requests that this filter be compared *twice* with each packet.The first time the source filter information is matched against the source information in the IP packet and the destination filter information is matched against the destination information in the IP packet. The second time the source filter information is matched against the destination information in the IP packet and the destination filter information is matched against the source information in the IP packet.

-c <*count of times rule used*>
This option requests a counter for this filter. If specified, a count is kept of how many IP packets have matched this filter since the router was restarted or rebooted.To see the current count for a filter, use the **remote ipfilter list** command. To clear a counter, use the **remote ipfilter clear** command.

-ipsec <*IPSec record name*>

Use this option when the *action* specified is **inipsec** or **outipsec**. It specifies the IPSec Security Association that uses the filter.

-q or **-v**

Specify one of these options to determine when watch messages are sent for this filter. The messages are sent to the console serial port (and to any Syslog servers; see page 168).

If neither **-q** or **-v** are specified for the filter, and an **remote ipfilter watch on** command is entered for the interface, a message is sent each time this filter causes a packet to be dropped or rejected.

If **-q** (quiet) is specified, no messages are printed for this filter, even if it causes a packet to be dropped or rejected.

If **-v** (verbose) is specified, a message is printed every time this filter matches a packet, regardless of the filter action.

The *remote name* specifies the entry in the remote router database that the command applies to. To see the remote names, use the command **remote list**.

Examples:

This command deletes all IP filters of type Receive for the remote interface **internet**.

```
remote ipfilter flush receive internet
```

The following two commands have the same effect: they deny all IP traffic for the remote interface **internet** from the specified destination addresses. The addresses can be specified as 192.168.0.0 masked with 255.255.0.0 or as the range 192.168.0.0 through 192.168.255.255.

```
remote ipfilter append receive drop -da 192.168.0.0 -dm 255.255.0.0 internet
```

```
remote ipfilter append receive drop -da 192.168.0.0:192.168.255.255 internet
```

This command lists all IP filters of type Input for the remote interface **internet**.

```
remote ipfilter list input internet
```

REMOTE LIST

Lists the remote router entry (or all the entries) in the remote router database. The result is a complete display of the current configuration settings for the remote router(s), except for the authentication password/secret.

remote list [<*remoteName*>]

remoteName Name of the remote router to be listed (character string).If you omit the name, all remote router entries are listed.

Example:

```
# rem list internet
INFORMATION FOR <internet>
  Status.............................. enabled
  Our System Name when dialing out.....
  Our Password used when dialing out... no
  Protocol in use..................... PPP
  ATM traffic shaping................. no
  Authentication...................... disabled
  Authentication level required........ PAP
  Use periodic LCP pings.............. yes
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation.............. off
  IP filters defined.................. no
  Send/Receive Multicast.............. off
  Block NetBIOS Packets............... off
  Compression Negotiation............. off
  IP slave mode (PPP)................. no
  Try to reacquire IP addr (PPP)...... yes
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes.............. 1
           10.0.0.0/255.255.0.0/1
  IPX network number.................. 00000000
  Use IPX RIP/SAP (negotiate with PPP): yes
```

```
Total IPX remote routes.............. 0
Total IPX SAPs....................... 0
Bridging enabled..................... no
  Exchange spanning tree with dest... no
TX Encryption........................ unknown
RX Encryption........................ unknown
mtu.................................. 1500
```

REMOTE LISTBRIDGE

Lists the current bridge settings for the specified remote router entry.

remote listBridge <*remoteName*>

remoteName Name of the remote router (character string). If a name is omitted, the bridge settings for all remote router entries are listed.

Example:

```
# remote listbridge
BRIDGING INFORMATION FOR <internet>
 Bridging enabled..................... yes
  Exchange spanning tree with dest... no
  Bridge only PPPoE with dest........ no

BRIDGING INFORMATION FOR <efficient>
 Bridging enabled..................... no
  Exchange spanning tree with dest... no
  Bridge only PPPoE with dest........ no
  Mac addresses that dial remote..... none
```

REMOTE LISTIPROUTES

Lists IP information for a remote router or, if the router name is omitted, for all routers in the remote router database. The IP information includes all network or station IP addresses defined for the LAN connected beyond the remote router.

This command lists all routes defined for the remote router, including those defined in the default routing table and in any virtual routing tables.

remote listIpRoutes [*remoteName*]

remoteName Name of the remote router (character string).

Example: `remote listIproutes HQ`

Example:

The following command lists routing information for remote router HQ. It lists five routes that use HQ, the first four are in the default routing table and the fifth is in virtual routing table FRANCISCO.

```
# remote listIproutes HQ
IP INFORMATION FOR <HQ>
```

```
Send IP RIP to this dest.............. no
  Send IP default route if known..... no
Receive IP RIP from this dest........ rip-1 compatible
  Receive IP default route by RIP.... no
Keep this IP destination private..... yes
Total IP remote routes............... 5
         192.168.210.0/255.255.255.0/1
         10.0.0.0/255.0.0.0/1
         172.16.0.0/255.240.0.0/1
         192.168.0.0/255.255.0.0/1
         10.1.2.0/255.255.255.0/1 through <FRANCISCO>
```

REMOTE LISTIPXROUTES

Lists all network IPX route addresses defined for the LAN connected beyond the remote router. The network number, hop count, and ticks are displayed. If the remote name is not specified, a list of IPX routes is displayed for each remote router in the database.

remote listIpxroutes [*remoteName*]

remoteName Name of the remote router (character string).

Example: `remote listIpxroutes HQ`

Response:

```
IPX ROUTE INFORMATION FOR <HQ>
Total IPX remote routes ..............1     00001001/1/4
```

REMOTE LISTIPXSAPS

Lists all services defined for the LAN connected beyond the remote router. Each service includes the server name, network number, node number, socket number, server type, and hop count. If the remote name is not specified, a list of IPX SAPs is displayed for each remote router in the database.

remote listIpxsaps [*remoteName*]

remoteName Name of the remote router (character string.)

Example: `remote listIpxsaps HQ`

Response:

```
IPX SAP INFORMATION FOR <HQ>
Total IPX SAPs ...................  1
SERV312_FP 00001001 00:00:00:00:00:01 0451 0004 1

IPX SAP INFORMATION FOR <ISP>
Total IPX SAPs ...................  0
   SERV312_FP 00001001 00:00:00:00:00:01 0451 0004 1
```

REMOTE LISTPHONES

Lists the PVC numbers available for connecting to the remote router.

remote listPhones <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote listPhones HQ`

Response:

```
PHONE NUMBER(s) FOR <HQ>
Connection Identifier (VPI*VCI)...... 0*38
```

Note: If the remote name is not specified, a list of phone numbers is displayed for each remote router in the database.

REMOTE RESTART

Stops the current active session and starts a new active session for a remote.

Certain configuration changes for a remote become effective only after the remote is restarted or the router is rebooted. Remember to **save** the changes before the restart or reboot.

Note: Use **restart** instead of **reboot** whenever possible. A restart does not affect other interfaces, allowing their traffic to continue. For example, using restart, you can add an IP route without killing voice traffic.

To restart an Ethernet interface, use **eth restart** (page 281).

remote restart <*remoteName*>

remoteName Remote interface name.

Example:

The following command restarts the active session for remote HQ.

```
remote restart HQ
```

REMOTE SETATMNSAP

RFC1577 (Classical IP over ATM) specifies a mechanism to map an ATM Name (called an NSAP) to a PVC. NSAP's are normally not needed, but if they are used, they have a syntax defined by using either the ATM or E164 encodings. By convention, octets 2-7 contain a unique identifier for the router, such as a MAC address.

In the command **remote setATMnsap**, the complete 20 octets of the NSAP are specified. If Partial mode is selected, the router substitutes the MAC address of the router for octets 2-7. In Full mode, no change is made to the NSAP.

To see an ATM NSAP that has been set, use the **remote list** command.

remote setATMNSAP ATMF|E164 partial | full <*nsap*> <*remoteName*>

ATMF	ATM forum encoding
E164	ITU E164 encoding.
partial	The MAC address of the router is substituted for octets 2-7 of the NSAP.
full	No change is made to the specified NSAP.
<*nsap*>	NSAP specified as 40 hex digits or 20 octets (2-digit pairs separated by colons).
<*remoteName*>	Name of the remote router (character string).

Example:

```
# rem setatmnsap atfm partial 11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:10:11:12:13 co

# remote list

      INFORMATION FOR <co>
        Status..................................... enabled
        Our Password used when dialing out... no
        Protocol in use........................... PPP
        ATM NSAP.............................. ATM Forum / Partial
            value  11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:10:11:12:13
```

REMOTE SETAUTHEN

Sets the authentication protocol used communicate with the remote router. The authentication protocol is the *minimum* security level that the target router must use with the remote router; this level is verified during security negotiation. The router will *always* attempt to negotiate the highest level of security possible (CHAP). The router will not accept a negotiated security level less than this minimum authentication method.

The parameter in the remote router database is used for the local side of the authentication process; this is the minimum security level used by the target router when it challenges or authenticates the remote router.

remote setAuthen <*protocol*> <*remoteName*>

protocol	**c**hap, pap, or none. The default is pap.
remoteName	Name of the remote router (character string).
Example:	remote setAuthen pap HQ

REMOTE SETBOD

Sets the bandwidth on demand (BOD) management option for a DOD (dial on demand) connection, that is, a connection where the link goes up and down. These links include those for ISDN, L2TP tunnels, IPSec tunnels, and dial backup.

The bandwidth on demand management option can be set to apply to incoming, outgoing, or both incoming and outgoing traffic. The bandwidth threshold set by the **remote setBWthresh** command (page 309) applies to the direction of traffic set by this command.

remote setBOD in | out | both <*remoteName*>

in | out | both Incoming traffic, outgoing traffic, or both. The default is both.

remoteName Name of the remote router (character string).

Example:

```
remote setBOD out HQ
```

REMOTE SETBROPTIONS

Sets controls on bridging for the remote router entry.

To see the current bridging settings for remote router entries, use the **remote listbridge** command (page 305).

Warning: Do not change the **stp** setting without approval from your system administrator.

remote setBrOptions <*option*> **on** | **off** <*remoteName*>

option **stp**

Set this option to **on** to use the Spanning Tree Protocol (STP). The default is **on**.

STP is used to detect bridging loops. Set this option to **off** only if the bridging peers do not support the Spanning Tree Protocol or if you are certain that no bridging loops could exist. When STP is disabled on an interface, any STP packets received on that interface are ignored.

Note: The Spanning Tree Protocol adds a 40-second delay each time the ADSL or ATM link comes up while the interface determines if there is a bridging loop.

pppoeOnly

Set this option to **on** to limit this remote router entry to bridging PPPoE traffic only. If the option is set to **off**, then the entry can bridge any traffic, including PPPoE traffic. The default is **off**.

remoteName Name of the remote router (character string).

Examples:

The following command requests the spanning tree protocol for remote router HQ.

```
remote setBrOptions stp on HQ
```

The following command configures remote router PPPoEbridge as the remote through which only PPPoE traffic is bridged.

```
remote setBrOptions pppoeonly on PPPoEbridge
```

REMOTE SETBWTHRESH

Sets the bandwidth threshold for a DOD (dial on demand) connection, that is, a connection where the link goes up and down. These links include those for ISDN, L2TP tunnels, IPSec tunnels, and dial backup.

The threshold is used in bandwidth on demand management. Initially, a call is activated on one B-channel. When bandwidth utilization reaches the bandwidth threshold, the second B-channel is activated. (The additional channel is available if the maximum links was set to 2 by a **remote setmaxline** command, page 313.)

Both channel are utilized until the bandwidth utilization drops below the threshold. The default is 0% utilization, in which case, both channels are always used for data transmission.

If you wish, you can have the bandwidth threshold apply only to incoming or outgoing traffic; see the **remote setBOD** command (page 308).

remote setBWthresh *<threshold> <remoteName>*

threshold Percentage of bandwidth utilization (0 through 100). The default is 0, in which case, whenever data transmission occurs, the maximum number of links is allocated.

remoteName Name of the remote router (character string).

Example:

```
remote setBWthresh 75 HQ
```

REMOTE SETCOMPRESSION

Enables or disables negotiation of the Stac LZS compression of the payload (RFC 1974). The CCP (Compression Control Protocol, RFC 1962) negotiates and handles any compression between the local router and the remote router.

The default setting is **off** because LZS compression has a negative effect with high bit rates (greater than 768 Kb/s).

To see the current setting for payload compression, enter **remote list** and check the *Compression Negotiation*line. If desired, you can follow the negotiation of the Stac LZS compression within CCP using the debug command **mlp debug ccp**.

remote setCompression on | off *<remoteName>*

on Enables compression negotiation between the local and the remote router *if* both routers are set to perform compression and *if* they both share a common compression protocol.

off Disables compression negotiation. The default is **off**.

remoteName Name of the remote router (character string).

Example:

```
remote setCompression on HQ
```

REMOTE SETENCRYPTION (RFC 1969 Encryption)

This command is used to specify a PPP DES (Data Encryption Standard) 56-bit key with fixed transmit and receive keys.

remote setEncryption DESE RX|TX <*key*> <*remoteName*>

RX — Receive key

TX — Transmit key

key — Key in the format of an eight-hexadecimal number.

remoteName — Name of the remote router (character string).

Example:

```
remote setEncryption dese tx 1111111111111111 HQ
remote setEncryption dese rx 2222222222222222 HQ
```

REMOTE SETENCRYPTION (Diffie-Hellman Encryption)

This command is used to specify encryption based on the Diffie-Hellman key-exchange protocol. Each router possesses an internal encryption file that is associated with a public key providing 768-bit security. The predefined keys can be replaced by the user. The configuration file on the router must have a “num” suffix (e.g., dh96.num).

remote setEncryption DESE_1_KEY|DESE_2_KEY [<*filename*>] <*remoteName*>

DESE_1_KEY — Specifies that the same key is used in both directions

DESE_2_KEY — Specifies that the keys are different

filename — Name of the file containing the Diffie-Hellman values. If the file is not specified, default values built into the router’s kernel are automatically selected.

remoteName — Name of the remote router (character string).

Example:

```
remote setEncryption DESE_1_KEY dh96.num HQ
```

REMOTE SETIPOPTIONS

Turns on or turns off the selected IP option for the WAN interface. To select IP options for the Ethernet interface, use the command **eth ip options** (page 276).

Several RIP options are available. RIP is a protocol used for exchanging IP routing information among routers. The RIP options allow you to set IP routing information protocol controls over a point-to-point WAN. For more information, see RIP Controls, on page 83.

remote setipoptions <*option*> on|off <*remoteName*>

option — Specify one of the following options:

rxrip — Receive and process IP RIP-1 compatible packets and RIP-2 broadcast packets from the remote site. Also receive and process RIP-2 multicast packets. Set this option if the local router is to discover route information from other sites connected to the remote router. This is useful for hierarchical organizations. If you are connecting to another company or an Internet Service Provider, you may wish to set this option off. The default is off.

rxrip1 Receive and process RIP-1 packets only.

rxrip2 Receive and process RIP-2 packets only.

rxdef Receive default IP route address. When this option is set on, the local router receives the remote site's default IP route. The default is off**.**

txrip Transmit IP RIP-1 compatible broadcast packets and RIP-2 multicast packets to the remote site. When this option is set on, the local router sends routing information packets to the remote site. The default is off.

txrip1 Transmit broadcast RIP-1 packets only.

txrip2 Transmit multicast RIP-2 packets only.

txdef Transmit the local router's default IP route. When this option is set to on, the local router sends the default route to the remote site. The default is off**.**

private Keep IP routes private. Used to prevent advertisement of this route to other sites by the remote router. Used as a security mechanism when the remote site is outside your company (an Internet Service Provider, for example), or whenever you want to keep the identity of the site private. The default is on**.**

multicast Allows the remote router to forward IP multicast traffic.

LANconfig Accept LAN configuration information. Indicates that this PPP remote can receive IPCP information for dynamically reconfiguring the Ethernet interface.

lcpecho Use periodic echo.

remoteName Name of the remote router (character string).

Example:

```
remote setipoptions private on HQ
```

REMOTE SETIPSLAVEPPP

Sets the IP Slave PPP mode. If the slave mode is yes, the router accepts the IP address that the remote end informs the router that it has; the router disregards any IP address specified in its own configuration. If the mode is no, the router tries to use the address in its configuration.

Normally there is no need to change the default (no) value of this option. However, in certain situations where the router is managed by another party, (as part of a managed service), you could set this value to yes to ensure that the central management site always specifies the IP address of the router.

remote setIPSlavePPP yes | no <*remoteName*>

yes | no Slave mode setting. The default is no.

remoteName Name of the remote router (character string).

Example:

```
remote setipslaveppp yes branchremote
```

REMOTE SETIPTRANSLATE

This command is used to control Network Address Translation on a per remote router basis. It allows several PCs to share a single IP address to the Internet. The remote router must assign the source WAN IP address to the routers' local WAN port. This command requires that you define a Source WAN IP Address with the command: **remote setSrcIpAddr**

remote setIPTranslate on|off <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote setIPTranslate on HQ`

REMOTE SETIPXADDR

Sets the IPX network number for the remote WAN connection. For more information about IPX configuration, see IPX Routing Concepts, on page 408.

remote setIpxaddr <*ipxNet*> <*remote*>

ipxNet IPX network number represented by 8 hexadecimal characters.

remoteName Name of the remote router (character string).

Example:

```
remote setIpxaddr 789 HQ
```

REMOTE SETIPXOPTIONS

Turns on or turns off the IPX option RIPSAP for the remote WAN connection.

remote setIpxOptions ripsap on | off <*remoteName*>

on | off Sets the option on or off.

remoteName Name of the remote router (character string).

Example:

```
remote setIpxOptions ripsap on HQ
```

REMOTE SETMAXLINE

Sets the maximum links (1 or 2) for a DOD (dial on demand) connection, that is, a connection where the link goes up and down. These links include those for ISDN, L2TP tunnels, IPSec tunnels, and dial backup.

If you set the maximum links to 2, bandwidth on demand management determines their actual usage; see the **remote setBWthresh** command (page 309).

remote setMaxLine 1 | 2 <*remoteName*>

1 | 2 Maximum number of links to be used for the connection (1 or 2). The default is 1.

remoteName Name of the remote router (character string).

Example:

```
remote setMaxLine 2 HQ
```

REMOTE SETMGMTIPADDR

This command assigns to the remote router entry an IP address which is to be used for management purposes only and not for IP address translation. This management IP address is generally a private network address used solely by the ISP.

The management IP address is separate from the IP address used for IP address translation. The IP address used for address translation is generally a public IP address valid on the Internet. It is set by the **remote setSrcIpAddr** command (page 322).

Note: The management address is not effective until after the next **save** and **remote restart** or **reboot**.

Note: To use the management address as the source address for a ping, you must specify it using the **-I** option on the **ping** command (page 217). For example, to use management address 192.168.1.2 when pinging destination address 192.168.100.100, specify:

```
ping -I 192.168.1.2 192.168.100.100
```

Note: To use the management address as the source address for a copy, you must specify both the source and destination addresses on the **copy** command (page 226).

To list the current management address for the remote router, if any, use the **remote list** command (page 304). To set a management address for an Ethernet interface, see **eth ip mgmt** (page 275).

remote setMgmtIpAddr <*ipaddr*> <*mask*> <*remoteName*>

ipaddr IP address (4 decimals separated by periods).

mask IP subnet mask (4 decimals separated by periods).

remoteName Name of the remote router (character string).

Example:

```
remote setMgmtIpAddr 10.0.0.3 255.255.255.0 internet
```

REMOTE SETMINLINE

This command is used for dial-up connections and other connections that behave like dial-up connections, such as L2TP and PPPoE sessions. The command sets the minimum number of channels to be continually allocated to the connection. The default is 0, in which case a channel is allocated only when needed.

For example, if your service provider charges by the hour, you might prefer the minlines default value (0) so that a channel is allocated only when needed. However, if you are not charged by the hour, then having a channel allocated continually would save you the 2-3 second wait time required for each channel re-allocation.

remote setMinLine <*minlines*> <*remoteName*>

minlines Minimum number of channels to be continually allocated for the connection (0, 1, or 2). If you specify 0, a channel is allocated for the connection only when needed. The default is 0.

remoteName Name of the remote router (character string).

Examples:

The following command keeps a channel allocated for the session even when there is no traffic.

```
remote setMinLine 1 PPPoEuser
```

The following commands set up a timeout period so that, if there is no traffic for 10 minutes (600 seconds), the channel is deallocated.

```
remote setMinLine 0 PPPoEuser
remote settimer 600 PPPoEuser
```

REMOTE SETMTU

Sets the maximum transfer unit for the remote interface.

To see the current MTU size for an active remote that is doing IP routing, use the **ipifs** command (page 215). To change the MTU for an Ethernet interface, use the command **eth mtu** (page 281).

If the protocol in use is PPP, you can see the MRU and MTU sizes using the command **mlp show**. The MRU is the maximum receive unit. Other information in the **mlp show** output includes the *maxtu* (the maximum packet size that can be sent; it is based on the peer's MRU size), the *ourmru* (the maximum PPP packet size that can be received if multilink *is not* running), and *ourmrru* (the maximum PPP packet size that can be received if multilink *is* running).

remote setmtu <*size*> <*remoteName*>

size Maximum number of bytes that can be transferred as a unit.

remoteName Name of the remote router (character string).

Example:

The following command decreases the MTU size for remote interface HQ to 1400 bytes.

```
remote setmtu 1400 HQ
```

REMOTE SETOURPASSWD

Sets a unique CHAP or PAP authentication password for the local router that is used for authentication when the local router connects to the specified remote router. This password overrides the password set in the **system passwd** command. A common use is to set a password assigned to you by Internet Service Providers.

remote setOurPasswd <*password*> <*remoteName*>

password Authentication password of the local router for use in connecting to the remote router.

Note: The password is case-sensitive and its maximum length is 39 characters.

remoteName Name of the remote router (character string).

Example: `remote setOurPasswd s1dpxl7 HQ`

REMOTE SETOURSYSNAME

Sets a unique CHAP or PAP authentication system name for the local router that is used for authentication when the local router connects to the specified remote router. This system name overrides the system name set in the **system name** command. A common use is to set a password assigned to you by Internet Service Providers.

remote setOurSysName <*name*> <*remoteName*>

name System name of the target router.

Note: The system name is case-sensitive and may be no more than 255 characters.

remoteName Name of the remote router (character string).

Example: `remote setOurSysName s1dpxl7 HQ`

REMOTE SETPASSWD

Sets the CHAP or PAP authentication password that is used when the remote router establishes a connection or is challenged by the target router.

remote setPasswd <*password*> <*remoteName*>

password Authentication password of the remote router. Note that the password is case-sensitive and its maximum length is 40 characters.

remoteName Name of the remote router (character string).

Example: `remote setPasswd s2dpxl7 HQ`

REMOTE SETPHONE

Specifies the phone number to be used for the dial on demand (DOD) connection, that is, a connection where the link goes up and down. These links include those for ISDN, L2TP tunnels, IPSec tunnels, and dial backup.

For dial backup, the phone number is used when dialing out using the backup V.90 modem connected to the console port. You may specify both a primary number and an alternative phone number. For more information on the Dial Backup option, see page 164.

remote setPhone async | isdn 1 | 2 *<phone#> <remoteName>*

async — Asynchronous connection

isdn — ISDN connection

1 — Primary phone number or first ISDN channel

2 — Alternative phone number or second ISDN channel.

phone# — Decimal number representing the exact digits to be dialed. Digits, the asterisk, and the # characters are accepted; use a comma to specify a 2-second pause.

remoteName — Name of the remote entry (character string).

Example:

The following is an example of phone numbers and bit rates for an asynchronous interface used for Dial Backup.

```
# The phone number begins with 9 (to get an outside line), a comma (for a 2-second
# pause), and finally the 7-digit local number.
remote setphone async 1 9,3801100 backup
remote setspeed 115200 async 1 backup
# Specifies the alternative phone number to be used and its bit rate.
remote setphone async 2 9,3801101 backup
remote setspeed 115200 async 2 backup
```

The following is an example of a command specifying two ISDN phone numbers, 555-2000 and 555-4000.

```
remote setphone async 1 5552000&5554000 backup
```

REMOTE SETPPPOPTIONS

Turns on or turns off a PPP option.

The default settings vary with the option. To see the current settings of the PPP options, use the command **remote list** (page 304).

remote setPPPoptions *<option>* on | off *<remoteName>*

option — Option to be turned on or turned off.

- **compression** — Van Jacobson compression of TCP/IP headers (RFC 1144), also known as IPCP compression.
- **ipSlaveMode** — Always accept peer proposal for our WAN IP address.
- **lcpecho** — Use periodic echo (if permanent interface or PPPoE).

reacqIPAddr Try to reacquire the IP address. Turn this option off if the router should always request a new IP address when the PPP session is terminated.

RIPSAP Use IPX RIP/SAP protocols.

on | off Desired setting for the option.

remoteName Name of the remote entry (character string).

Example:

```
remote setPPPopt compression on HQ
```

The following command forces the router to always request a new IP address whenever the PPP session is terminated. (This could be useful if the other PPP system does not completely support IP address negotiation.)

```
remote setPPPoptions reacqIPAddr off HQ
```

REMOTE SETPPPRETRYTIMER

Turns on or turns off the PPP retry timer for a remote. The default is off (**0**).

The PPP retry timer is useful in a network where several routers are connected to the same PPP server. If the link to the PPP server goes down, all PPP sessions on the connected routers go down. Then, when the link comes back up, all routers attempt reconnection at the same time and this could crash the PPP server. To solve this problem, turn on the PPP retry timer for each remote. Then, when the link comes back up, each router waits a random time before attempting reconnection.

To see the current setting of the retry timer for a remote, use the **remote list** command and check the output line:

```
Retry Timer (PPP) ..................... 0
```

Note: A change is effective immediately; **save** the change if it is to persist across restarts and reboots.

remote setPPPRetryTimer *<timerValue> <remoteName>*

timerValue Timer setting. To turn off the retry timer, specify **0**; this is the default value.

To turn on the retry timer, specify a value from **1** to **60**. The value is the maximum number of seconds before the router attempts reconnection. For example, if you enter 60, the router waits from 1 to 60 seconds before attempting reconnection.

remoteName Name of the remote entry (character string).

Example:

```
remote setPPPRetryTimer 60 HQ
```

REMOTE SETPREFER

Changes the interface for the remote entry. Normally, a new remote profile defaults to the type of the WAN port present in the router: FR for Frame-Relay WANs (IDSL and some SDSL routers) or HSD for all ATM routers.

Use this command when defining the remote profile for Dial Backup. Dial Backup uses the console port as a serial port connected to an asynchronous modem; its interface must be asynchronous (see Specifying the Dialup Parameters, on page 110).

To see the current setting for a remote profile, use the **remote list** command and check the `Interface in use` line. Changing the interface preference changes the lines presented in the display; phone numbers are displayed only for asynchronous. See the example below.

remote setPrefer < async | fr | hsd > <*remoteName*>

async Asynchronous. This preference allows you to specify phone numbers and bit rates in the remote profile.

fr Frame Relay.

hsd High-Speed Data. Use this option for ATM virtual circuits; in this case, phone numbers take the form <VPI>*<VCI>.

remoteName Name of the remote router (character string).

Example:

The information displayed by a **remote list** command changes depending on the interface preference. The following example shows how the information displayed changes from asynchronous to frame relay:

```
# remote setprefer fr backup
# remote list backup
INFORMATION FOR <backup>
  Status.............................. enabled
  Our System Name when dialing out..... gwbush
  Our Password used when dialing out... yes
  Interface in use..................... FR
  Protocol in use...................... PPP
  Authentication....................... disabled
  Authentication level required........ PAP
  Use periodic LCP pings............... yes
     . . . (subsequent lines same as for async) . . .
# remote setPrefer async backup
# remote list backup
INFORMATION FOR <backup>
  Status.............................. enabled
  Our System Name when dialing out..... gwbush
  Our Password used when dialing out... yes
  Disconnect timeout (in seconds)...... 60
  Min/max channels..................... 0/1
  Interface in use..................... ASYNC
  Protocol in use...................... PPP
  Authentication....................... disabled
  Authentication level required........ PAP
  Bandwidth management criteria........ both
  Use periodic LCP pings............... yes
  1. ASYNC telephone number, speed 115200 9,5554218
  2. ASYNC telephone number, speed 115200 9,5554219
  1. HSD telephone number, speed auto
  2. HSD telephone number, speed auto
  Dial Back............................off
  Request PPP Call Back................no
```

. . . (subsequent lines same as for fr) . . .

REMOTE SETPROTOCOL

Sets the link protocol for the remote router.

Note: The link protocol and encapsulation option must match those at the other end of the connection (the settings in the DSLAM).

The encapsulation options are described in Encapsulation Options, on page 29. Protocol configuration is described in Protocols to be Used, on page 35.

remote setProtocol PPP | PPPLLC | RFC1483 | RFC1483MER | FRF8 | RAWIP <*remoteName*>

PPP	PPP protocol with VC multiplexing encapsulation.
PPLLC	PPP protocol with LLC SNAP encapsulation (used with frame relay internetworking units).
RFC1483	RFC 1483 protocol.
RFC1483MER	RFC 1483MER (MAC Encapsulated Routing) protocol.
FRF8	This protocol implements ATM to frame relay as defined in the Frame Relay Forum FRF.8 Interworking Agreement.
RAWIP	RawIP protocol.
remoteName	Name of the remote router (character string).

Example:

```
remote setProtocol ppp fp1
```

REMOTE SETPVC

Specifies the PVC number for connecting to the remote router.

remote setPVC <*vpi number*>*<*vci number*> <*remoteName*>

vpi number	Virtual Path ID — number that identifies the link formed by the virtual path.
vci number	Virtual Circuit ID — number that identifies a channel within a virtual path in a DSL/ATM environment.
remoteName	Name of the remote router (character string).

Example: `remote setPVC 0*38 HQ`

REMOTE SETRMTIPADDR

Sets the WAN IP address for the remote router. This address is required only if the remote router does not support IP address negotiation under PPP (i.e., numbered mode is required, and the remote router cannot specify a WAN IP address for use during the negotiation process).

remote setRmtIpAddr <*ipaddr*> <*mask*> <*remoteName*>

ipaddr	IP address of the remote router, in the format of 4 decimals separated by periods.
mask	IP network mask of the remote router, in the format of 4 decimals separated by periods.
remoteName	Name of the remote router (character string).
Example:	`remote setRmtIpAddr 128.1.210.65 255.255.255.192 HQ`

REMOTE SETSPEED

Specifies the speed to be used when dialing out using the backup V.90 modem connected to the console port. Specify a speed for each phone number you provide (primary and alternative).

For more information specifying phone numbers for the Dial Backup feature, see Specifying the Dialup Parameters, on page 110.

remote setSpeed <*bitrate*> | default **async** 1 | 2 <*remoteName*>

bitrate	Bit rate to be used for the phone number. Possible speeds are 38400, 57600, 115200, or 230400.
default	Use the default speed.
1	Primary phone number.
2	Alternative phone number.
remoteName	Name of the remote entry (character string).

Example:

```
# Specifies the primary phone number and its bit rate.
remote setphone async 1 9,5551288 backup
remote setspeed 115200 async 1 backup
# Specifies the alternative phone number to be used and its bit rate.
remote setphone async 2 9,5551289 backup
remote setspeed 115200 async 2 backup
```

REMOTE SETSRCIPADDR

Sets the IP address for the target WAN connection to the remote router. You may set this address when the remote router requires the target and the remote WAN IP addresses to be on the same subnetwork. Another instance is to force numbered mode and to prevent the remote router from changing the target WAN IP address through IPCP address negotiation. The target WAN IP address defaults to the Ethernet LAN IP address.

remote setSrcIpAddr <*ipaddr*> <*mask*> <*remoteName*>

ipaddr Target IP address of the WAN connection to the remote router, in the format of 4 decimals separated by periods.

mask IP network mask, in the format of 4 decimals separated by periods.

remoteName Name of the remote router (character string).

Example: `remote setSrcIpAddr 128.1.210.151 255.255.255.192 HQ`

REMOTE SETTIMER

This command is used for dial-up connections and other connections that behave like dial-up connections, such as L2TP and PPPoE sessions. The command sets the length of the timeout period before disconnection.

When the connection has had no traffic for the timeout period, the channel is deallocated. A channel is re-allocated when it is needed.

A timeout period is desirable if your service provider charges by the hour. However, the connection has to wait a few seconds each time a channel is re-allocated.

Note: The timeout period set by this command is not effective if a **remote setMinLines** command has changed the minlines value from its default (0) to 1 or 2.

remote setTimer <*seconds*> <*remoteName*>

seconds Number of seconds in the timeout period. The default is 60.

remoteName Name of the remote router (character string).

Example:

The following commands set up a timeout period so that, if there is no traffic for 10 minutes (600 seconds), the channel is deallocated.

```
remote setMinLine 0 PPPoEuser
remote settimer 600 PPPoEuser
```

REMOTE START

If the remote is not currently active, this command attempts to start an active session.

Note: A reboot ends the active session; to start a session after the reboot, you must enter another **remote start** command.

To stop an active session for the remote, use the command **remote stop** (page 324). To stop and immediately restart a session for the remote, use the command **remote restart** (page 307).

remote start <*remoteName*>

remoteName Remote interface name.

Example:

The following command starts remote interface HQ.

```
remote start HQ
```

REMOTE STATS

Shows the current status of the connection to the remote router, including the bandwidth and data transfer rate.

remote stats [<*remoteName*>]

remoteName Name of the remote router (character string).

Example: `remote stats HQ`

Response:

```
STATISTICS FOR <HQ>:
  Current state ......................  currently connected
  Current output bandwidth ............. 0 bps
  Current input bandwidth .............. 0 bps
  Current bandwidth allocated .......... 25600000 bps
  On port ATM_VC/1 ..................... 0+01:02:36 (0%/0% of 25600000 bps)
  Total connect time ................... 0+01:11:48
  Total bytes out ...................... 15896
  Total bytes in ....................... 0

STATISTICS FOR <internet>:
  Current state ......................  not connected
  Current output bandwidth ............. 0 bps
  Current input bandwidth\ ............. 0 bps
  Current bandwidth allocated .......... 0 bps
  Total connect time ................... 0+00:00:00
  Total bytes out ...................... 0
  Total bytes in ....................... 0
```

where:

Current state: connected, not connected, currently connecting, currently attempting to connect, currently closing, out of service, or not known.

Bandwidth state: idle, increasing, decreasing, decreasing hold, unknown, or idle.

REMOTE STATSCLEAR

Allows the user to reset the statistics counter for a given remote router.

remote statsclear <*remoteName*>

remoteName Name of the remote router (character string).

Example: `remote statsclear HQ`

REMOTE STOP

If the remote is active, this command stops the active session.

Note: To keep certain configuration changes, you must enter a **save** command before stopping the remote interface.

The stop command does not disable the remote entry so another session can be started for the remote.To start an active session for the remote, use the command **remote start** (page 322). To stop and immediately restart a session for a remote, use the command **remote restart** (page 307).

remote stop <*remoteName*>

remoteName Remote interface name.

Example:

The following command stops the active session for remote HQ.

```
remote stop HQ
```

REMOTE UNBINDIPVIRTUALROUTE

Removes a remote route from the named IP virtual routing table.

To list the remote routes, use the **remote listIProutes** command, page 305. To add a remote route, use the **remote bindIPVirtualRoute** command, page 293.

Note: A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **remote restart** or **reboot**.

remote unbindIPVirtualRoute <*ipaddr*> <*tablename*> <*remoteName*>

ipaddr IP address of the remote network or station (4 decimals separated by periods).

tablename IP virtual routing table from which the route is removed (character string).

remoteName Name of the remote router (character string).

Example:

The following command removes a route from virtual routing table FRANCISCO. The route removed is for IP address 10.1.2.0 and remote router HQ.

```
remote unbindIPVirtualRoute 10.1.2.0 FRANCISCO HQ
```

WAN Interface Commands

This section contains subsections of commands applicable to specific WAN interfaces. The subsections are:

adsl ADSL (Asymmetric Digital Subscriber Line) commands See page 326.

atm ATM (Asynchronous Transfer Mode) commandsSee page 328.

dmt DMT (Discrete Multi-Tone) commandsSee page 331.

eth Dual-Ethernet commandsSee page 332.

frame Frame Relay commands See page 334.

hdsl HDSL (High-speed Digital Subscriber Line) commandsSee page 336.

isdn IDSL (ISDN Digital Subscriber Line) commandsSee page 339.)

sdsl SDSL (Symmetric Digital Subscriber Line) commandsSee page 342.

shdsl G.shdsl commands See page 346.

ADSL Commands

Use the following commands to manage the ADSL (Asymmetric Digital Subscriber Line) link for an ADSL router.

ADSL ?

Lists the supported keywords.

adsl ?

Response:

```
ADSL commands:
?         restart     stats       speed
```

ADSL RESTART

Resynchronizes the modem with the CO (Central Office) equipment.

adsl restart

Response:

```
# 12/02/1997-12:47:46:ADSL: Idle
12/02/1997-12:47:46:ADSL: Startup initiated
12/02/1997-12:47:48:ADSL: Startup training in progress
12/02/1997-12:47:54:ADSL: Modem started successfully
12/02/1997-12:47:54:ADSL: Near Avg SQ #: 44 dB [   3]
12/02/1997-12:47:54:ADSL: Far  Avg SQ #: 44 dB [   3]
12/02/1997-12:47:54:ADSL: Downstream rate: 6272 Kb/s, Upstream rate:
1088 Kb/s
12/02/1997-12:47:54:DOD: connecting to internet @ 0*38 over ATM_VC/1
12/02/1997-12:47:56:ADSL: Data Mode
DUM: BR CHG ATM_VC/1 - to internet now forwarding
```

ADSL SPEED

Displays the current downstream and upstream rates.The actual speed is set by the DSLAM.

adsl speed

Example: `adsl speed`

Response:

```
downstream rate: 6272 Kb/s, upstream rate: 1088 Kb/s
```

ADSL STATS

Shows the current error status for the ADSL connection.

adsl stats [clear]

clear Option used to reset the counters.

Example: `adsl stats`

Response:

```
ASDL Statistics:
  Out of frame errors .....  0
  HEC errors received .....  0
  CRC errors received .....  0
  FEBE errors received ....  0
Remote Out-of-frame .......  0
Remote HEC errors .........  0
```

ATM Commands

Use the following commands to manage the ATM-25 (Asynchronous Transfer Mode) link for an ATM router.

atm pcr **atm speed**	Sets the upstream data rate in cells per second (**pcr**) or kilobits per second (**speed**).
remote setatmtraffic	Allocates bandwidth among remotes.
atm save	Saves the ATM settings.

Commands available to help debug ATM problems are listed on page 204 and page 207.

ATM ?

Lists the supported keywords.

atm ?

Example: `atm ?`

Response:

```
ATM commands:
?               save            speed
pcr
```

ATM PCR

Sets the speed of the ATM link in cells per second.

The default upstream speed is 768 cells/second. Generally, your Network Service Provider should provide you with your speed value. If your service provider states your speed value in kilobits per second, enter the value using the command **atm speed** (page 329).

Note: The speed value you enter may not be the actual upstream speed you get. When the command changes the processor clocks, only certain discrete values are allowed. The speed you get is the allowed speed value that is equal to or the next lower value to the value you entered (see the example below).

atm pcr [*cells/seconds*]

cells/second	Upstream speed in cells/seconds as provided by your service provider (integer, 294 through 18867). If you omit this value, a message states the current upstream speed.

Examples:

The following command requests the current speed.

```
# atm pcr
ATM Upstream Rate:   326 Kb/sec or   768 cells/sec
```

The following command requests a speed of 1200 cells/second. However, 1200 is not one of the discrete speed values allowed, so the next lower value, 1179 cells/second, is set, as indicated by the message.

```
# atm pcr 1200
ATM Upstream Rate:   500 Kb/sec or  1179 cells/sec
```

ATM SAVE

Saves the ATM configuration settings.

atm save

Example: `atm save`

ATM SPEED

Sets the speed of the ATM link in kilobits per second.

The default upstream speed is 326 Kb/s. Generally, your Network Service Provider should provide you with your speed value. If your service provider states your speed value in cells per second, enter the value using the command **atm pcr** (page 328).

Note: The speed value you enter may not be the actual upstream speed you get. When the command changes the processor clocks, only certain discrete values are allowed. The speed you get is the allowed speed value that is equal to or the next lower value to the value you entered (see the example below).

atm speed [*Kb/s*]

Kb/s Upstream speed in kilobits/second as provided by your service provider (integer, 125 three 8000). If you omit this value, a message states the current upstream speed.

Examples:

The following command requests the current speed.

```
# atm speed
ATM Upstream Rate:   326 Kb/sec or   768 cells/sec
```

The following command requests a speed of 512 kilobits/second. However, 512 is not one of the discrete speed values allowed, so the next lower value, 500 kilobits/second, is set, as indicated by the message.

```
# atm speed 512
ATM Upstream Rate:   500 Kb/sec or  1179 cells/sec
```

REMOTE SETATMTRAFFIC

Sets ATM traffic-shaping on a remote router. ATM traffic-shaping allows the user to set the average rate at which cells are sent, that is, the Sustained Cell Rate (SCR), to a value lower than the ATM link speed, the Peak Cell Rate (PCR).

ATM traffic-shaping should be used to allocate bandwidth whenever more than one remote router is defined. Enter a **remote setATMTraffic** command for each remote. For example, if you have five remotes, enter five commands to allocate the bandwidth.

If no ATM traffic values are set, ATM traffic for the remote is shaped using UBR (unspecified bit rate).

If a CBR (constant bit rate) is required, then specify 1 as the Maximum Burst Size (MBS). If a VBR (Variable Bit Rate) is required, specify a value greater than 1 as the Maximum Burst Size (MBS).

To disable ATM traffic-shaping, use the command **remote setATMTraffic 0 0** <*remoteName*>

remote setATMTraffic <*scr*> <*mbs*> <*remoteName*>

scr	Sustained Cell Rate (cells per second).
mbs	Maximum Burst Size (cells). For a constant bit rate (CBR), specify 1; for a variable bit rate (VBR), specify a value greater than 1.
remoteName	Name of the remote router (character string).

Examples:

Assuming that the ATM link speed (upstream) is 200 Kb/s 471 cells/s and an average upstream data rate of 20 Kbps (47 cells/s) is desired, you would issue the following command:

```
remote setATMtraffic 47 31 HQ
```

If a constant bit rate (CBR) is required, use the following command:

```
remote setATMtraffic 47 1 HQ
```

The following command disables ATM traffic-shaping on remote router HQ:

```
remote setATMtraffic 0 0 HQ
```

DMT Commands

These commands manage the ADSL DMT (Discrete MultiTone) router. To see additional DMT debug commands, see ADSL DMT Router Debug Commands, on page 206.

DMT LINK

Selects the link type for the ADSL DMT router. The link type survives reboots.

Normally, the CO and CPE negotiate the link type to be used. Use the **dmt link** command when you do not want the CO and CPE to negotiate the link type, but instead want to specify the type of data link required.

Caution: This command forces the CPE into the specified mode. It is not for normal use.

dmt link DEFAULT | T1_413 | G_DMT | G_LITE | MULTIMODE

DEFAULT	Default value. The CO and CPE negotiate the link type used.
T1_413	ANSI standard T1.413
G_DMT	G.dmt standard
G_LITE	ITU G.Lite standard
MULTIMODE	The CO and CPE negotiate the link type used.

DMT MODE

The **dmt mode** command can request one of three modes: ANSI, no_Trellis_ANSI, and UAWG.

UAWG mode is becoming obsolete.

No Trellis encoding for T1.413 ANSI ADSL is only needed where auto-negotiation is not supported for Trellis.

dmt mode ansi | no_trellis_ansi | uawg

Dual-Ethernet Router (ETH) Commands

The following Ethernet commands are used to manage the Ethernet interfaces of the Dual-Ethernet (Ethernet-to-Ethernet) router and thus are *specific to that type of router only*. For the other Ethernet commands, see page 262.

- The Dual-Ethernet router has two interfaces:

  ETH/0 Hub with four 10Base-T connectors

  ETH/1 Single 10Base-T connector

- This Dual-Ethernet router may be configured via the Web Browser GUI or from the Command Line Interface (CLI). To set up any DHCP options and to configure optional features like IP filtering, you must use the CLI.
- For configuration information, refer to Dual-Ethernet Router Configuration, on page 47 and the *Customer Release Notes* provided with the Dual-Ethernet router.
- If you use the **Boot from Network** option from the boot menu to perform a boot code update, the boot request is sent from the ETH/0 interface only.

ETH BR ENABLE

Enables bridging in a Dual-Ethernet environment. This command requires rebooting the router for the change to take effect.

eth br enable

Example: `eth br enable`

ETH BR DISABLE

Disables bridging in a Dual-Ethernet environment.

Note: This command requires rebooting the router for the change to take effect.

eth br disable

Example: `eth br disable`

ETH BR OPTIONS

Sets controls on bridging for the Ethernet interface To see the current bridge settings for the Ethernet interface, use the **eth list** command.

Warning: Do not change the **stp** setting without approval from your system administrator.

eth br options <*option*> **on** | **off** [<*port#*>]

option **stp**

Set this option to **on** to use the Spanning Tree Protocol (STP). The default is **on**.

STP is used to detect bridging loops. Set this option to **off** only if the bridging peers do not support the Spanning Tree Protocol or if you are certain that no bridging loops could exist. When STP is disabled on an interface, any STP packets received on that interface are ignored.

Note: The Spanning Tree Protocol adds a 40-second delay each time the ADSL or ATM link comes up while the interface determines if there is a bridging loop.

pppoeOnly

Set this option to **on** to limit this Ethernet port to bridging PPPoE traffic only. If the option is set to **off**, then the port can bridge any traffic, including PPPoE traffic. The default is **off**.

port# Ethernet port number (0 or 1). The default is 0.

Examples:

The following command turns off the spanning tree protocol for Ethernet port 0.

```
eth br options stp off
```

The following command configures Ethernet port 1 so that only PPPoE traffic is bridged through it.

```
eth br options pppoeonly on 1
```

Frame Commands

FRAME ?

Lists the frame commands.

frame ?

Example:

```
# frame ?
Frame Commands:
?                   help                lmi
cmpPlay             voice
```

FRAME CMPPLAY

Selects activation in routing or bridge mode. The default is routing mode. This command is applicable only when the router is configured using Copper Mountain Plug & Play (see Bridge or Router?, on page 49).

frame cmPPlay < route | bridge >

route Selects routing mode.

bridge Selects bridging mode.

Example:

```
frame cmpplay bridge
```

FRAME LMI

Turns frame LMI either on or off.

frame lmi < on | off>

Example:

```
# frame lmi on
LMI is on
```

FRAME STATS

Displays frame relay statistics.

Although it is not an end-to-end loopback test, the command output does show counters for data sent and received as well as LMI events.

frame stats

Example:

```
# frame stats
FR/0 Frame Relay Statistics
ANSI LMI:
Protocol Errors..................... 0
Unknown Msg Recv.................... 0
T391 Timeouts....................... 0
PVC Status Changes.................. 0
StatusEnq Sent...................... 0
Status Recv......................... 0
StatusEnq Recv...................... 0
Unconfigured DLCIs recv in Status Msgs. 0

    LMI Stats for DLCI.............. 22
    LMI State....................... UNKNOWN
    Status State Changes............ 0
    Active to Not Active Changes.... 0
    Not Active to Active Changes.... 0
    Data Packets In................. 0
    Data Packets Out................ 0
    Data Packets Out Queued......... 0
    Data Packets Out (dropped Q Full).. 0
    Voice Cells In.................. 0
    Voice Cells In (with errors).... 0
    Voice Cells Out................. 0

    LMI Stats for DLCI.............. 16
    LMI State....................... UNKNOWN
    Status State Changes............ 0
    Active to Not Active Changes.... 0
    Not Active to Active Changes.... 0
    Data Packets In................. 0
    Data Packets Out................ 0
    Data Packets Out Queued......... 0
    Data Packets Out (dropped Q Full).. 0
    Voice Cells In.................. 0
    Voice Cells In (with errors).... 0
    Voice Cells Out................. 0

    Data Out (Delayed by Voice)..... 0
```

FRAME VOICE

Displays the voice DLCI for voice routers.

frame voice

Example:

```
# frame voice
Voice DLCI is 22
```

HDSL Commands

Use the following commands to manage the HDSL (High-Speed Digital Subscriber Line) link for an HDSL router.

General Information about HDSL

- **Line activation**

  Line activation is independent of network settings. During activation, the Link light (on the front panel of the router) first is yellow and then turns green when the link becomes active.

  The router at the CPE end will try auto-speed detection, starting at 384 and then try to detect the next higher speed (for about 30 seconds per speed). The WAN light should turn yellow, then green, when the link has activated.

  Auto-speed detection can be turned off with the command **hdsl speed noauto**.

  If the line was previously set to "no auto-speed" (noauto), the Link light will be amber instead, when the line tries to activate.

  The **ifs** command displays the Link as either off or opened when successfully activated. Following is a sample output.

  Sample:

```
ifs
Interface      Speed      In %     Out %    Protocol     State
Connection
ETHERNET/0     10.0mb     0%/0%    0%/0%    (Ethernet)   OPENED
HDSL/0         384kb      0%/0%    0%/0%    (HDSL)       OPENED
CONSOLE/0      9600 b     0%/0%    0%/0%    (TTY)        OPENED
```

- **Auto-speed sequence**

  Auto-speed starts with the lower speed (384) and then tries to activate for 30 seconds. If no activation takes place, it attempts the next higher speed. The time intervals between activation may change if the modems don't activate as expected. Following is a correct activation output.

```
03/09/1998-17:11:59:HDSL: Deactivated
03/09/1998-17:12:22:HDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:13:00:HDSL: Deactivated
03/09/1998-17:13:01:HDSL: CPE is Activating at 1168 Kb/s
03/09/1998-17:13:32:HDSL: Deactivated
03/09/1998-17:13:32:HDSL: CPE is Activating at 1168 Kb/s
03/09/1998-17:14:11:HDSL: Deactivated
03/09/1998-17:14:12:HDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:14:51:HDSL: Activated
03/09/1998-17:14:53:FRAMER: The framer is synchronized
```

HDSL ?

Lists the supported keywords.

hdsl ?

Example: `hdsl ?`

Response:

```
HDSL commands:
?               help          terminal
save            speed
```

HDSL SAVE

Saves the HDSL-related changes across restarts and reboots.

hdsl save

Example: `hdsl save`

HDSL SPEED

Manages the line speed for the HDSL interface, as follows

CO end: Sets the speed manually on the Central Office (CO) end only.

CPE end: The router on the Customer Premises End (CPE) is always in auto-speed mode: it uses an auto-speed algorithm to attempt to match the CO speed. The command **hdsl speed noauto** is used to override auto-speed.

Note 1: The command **hdsl speed** (with no option) displays the current speed if the modem has activated successfully.

Note 2: hdsl speed noauto should be followed by the command **hdsl save** to be persistent across restarts and reboots.

Note 3: During auto-speed search, use the command **hdsl speed** <*speed*> to stop the search and restart it at the speed you just entered.

hdsl speed [384 | 1168 | noauto]

384	Default speed for the CO.
1168	Authorized non-default speeds for the CO in Mbps.
noauto	Used to override auto-speed on the CPE.

Example: `hdsl speed 1168`
`hdsl speed noauto`
`hdsl speed`

HDSL TERMINAL

The router is by default configured as the Customer Premises Equipment (CPE). Use this command if you intend to configure the router as the Central Office equipment (CO).

hdsl terminal cpe defines the CPE end (default configuration)

hdsl terminal co defines the CO end.

hdsl terminal displays the current settings.

hdsl terminal [cpe|co]

co This option lets you define the router as the CO.

Example: `hdsl terminal`

Response:
`Customer Premises`

Example: `hdsl terminal co`

IDSL Commands

An IDSL (ISDN Digital Subscriber Line) delivers a maximum symmetric 144 Kbps of bandwidth. The IDSL bandwidth is composed of two 64 Kbps B channels, plus one 16 Kbps D channel. Your speed setting indicates the channels that you are using.

When using Frame Relay:

- Your IDSL switch setting indicates your committed bandwidth (FR64, FR128, or FR144).
- The IDSL router can support several DLCI virtual circuits over a Frame-Relay IDSL link. However, a typical connection to the Internet requires only one DLCI. The DLCI number must match the DLCI of the remote end.

This section describes the following commands used to manage your IDSL router:

idsl list	Lists the current Frame Relay switch type (FR64, FR128, or FR144).
idsl save	Saves the IDSL changes.
idsl set switch	Changes the Frame Relay switch type.
idsl set speed	Changes the IDSL speed (64, 128, or 144).
remote setdlci	Specifies the DLCI for the remote router entry.
remote setprotocol	Selects the link protocol for the remote router entry (PPP, Frame Relay, or MER).

IDSL LIST

Lists the current switch type. To change the switch type, use the **idsl set switch** command.

idsl list

Example:

```
# idsl list
Switch type is FR128
```

IDSL SAVE

Saves IDSL-related changes across restarts and reboots. Changes that are not saved are discarded.

idsl save

Example:

```
# idsl save
```

IDSL SET SPEED

Specifies the speed of the IDSL connection.

The IDSL bandwidth is composed of two 64 Kbps B channels, plus one 16 Kbps D channel. Your speed setting indicates the channels that you are using.

idsl set speed 64 | 128 | 144

64 64 Kbps (one channel)

128 128 Kbps (two channels)

144 144 Kbps (three channels)

Example:

```
# idsl set speed 144
```

IDSL SET SWITCH

Specifies link speeds of 64, 128, or 144 Kbps for the IDSL connection.

idsl set switch FR64 | FR128 | FR144

FR64 Link speed of 64 Kbps

FR128 Link speed of 128 Kbps

FR144 Link speed of 144 Kbps

Example:

```
# idsl set switch fr144
```

REMOTE SETDLCI

This command sets the DLCI for the remote router entry. The DLCI (Data Link Connection Identifier) is an address identifying a logical connection in a Frame Relay environment. The DLCI is generally provided by the Network Service Provider.

The IDSL router can support several DLCI virtual circuits over a Frame-Relay IDSL link. However, a typical connection to the Internet requires only one DLCI. The DLCI number must match the DLCI of the remote end.

remote setDLCI <*dlcinumber*> <*remoteName*>

dlcinumber Frame Relay number identifying the data-link connection.

remoteName Name of the remote router (character string).

Example: `remote setDLCI 16 HQ`

REMOTE SETPROTOCOL

This IDSL-specific command is used to select the appropriate link protocol for your IDSL connection. Your Network Service Provider will tell you which link protocol to use.

remote setProtocol PPP | FR | MER *<remoteName>*

PPP	PPP protocol with no encapsulation.
FR	RFC 1490 protocol (Multiprotocol encapsulation over Frame Relay).
MER	RFC 1490 protocol with MAC Encapsulated Routing.
remoteName	Name of the remote router (character string).

Example: `remote setProtocol FR HQ`

SDSL Commands

The commands in this section manage the Symmetric Digital Subscriber Line (SDSL) link for an SDSL router.

- **sdsl preact** Disables or re-enables autobaud pre-activation.
- **sdsl speed** Displays and sets the line speed.
- **sdsl stats** Displays and clears SDSL statistics.
- **sdsl terminal** Redefines the router as CO equipment.

SDSL Line Activation

Line activation is independent of network settings. During activation, the LINK LED (on the front panel of the router) is first amber and then turns green when the link becomes active. The WAN LED should turn amber, then green, when the link has activated.

If auto-speed detection was turned off for the line (**sdsl speed noauto**), the Link LED is red when the line tries to activate.

The **ifs** command displays the Link as either off or opened when it has successfully activated. The following is a sample output.

Sample:

```
ifs
Interface      Speed      In %     Out %     Protocol    State
Connection
ETHERNET/0     10.0mb     0%/0%    0%/0%    (Ethernet)   OPENED
SDSL/0         384kb      0%/0%    0%/0%    (ATM)        OPENED
CONSOLE/0      9600 b     0%/0%    0%/0%    (TTY)        OPENED
```

SDSL Line Speed

In general, the line activates at the speed it was last activated. The line speed can be changed by several means depending on the router model. One way is by setting the speed manually with an **sdsl speed** command.

In general, if the speed is not set otherwise, the router at the CPE end will try auto-speed detection (unless it has been disabled).

Auto-speed detection attempts to activate the line at different speeds (for about 30 seconds per speed) until the line is activated. The LINK LED may flash more rapidly when faster speeds are being attempted. The following is output from a successful activation.

```
03/09/1998-17:11:59:SDSL: Deactivated
03/09/1998-17:12:22:SDSL: CPE is Activating at 768 Kb/s
03/09/1998-17:13:00:SDSL: Deactivated
03/09/1998-17:13:01:SDSL: CPE is Activating at 1152 Kb/s
03/09/1998-17:13:32:SDSL: Deactivated
03/09/1998-17:13:32:SDSL: CPE is Activating at 1152 Kb/s
03/09/1998-17:14:11:SDSL: Deactivated
03/09/1998-17:14:12:SDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:14:51:SDSL: Activated
03/09/1998-17:14:53:FRAMER: The framer is synchronized
```

```
03/09/1998-17:15:19:DOD: connecting to co @ 0*38 over ATM-VC/1
03/09/1998-17:15:35:DOD: link to co over ATM-VC/1 is now up
03/09/1998-17:15:57:SDSL: Line Rate at last activation saved
```

Autobaud pre-activation

The previous section showed an example in which auto-speed detection attempted several speeds, before settling on the best speed for the connection. In some cases, this process can require substantial time. The autobaud feature, if available for your router and DSLAM, can shorten the connection set-up time by determining the probable optimal speed before the connection begins.

If the autobaud feature is available and selected, its pre-activation phase automatically determines the maximum speed that can be supported by a specific loop. It probes the channel and characterizes the line to allow the connection to begin at a speed closer to the optimal speed.

Later, after activation, the autobaud feature checks the line quality to determine the optimal speed. If the autobaud feature is used, the auto-speed detection described in the previous section is not performed; however, you may still set the speed manually with an **sdsl speed** command if you wish.

A command is available to turn off autobaud pre-activation (**sdsl preact off**, page 344). A message is sent before line activation indicating whether autobaud pre-activation was used. The following example shows the message sent when pre-activation is available and the response to the **sdsl speed** command:

```
08/16/2000-16:11:06:SDSL: Using preactivation-determined rate of 2320 Kb/s

08/16/2000-16:11:06:SDSL: CPE is Activating at 2320 Kb/s
08/16/2000-16:11:20:SDSL: Activated at 2320 Kb/s
08/16/2000-16:11:20:FRAMER: The framer is synchronized
# sdsl speed
SDSL Current Speed (CO-controlled): 2320 Kb/s
```

The next example shows the message sent when pre-activation is not available:

```
08/16/2000-16:13:28:SDSL: Preactivation unavailable, using rate of 2320 Kb/s

08/16/2000-16:13:28:SDSL: CPE is Activating at 2320 Kb/s
08/16/2000-16:13:41:SDSL: Activated at 2320 Kb/s
08/16/2000-16:13:42:FRAMER: The framer is synchronized
```

SDSL ?

Lists the supported keywords for the **sdsl** command.

sdsl ?

Example:

```
# sdsl ?
SDSL commands:
?                   help                speed
save                stats               terminal
```

SDSL PREACT

Displays and/or changes the autobaud pre-activation status.

The default status is on. However, to be effective, autobaud pre-activation must also be enabled at the Central Office (CO) end of the connection.

Note: Remember to enter an **sdsl save** or **save** command to save SDSL changes across restarts and reboots.

To determine the current pre-activation status, enter **sdsl preact**.

For more information on the autobaud feature, see Autobaud pre-activation, on page 343.

sdsl preact [on | off]

on Enables pre-activation at the customer premises (CPE) end. (To be effective, pre-activation must also be enabled at the CO end.)

off Disables pre-activation.

Example:

The first command displays the current pre-activation status. The second command disables pre-activation.

```
# sdsl preact
Preactivation enabled
# sdsl preact off
Preactivation disabled
```

SDSL SAVE

Saves SDSL configuration changes across restarts and reboots.

sdsl save

Example: `sdsl save`

SDSL SPEED

Manages the speed of the SDSL line.

At the Central Office (CO) end, the command sets the speed manually only.

At the Customer Premises Equipment (CPE) end, the command can:

- Display the current speed setting and list the available speeds (**sdsl speed**)
- Manually set the speed (**sdsl speed** <***speed***>)
- Override auto-speed detection (**sdsl speed noauto**)

Note: To re-instate auto-speed detection, enter an **sdsl speed** <*speed*> command.

Note: Remember to enter an **sdsl save** or **save** command to save SDSL changes across restarts and reboots.

sdsl speed [<*speed*> | noauto]

speed Speed in kbps. To see the speeds available for the model type, enter **sdsl speed**. If the auto-speed search is in progress, this command stops the search and sets the line speed as specified on the command.

noauto Overrides auto-speed detection. If auto-speed detection is disabled, the Link light on the front panel is amber when the line tries to activate.

(Auto-speed detection is reinstated if you enter an **sdsl speed** <*speed*> command.)

Example:

The example shows three commands:

1. Displays the current line speed, indicates that the line speed is set by auto-speed detection [AUTO], and lists the available speed options.
2. Requests a line speed of 1152 Kb/s.
3. Shows that the line speed has been changed to 1151 Kb/s and that auto-speed detection is no longer in effect (the [AUTO] indicator is not displayed).

```
# sdsl speed
SDSL Current Speed: [AUTO] 768 Kb/s
usage: sdsl speed <value in Kb/s> [ 192 384 768 1152 1536 ] | noauto
# sdsl speed 1152
# sdsl speed
SDSL Current Speed: 1152 Kb/s
usage: sdsl speed <value in Kb/s> [ 192 384 768 1152 1536 ] | noauto
```

SDSL STATS

Displays SDSL frame statistics. It can also clear the SDSL statistic counters.

sdsl stats [clear]

clear Clears all SDSL statistics counters.

Example:

```
# sdsl stats
FRAMER Statistics:
  Framer Interrupts...... 2118
  Out of frame errors.... 1
  HEC errors received.... 16
  CRC errors received.... 3
  FEBE errors received... 2
  Remote Out-of-frame.... 16
  Remote HEC errors...... 0
```

SDSL TERMINAL

Displays and/or changes the router’s status as CO or CPE.

The router is, by default, configured as Customer Premises Equipment (CPE). Use this command if you intend to configure the router as Central Office equipment (CO).

To determine the current CO/CPE setting, enter **sdsl terminal**.

sdsl terminal [cpe | co]

cpe Defines the router as the customer premises (CPE) equipment.

co Defines the router as the central office (CO) equipment.

Example:

```
# sdsl terminal
Customer Premises
# sdsl terminal co
Central Office
```

SHDSL Commands

The commands in this section manage the WAN link for a G.shdsl router.

- **shdsl annex** Selects annex A or annex B.
- **shdsl list** Lists G.shdsl configuration.
- **shdsl margin** Changes the acceptable noise margin.
- **shdsl rateMode** Selects adaptive or fixed rate mode.
- **shdsl restart** Restarts the G.shdsl WAN interface.
- **shdsl save** Saves SHDSL configuration changes.
- **shdsl speed** Displays and sets the line speed.
- **shdsl stats** Displays or clears G.shdsl statistics.
- **shdsl terminal** Redefines the router as CO (Central Office) equipment.
- **shdsl ver** Displays the G.shdsl version level.

SHDSL ?

Lists the supported keywords for the **shdsl** command.

shdsl ? | help

Example:

```
# shdsl ?
SHDSL commands:
?                help             terminal
restart          stats            speed
ver              annex            rateMode
```

```
margin              save                list
```

SHDSL ANNEX

Selects annex A or annex B of the G.shdsl standard. The annex used depends on the DSLAM the router is to connect to. In general, annex B is used in Europe and annex A is used in the rest of the world.

To see the current annex selection, enter **shdsl annex** without a parameter.

shdsl annex [A | B]

Example:

```
# shdsl annex
Annex A
# shdsl annex B
```

SHDSL LIST

Lists the current configuration of the G.shdsl interface.

shdsl list

Example:

```
# shdsl list
G.SHDSL INTERFACE CONFIGURATION:
  Terminal ..................... CPE
  GTI SHDSL version ............ R1.2
  Requested speed .............. 2312 Kb/s
  Actual speed ................. 0 Kb/s
  Startup margin ............... 6 dB
  Rate mode .................... Adaptive
  Annex ........................ B
```

SHDSL MARGIN

Specifies the acceptable noise margin in decibels. If the connection is unstable, you may need to increase the margin.

To see the current noise margin, enter **shdsl margin** without a parameter.

shdsl margin [*dB*]

dB Noise margin in decibels (0 - 15). The default is 6.

Example:

```
# shdsl margin
Margin = 6
# shdsl margin 7
```

SHDSL RATEMODE

Selects adaptive or fixed rate mode. The default is adaptive rate mode.

To see the current rate mode, enter **shdsl rateMode** without a parameter.

shdsl rateMode [Adaptive | Fixed]

Example:

```
# shdsl ratemode
Adaptive
# shdsl ratemode fixed
```

SHDSL RESTART

Restarts the G.shdsl WAN interface. (Unlike a reboot, a restart does not discard unsaved changes.)

Note: The WAN interface is restarted automatically when you change the speed (**shdsl speed**) or change the CO or CPE designation (**shdsl terminal**).

shdsl restart

SHDSL SAVE

Saves SHDSL configuration changes across restarts and reboots.

(To save SHDSL changes *and* all other configuration changes, use the command **save**.)

shdsl save

Example: `shdsl save`

SHDSL SPEED

Manages the speed of the SHDSL line.

Note: By default, it is assumed that the router is Customer Premises Equipment (CPE) and the line speed desired is the maximum allowed by the central office (CO).

This command can:

- Display the current requested speed and actual speed (**shdsl speed** with no parameter).

  If the actual speed shown is 0 (zero), the line is down.

- Manually set the speed (**shdsl speed** <***speed***>) (You might request a lower speed to improve stability.)

  Note: A speed change automatically restarts the G.shdsl WAN interface. Remember to **save** the speed change if you want it to persist across reboots.

- Select auto-speed detection (**shdsl speed auto**). You should then restart the WAN interface with the command **shdsl restart**.

shdsl speed [<*speed*> | auto]

speed Requested speed in kbps. The possible speeds range from 72 kbps to 2312 kbps in increments of 64 kbps. If you specify a value between steps, the speed is set to the next lower step.

auto Selects auto-speed detection. Enter the command **shdsl restart** to carry out this change.

Example:

```
# shdsl speed
Requested speed: 2312 Kb/s
Actual speed:    2312 Kb/s
# shdsl speed 1096
```

SHDSL STATS

Displays SHDSL statistics. The statistics are kept for 24 hours and then cleared. You can also manually clear the statistics with the **clear** option.

Statistics kept include: line signal quality (SQ), loss of sync word (LOSW), far end bit error (FEBE), and loop attenuation (Loop Attn).

shdsl stats [clear]

clear Resets the statistics counters to zero.

Example:

```
# shdsl stats
SHDSL 24hr statistics displayed in time period of 15 minutes:

System up:   0 days 2 hours 9 minutes
Line up:     0 days 2 hours 9 minutes
Line SQ:     38 38 38 40 40 39 39 39 40
CRC Errors:  2 0 0 0 0 0 0 0 0
LOSW Errors: 0 0 0 0 0 0 0 0 0
FEBE Errors: 0 0 0 0 0 0 0 0 0
Loop Attn:   -2 -2 -2 -2 -2 -2 -2 -2 -2
# shdsl stats clear
# shdsl stats
SHDSL 24hr statistics displayed in time period of 15 minutes:

System up:   0 days 2 hours 9 minutes
Line up:     0 days 0 hours 0 minutes
Line SQ:     40
CRC Errors:  0
LOSW Errors: 0
FEBE Errors: 0
Loop Attn:   0
```

SHDSL TERMINAL

Displays and/or changes the router's designation as CO (Central Office) or CPE (Customer Premises Equipment).

By default, the router is assumed to be CPE. Use this command if you intend to use the router as CO.

To determine the current CO/CPE setting, enter **shdsl terminal** without a parameter.

shdsl terminal [cpe | co]

cpe	Defines the router as the customer premises (CPE) equipment.
co	Defines the router as the central office (CO) equipment.

Example:

```
# shdsl terminal
We are in CPE mode
Usage: shdsl terminal [co|cpe]
# shdsl terminal co
```

SHDSL VER

Displays the G.shdsl version level of the modem firmware.

shdsl ver

Example:

```
# shdsl ver
GTI SHDSL Version R1.2
```

DHCP Commands

The following DHCP (Dynamic Host Configuration Protocol) commands allow you to:

- Enable and disable subnetworks and client leases.
- Add subnetworks and client leases.
- Set the lease time.
- Change client leases manually.
- Set option values globally, for a subnetwork, or for a client lease.
- Enable/disable BootP.
- Use BootP to specify the boot server.
- Define option types.

To read about DHCP concepts and the DHCP configuration process, see DHCP (Dynamic Host Configuration Protocol), on page 85.

DHCP ?

Lists the supported keywords.

dhcp ?

Response:

```
Sub-commands for dhcp
?                   help                set
list                bootp               clear
enable              add                 addrelay
del                 delrelay            disable
```

DHCP ADD

Provides one of three types of DHCP definitions: subnetwork, client lease, or option type.

To delete any of these DHCP definitions, use the command **dhcp del** (page 355).

Defines a subnetwork:

dhcp add *<net> <mask>*

net IP address of the subnetwork lease (4 decimals separated by periods).

mask IP network mask (4 decimals separated by periods).

Example:

```
dhcp add 192.168.254.0.255.255.255.0
```

Defines a client lease:

dhcp add *<ipaddr>*

ipaddr IP address of the client lease (4 decimals separated by periods).

Example:

```
dhcp add 192.168.254.31
```

Defines an option type:

dhcp add *<code> <min> <max> <type>*

code User-defined code (128 - 254, or a keyword).

min Minimum number of values.

max Maximum number of values.

type Byte | word | long | longint | binary | ipaddress | string

Example:

```
dhcp add 128 1 4 ipAddress
```

The code, 128, allows IP addresses, the server has a minimum of one, up to a maximum of four, IP addresses, and the type is “ipaddress”).

DHCP ADDRELAY

Adds an address to the DHCP relay list. (This list is also the BootP server list.) To see the current server address, use the command **dhcp addrelay** with no parameters.

While the relay list contains at least one address, the DHCP server in the router is disabled, and the router forwards all DHCP requests and BootP requests to all servers in the relay list. (A DHCP request is issued whenever a device attempts to acquire an IP address). It forwards every reply received from any of the servers in the relay list to the appropriate LAN.

To remove an address from the list, use the **dhcp delRelay** command (page 355). For further discussion, see Configuring BootP/DHCP Relays, on page 92.

dhcp addRelay <*ipaddr*>

ipaddr IP address of a server (4 decimals separated by periods).

Example:

```
# dhcp addrelay 128.1.210.64
# dhcp addrelay
BOOTP/DHCP Server address: 128.1.210.64
```

DHCP BOOTP ALLOW

Allows a BootP request to be processed for a particular client or subnet.

dhcp bootp allow <*net*>|<*ipaddr*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Example: `dhcp bootp allow 192.168.254.0`

DHCP BOOTP DISALLOW

Denies processing of a BootP request for a particular client or subnet.

dhcp bootp disallow <*net*>|<*ipaddr*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Example: `dhcp bootp disallow 192.168.254.0`

DHCP BOOTP FILE

Specifies the boot file name (kernel) and the subnet to which it applies.

Note: Be sure to specify the TFTP server IP address when you specify the file using the command **dhcp bootp tftpserver** (page 355).

dhcp bootp file [<*net*>|<*ipaddr*>] <*name*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

name Name of the file to boot from; the default name for this file is KERNEL.F2K.

Example: `dhcp bootp file 192.168.254.0 Kernel.f2k`

DHCP BOOTP TFTPSERVER

Specifies the TFTP server (boot server).

dhcp bootp tftpserver [<*net*>|<*ipaddr*>]<*tftpserver ipaddr*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

tftpserver ipaddr IP address of the TFTP server in the format of 4 decimals separated by periods. To clear the IP address of the server, use 0.0.0.0.

Examples:

```
dhcp bootp tftpserver 192.168.254.7
dhcp bootp tftpserver 192.168.254.0 192.168.254.8
dhcp bootp tftpserver 192.168.254.21 192.168.254.9
dhcp bootp tftpserver 0.0.0.0
```

DHCP CLEAR ADDRESSES

Clears the values from a pool of addresses.

dhcp clear addresses <*net*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

Example: `dhcp clear addresses 192.168.254.0`

DHCP CLEAR ALL RECORDS

Clears all DHCP information, including all leases and all global DHCP information.

Unlike **erase dhcp**, this command clears all DHCP information from memory, but leaves the DHCP.DAT file intact. If you want to clear the information in the DHCP.DAT file as well, enter a **save** command after **dhcp clear all records**.

Note: You cannot abbreviate the word **records** in the command.

dhcp clear all records

Example:

```
dhcp clear all records
```

DHCP CLEAR EXPIRE

Releases the client lease. It then becomes available for other assignments.

Note: The client is not updated; it still has the old value.

dhcp clear expire <*ipaddr*>

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Example: `dhcp clear expire 192.168.254.12`

DHCP CLEAR VALUEOPTION

Clears the value for a global option, for an option associated with a subnetwork, or with a specific client.

dhcp clear valueoption [<*net*>|<*ipaddr*>] <*code*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

code Code can be a number between 1 and 61 or a keyword. Use the command **dhcp list definedoptions** to list the codes and keywords.

Examples:

```
dhcp clear valueoption 4
dhcp clear valueoption 192.168.254.0 7
dhcp clear valueoption 192.168.254.2 gateway
```

DHCP DEL

Deletes a subnetwork lease, a specific client lease, or a code.

dhcp del <*net* |<*ipaddr*>|<*code*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

code User-defined code (number between 128 to 254 or a keyword).

Example 1: `dhcp del 192.168.254.0`
(deletes this subnetwork)

Example 2: `dhcp del 192.168.254.31`
(deletes this client lease

Example 3: `dhcp del 128`
(deletes this option with code 128)

DHCP DELRELAY

Removes an address from the DHCP relay list. (This list is also the BootP server list.)

To remove all addresses from the list, use **dhcp delRelay all**. If you remove all addresses from the DHCP relay list, the DHCP server is re-enabled and resumes processing DHCP requests and also BootP requests (if BootP processing is enabled).

To add an address to the list, use the **dhcp addRelay** command (page 352). For further discussion, see Configuring BootP/DHCP Relays, on page 92.

dhcp delRelay <*ipaddr*> | all

ipaddr IP address to be removed from the list (4 decimals separated by periods).

all Removes all addresses from the list.

Examples:

```
dhcp delrelay 128.1.210.64

dhcp delrelay all
```

DHCP DISABLE

Disables a subnetwork or a client lease.

dhcp disable all | <*net*> | <*ipaddr*>

all Disables all subnets.

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Examples:
```
dhcp disable 192.168.254.0
dhcp disable 192.168.254.17
```

DHCP ENABLE

Enables a subnetwork or a client lease.

dhcp enable all | <*net*>|<*ipaddr*>

all Enables all subnets.

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Examples:
```
dhcp enable 192.168.254.0
dhcp enable 192.168.254.17
```

DHCP LIST

Lists global, subnetwork, and client lease information.

dhcp list | <*net*>|<*ipaddr*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

ipaddr IP address of the client lease in the format of 4 decimals separated by periods.

Example 1:

The following command lists global information:

```
#dhcp list
   bootp server..............     none
   bootp file................
   DOMAINNAMESERVER (6).......     192.168.210.20 192.84.210.21
   DOMAINNAME (15)............     efficient.com
   WINSSERVER (44)............     192.168.254.73
Subnet 192.168.254.0, Enabled
   Mask......................     255.255.255.0
   first ip address..........     192.168.254.2
   last ip address...........     192.168.254.253
   lease.....................     Default
```

```
    bootp.....................     not allowed
    bootp server..............     none
    bootp file................
  GATEWAY (3)192.168.254.254
  client 192.168.254.2, Ena, jo-computer, Expired
  client 192.168.254.3, Ena, Jo, 1999/5/16 11:31:33
```

Example 2:

The following command lists information for client 192.168.254.3:

```
  #dhcp list 192.168.254.3
  Client 192.168.254.3, Enabled
      lease.....................    Default
      expires...................    1998/5/16 11:31:33
      bootp.....................    not allowed
      bootp server..............     none
      bootp file................
  HOSTNAME (12)..................    JO
  CLIENTIDENTIFIER (61)..........    1 2 96 140 76 149 180
```

Example 3:

The following command lists information for the subnetwork 192.168.254.0:

```
#dhcp list 192.168.254.0
Subnet 192.168.254.0, Enabled
    Mask                              255.255.255.0
    first ip address                  192.168.254.2
    last ip address                   192.168.254.253
    lease                             Default
    bootp                             not allowed
    bootp server                      none
    bootp file
    GATEWAY (3)                       192.168.254.254
client 192.168.254.2, Ena, Jo-computer, Expired
client 192.168.254.3, Ena, Jo, 1998/5/16 11:31:33
```

DHCP LIST DEFINEDOPTIONS

Lists all available predefined and user-defined options.

Note: For description of the predefined options listed below, refer to RFC 1533. A predefined code can be a number between 1 and 61 or a keyword. A user-defined code can be a number between 128 and 254 or a keyword.

dhcp list definedoptions | <*code*> | <*string*>

code Predefined or user-defined number or keyword.

string Character string.

Example 1: To list all available options (they may be predefined as in the list below, and/or user-defined), enter:

```
dhcp list definedoptions
```

Example 1:

The following command lists all available options (predefined and user-defined):

```
      #dhcp list definedoptions
      code TIMEOFFSET (2), 1 occurrence, type LONG
      code GATEWAY (3), 1 to 63 occurrences, type IPADDRESS
      code TIMESERVER (4), 1 to 63 occurrences, type IPADDRESS
      code NAMESERVER (5), 1 to 63 occurrences, type IPADDRESS
      code DOMAINNAMESERVER code SUBNETMASK (1), 1 occurrence, type IPADDRESS-RESERVED
       (6), 1 to 63 occurrences, type IPADDRESS
      code LOGSERVER (7), 1 to 63 occurrences, type IPADDRESS
      code COOKIESERVER (8), 1 to 63 occurrences, type IPADDRESS
      code LPRSERVER (9), 1 to 63 occurrences, type IPADDRESS
      code IMPRESSSERVER (10), 1 to 63 occurrences, type IPADDRESS
      code RESOURCELOCATION (11), 1 to 63 occurrences, type IPADDRESS
      code HOSTNAME (12), 1 to 255 characters, type STRING
      code BOOTFILESIZE (13), 1 occurrence, type WORD
      code MERITDUMPFILE (14), 1 to 255 characters, type STRING
      code DOMAINNAME (15), 1 to 255 characters, type STRING
      code SWAPSERVER (16), 1 occurrence, type IPADDRESS
      code ROOTPATH (17), 1 to 255 characters, type STRING
      code EXTENSIONSPATH (18), 1 to 255 characters, type STRING
      code IPFORWARDING (19), 1 occurrence, type BINARY
      code NONCALSOURCERTE (20), 1 occurrence, type BINARY
      code POLICYFILTER (21), 1 to 31 occurrences, type IPADDRESS
      code MAXDGMREASSEMBLY (22), 1 occurrence, type WORD
      code DEFAULTIPTTL (23), 1 occurrence, type BYTE
      code PATHMTUAGETMOUT (24), 1 occurrence, type LONGINT
      code PATHMTUPLATEAUTBL (25), 1 to 127 occurrences, type WORD
      code INTERFACEMTU (26), 1 occurrence, type WORD
      code ALLSUBNETSLOCAL (27), 1 occurrence, type BINARY
      code BROADCASTADDRESS (28), 1 occurrence, type IPADDRESScode PERFORMMASKDSCVR (29), 1
occurrence, type BINARY
      code MASKSUPPLIER (30), 1 occurrence, type BINARY
      code PERFORMRTRDSCVR (31), 1 occurrence, type BINARY
      code RTRSOLICITADDR (32), 1 occurrence, type IPADDRESS
      code STATICROUTE (33), 1 to 31 occurrences, type IPADDRESS
      code TRAILERENCAP (34), 1 occurrence, type BINARY
      code ARPCACHETIMEOUT (35), 1 occurrence, type LONGINT
      code ETHERNETENCAP (36), 1 occurrence, type BINARY
      code TCPDEFAULTTTL (37), 1 occurrence, type BYTE
      code TCPKEEPALIVEINTVL (38), 1 occurrence, type LONGINT
      code TCPKEEPALIVEGARBG (39), 1 occurrence, type BINARY
      code NETINFOSVCDOMAIN (40), 1 to 255 characters, type STRING
      code NETINFOSERVERS (41), 1 occurrence, type IPADDRESS
      code NETTIMEPROTOSRVRS (42), 1 occurrence, type IPADDRESS
      code VENDORSPECIFIC (43), 1 to 255 occurrences, type BYTE
      code WINSSERVER (44), 1 to 63 occurrences, type IPADDRESS
      code NETBIOSTCPDGMDIST (45), 1 to 63 occurrences, type IPADDRESS
      code NETBIOSTCPNODETYP (46), 1 occurrence, type BYTE
      code NETBIOSTCPSCOPE (47), 1 to 255 characters, type STRING
      code XWSFONTSERVER (48), 1 to 63 occurrences, type IPADDRESS
      code XWSDISPLAYMANAGER (49), 1 to 63 occurrences, type IPADDRESS
      code REQUESTEDIPADDR (50), 1 occurrence, type IPADDRESS-RESERVED
      code IPADDRLEASETIME (51), 1 occurrence, type LONGINT-RESERVED
      code OPTIONOVERLOAD (52), 1 occurrence, type BYTE-RESERVED
      code MESSAGETYPE (53), 1 occurrence, type BYTE-RESERVED
      code SERVERIDENTIFIER (54), 1 occurrence, type IPADDRESS-RESERVED
      code PARAMREQUESTLIST (55), 1 to 255 occurrences, type BYTE-RESERVED
      code MESSAGE (56), 1 to 255 characters, type STRING-RESERVED
      code MAXDHCPMSGSIZE (57), 1 occurrence, type WORD-RESERVED
      code RENEWALTIME (58), 1 occurrence, type LONGINT
```

```
code REBINDTIME (59), 1 occurrence, type LONGINT
code CLASSIDENTIFIER (60), 1 to 255 occurrences, type BYTE
code CLIENTIDENTIFIER (61), 2 to 255 occurrences, type BYTE
code NOTDEFINED62 (62), 1 to 255 occurrences, type BYTE
code NOTDEFINED63 (63), 1 to 255 occurrences, type BYTE
code NISDOMAIN (64), 1 to 255 characters, type STRING
code NISSERVERS (65), 1 to 63 occurrences, type IPADDRESS
code TFTPSERVERNAME (66), 4 to 255 characters, type STRING
code BOOTFILENAME (67), 1 to 255 characters, type STRING
code MOBILEIPHOMEAGNT (68), 0 to 63 occurrences, type IPADDRESS
code SMTPSERVERS (69), 1 to 63 occurrences, type IPADDRESS
code POP3SERVERS (70), 1 to 63 occurrences, type IPADDRESS
code NNTPSERVERS (71), 1 to 63 occurrences, type IPADDRESS
code WWWSERVERS (72), 1 to 63 occurrences, type IPADDRESS
code FINGERSERVERS (73), 1 to 63 occurrences, type IPADDRESS
code IRCSERVERS (74), 1 to 63 occurrences, type IPADDRESS
code STREETTALKSERVERS (75), 1 to 63 occurrences, type IPADDRESS
code STREETTALKDASRVRS (76), 1 to 63 occurrences, type IPADDRESS
```

Example 2:

The following command lists options starting with the string “ga”:

```
#dhcp list definedoptions ga
code,           number of values,       type of value
code GATEWAY (3), occurrence 1, type IPADDRESS
```

DHCP LIST LEASE

Lists the lease time.

dhcp list lease

Example: `dhcp list lease`

Response:

```
Default lease time ......... 168 hours
```

DHCP SET ADDRESSES

Creates or changes a pool of IP addresses that are associated with a subnetwork.

dhcp set addresses <*first ipaddr*> <*last ipaddr*>

first ipaddr First address in a pool of addresses for a particular subnetwork.

last ipaddr Last address in a pool of addresses for a particular subnetwork.

Example: `dhcp set addresses 192.168.254.1 192.168.254.250`

DHCP SET EXPIRE

This command is used to *manually* change a client lease expiration time to a certain value.

Note 1: Changing a client lease time manually is rarely required.

Note 2: The client information does not get updated. It will still have the old value.

dhcp set expire <*ipaddr*> <*hours*> | *default* | *infinite*

ipaddr	IP address of the client lease in the format of 4 decimals separated by periods.
hours	Lease time; minimum is 1 hour; the global default is 168 hours.
default	Lease time that has been specified at the subnetwork or global level.
infinite	No lease time limit; the lease becomes permanent.

Example: `dhcp set expire 192.168.254.18 8`

DHCP SET LEASE

Controls lease time.

dhcp set lease [<*net*>|<*ipaddr*>]<*hours*>|default|infinite

net	IP address of the subnetwork lease in the format of 4 decimals separated by periods.
ipaddr	IP address of the client lease in the format of 4 decimals separated by periods.
hours	Lease time; minimum is 1 hour; the global default is 168 hours.
default	Lease time that has been specified at the subnetwork or global level.
infinite	No lease time limit; the lease becomes permanent.

Example 1: `dhcp set lease 192.168.254.17 default`
(sets client lease time to default)

Example 2: `dhcp set lease 192.168.254.0 infinite`
(sets lease time to infinite for this subnet)

Example 3: `dhcp set lease 192.168.254.0 infinite`
(sets lease time to infinite for this subnet)

DHCP SET MASK

Used to conveniently change the mask of a DHCP subnet without having to delete and recreate the subnet and all its entries.

dhcp set mask <*net*> <*mask*>

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

mask IP network mask, in the format of 4 decimals separated by periods.

Example: `dhcp set mask 192.168.254.0 255.255.255.0`

DHCP SET OTHERSERVER

This command instructs the router's DHCP server to either continue or stop sending DHCP requests when another DHCP server is detected on the LAN. The default is **stop**.

dhcp set otherserver <*net*> continue|stop

net IP address of the subnetwork lease in the format of 4 decimals separated by periods.

continue The router's DHCP server continues sending DHCP requests, even if another DHCP server is detected on the LAN.

stop The router's DHCP server stops sending DHCP requests when another DHCP server is detected on the LAN.

Example: `dhcp set otherserver 192.168.254.17 stop`

DHCP SET VALUEOPTION

Sets values for global options, options specific to a subnetwork, or options specific to a client lease. For more information, see Setting Option Values, on page 89.

dhcp set valueoption [<*ipaddr*>|<*net*>] <*code*> <*value*>....

ipaddr Specify the client IP address if the option value applies only to the client lease (4 decimals separated by periods).

net Specify the subnetwork IP address if the option value applies only to the subnetwork lease (4 decimals separated by periods).

code Code specifying the option to be set. It can be a number between 1 and 61 or a keyword. Use the command **dhcp list definedoptions** to list the codes and keywords (see DHCP LIST DEFINEDOPTIONS, on page 357.)

value Value to be assigned to the specified option. It could be a byte, word, signed long, unsigned long, binary, IP address, or string depending on the option.

Example 1: This command does not specify an client or subnetwork address, and thus sets a *global* value for the *domainnameserver* option.

```
dhcp set valueoption domainnameserver 192.168.254.2 192.168.254.3
```

Example 2: This command sets the value for the *gateway* option associated with the subnetwork.

```
dhcp set valueoption gateway 192.168.254.0 192.168.254.254
```

Example 3: This command sets a value for the *winserver* option associated with a *specific client.*

```
dhcp set valueoption 192.168.254.251 winserver 192.168.254.7
```

Example 4: This command sets a static route (option 33) to IP address 192.168.253.253 through router 192.168.254.254. (No mask is specified.)

```
dhcp set valueoption 33 192.168.254.254 192.168.253.253
```

L2TP — Virtual Dial-Up Configuration Commands

This section contains L2TP command descriptions. For a complete discussion of L2TP tunneling, see L2TP Tunneling — Virtual Dial-Up, on page 137.

L2TP commands allow you to:

- Add, delete, and modify tunnels
- Configure L2TP router information including:
  - Names
  - Security authentication protocols and passwords
  - Addresses
  - Management of traffic performance
- Restrict a tunnel so it can be established only with a specific remote interface (**l2tp set wanif**).

Note: Two **remote** commands specific to L2TP are included in this section.

L2TP ?

Lists the supported keywords.

l2tp ?

Response:

```
L2tp Sub-commands:
?                add              del
forward          list             set
call             close
```

L2TP ADD

Creates a tunnel entry.

l2tp add *<TunnelName>*

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp add PacingAtWork`

L2TP CALL

This command is primarily used for debugging purposes and it establishes a tunnel without creating a session.

l2tp call <*TunnelName*>

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp call PacingAtWork`

L2TP CLOSE

Closes an L2TP tunnel and/or session.

l2tp close <*L2TP unit number*>|-*n*<*TunnelName*>|-*t*<*tunnelid*>|-*s*<*serialnum*>|-*c*<*callid*>

L2TP unit number

-n TunnelName Name of the tunnel (character string). The name is case sensitive.

-t tunnelid Local tunnel id.

-s serialnum Serial number of the call within the tunnel.

-c callid ID of the local call for the session.

Note: Either <*TunnelName*> or <*tunnelid*> must be specified.

Example: `l2tp close -n PacingAtWork`

L2TP DEL

Deletes a tunnel entry.

l2tp del <*TunnelName*>

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp del PacingAtWork`

L2TP FORWARD

The router can be configured to forward all incoming calls to an LNS without answering the incoming call. This feature is normally used when the router is acting as a LAC or both a LAC and LNS.

Note: Only *one* tunnel entry can have this option set.

l2tp forward all | none <*TunnelName*>

all Forward all incoming calls through the tunnel to an LNS

none No incoming calls are allowed to be forwarded through the tunnel to an LNS

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp forward PacingAtWork`

L2TP LIST

Provides a complete display of the current configuration settings for tunnel(s), except for the authentication password/secret.

l2tp list |<*TunnelName*>|

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp list PacingAtWork`

```
# l2tp list
INFORMATION FOR <pacingAtWork>
  type ..........................       L2TPClient (LAC-will not dial)/LNS
All Incoming Calls Tunneled here .       no
  CHAP challenge issued ..........       yes
  hidden AVPs used ...............       yes
  sequencing/pacing ..............       window pacing
    sequencing/pacing is .........       required
    window size for sequencing/pacing    10
  ip address .....................       10.0.0.1
  Our host name ..................       pacingAtHome

  ACTIVE TUNNEL ..................       UNKNOWN
    current state ................       CLOSED
    LOCAL TUNNEL ID ..............       1
    REMOTE TUNNEL ID .............       0
    remote firmware ..............       0
    remote ip address ............       10.0.0.1
    LAC SESSION serial number ... '      0
      current state ..............       CLOSED
      LOCAL CALL ID ..............       1
        local window size ........       10
        sequencing/pacing ........       WINDOW PACING
          sequencing/pacing is ...       required
      REMOTE CALL ID .............       0
        remote window size .......       0
```

L2TP SET ADDRESS

Used to define the IP address of the other end of the tunnel, either the remote L2TP Access Concentrator (LAC) or remote L2TP Network Server (LNS).

Caution: If the IP address of the remote tunnel is part of a subnet that is also reached through the tunnel, a routing table entry for this address *must* be explicitly added. Normally, this routing entry will be added to remote entry, which has the default route.

Note 1: When a remote router tries to create a tunnel, the remote router's IP address is not authenticated .

Note 2: If this command is not used, then <*ipaddr*> defaults to 0.0.0.0, and this end cannot initiate the tunnel.

l2tp set address <*ipaddr*> <*TunnelName*>

ipaddr IP address of the remote LAC or LNS.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set address 192.168.100.1 PacingAtWork`

L2TP SET AUTHEN

Enables or disables authentication of the remote router during tunnel establishment using the CHAP secret, if it exists. If the remote router tries to authenticate the local end during tunnel authentication, the local router will always attempt to respond, provided a CHAP secret has been configured.

l2tp set authen on | off <*TunnelName*>

on Enables authentication.

off Disables authentication.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set authen PacingAtWork`

L2TP SET CHAPSECRET

Creates a CHAP secret. This CHAP secret is used to authenticate the creation of the tunnel and is used for hiding certain control packet information. The LAC and the LNS can share a single CHAP secret for a given tunnel.

l2tp set CHAPSecret <*secret*> <*TunnelName*>

secret CHAP secret (character string) used to authenticate the creation of the tunnel.

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp set CHAPSecret PacingAtWork`

L2TP SET DIALOUT

Lets the LNS instruct the L2TP client to use an ISDN phone line to place a call on its behalf.

l2tp set dialout yes | no <*TunnelName*>

yes This option lets the router place outgoing calls.

no This option prevents the router from placing outgoing calls.The default is no.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set dialout yes PacingAtWork`

L2TP SET HIDDENAVP

Configures the router to protect some L2TP control information (such as names and passwords for a PPP session) using hidden AVPs. This command is often used to turn off hidden AVPs (no option), in cases where the other end of the tunnel does not support hidden AVPs.

l2tp set hiddenAVP yes | no <*TunnelName*>

yes This option lets the router hide AVPs. The default is yes.

no This option disables hidden AVPs.

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp set hiddenAVP yes PacingAtWork`

L2TP SET OURADDRESS

Specifies the source IP address used when the tunnel is originated.

Use this command when you want to specify a source IP address other than the WAN interface IP address. For example, if NAT (network address translation) is not being used, all IP addresses on the Ethernet LAN would be visible. You could then specify, as the source IP address, the Ethernet IP address of the router (which would be visible) instead of the WAN interface IP address.

l2tp set ouraddress <*ipaddr*> <*TunnelName*>

ipaddr Source IP address used for this tunnel (four decimals separated by periods).

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set ouraddress 192.168.254.254 PacingAtWork`

L2TP SET OURPASSWORD

Specifies the router's secret/password for PPP authentication on a per-tunnel basis.

l2tp set ourpassword <*password*> <*TunnelName*>

password Router's secret/password used for authentication when challenged by another router.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set ourpassword 7z8x9q0d6j1t3k PacingAtWork`

L2TP SET OURSYSNAME

Specifies the router's name for PPP authentication on a per-tunnel basis.

l2tp set oursysname <*name*> <*TunnelName*>

name Name of the router that is used for authentication when challenged by another router.

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp set oursysname myName PacingAtWork`

L2TP SET OURTUNNELNAME

This command creates local router's host name.

Note: If this command is not used, then, if it has been specified, the <*name*> from the **l2tp set ourSysName** command or the <*name*> from the command **system name** <*name*> is used.

l2tp set ourTunnelName <*name*> <*TunnelName*>

name Host name of the local router. This is the fully qualified domain name of the local router.

The name is case-sensitive

TunnelName Name of the tunnel (character string). The name is case sensitive.

Example: `l2tp set ourTunnelName isp PacingAtWork`

L2TP SET REMOTENAME

Creates the host name of the remote tunnel.

Note: If this command is not used, then <*TunnelName*> of the tunnel entry is used.

l2tp set remoteName *<name> <TunnelName>*

name Host name of the remote tunnel. This is the fully qualified domain name of the remote host.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set remoteName isp PacingAtWork`

L2TP SET TYPE

Defines the type of L2TP support for the tunnel. The router's role is defined on a per-tunnel basis.

l2tp set type all|lac|lns|l2tpclient|disabled *<TunnelName>*

all The router is configured to act as both a LAC/L2TP client and an LNS server.

lac The router is configured to act as a LAC for this tunnel.

lns The router is configured to act as an LNS for this tunnel.

l2tpclient The router is configured to act as an L2TP client for this tunnel.

disabled The tunnel entry is disabled.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Example: `l2tp set type l2tpclient PacingAtWork`

L2TP SET WANIF

Restricts the remote interface with which the L2TP tunnel can be established.

If this command is not used, no remote interface restriction is enforced. For example, no restriction would be enforced when the Dial Backup feature is used (see page 109). Thus, the tunnel would be terminated and re-established when switching back and forth between the primary interface and the backup interface. If the tunnel is to established *only* with the primary interface or *only* with the backup interface, you must specify that restriction with this command.

To clear the remote restriction for a tunnel, enter the **l2tp set wanif** command with a hyphen (**-**) as the remote name.

l2tp set wanif *<remote> <TunnelName>*

remote Name of the remote router profile that must be used when establishing the L2TP tunnel. To list the remote routers, use the command **remote list**.

For the dual-Ethernet router, specify the physical interface name, that is, either ETHERNET/0 or ETHERNET/1.

To clear the remote restriction for a tunnel, enter a hyphen (**-**) as the remote name.

TunnelName Name of the tunnel (character string). The name is case-sensitive.

Examples:

This command restricts the tunnel named OfficeTunnel to the remote interface named officertr.

```
l2tp set wanif officertr OfficeTunnel
```

This command clears the remote interface restriction for the tunnel named OfficeTunnel.

```
l2tp set wanif - OfficeTunnel
```

This command restricts the tunnel named OfficeTunnel to the physical interface ETHERNET/1.

```
l2tp set wanif ETHERNET/1 OfficeTunnel
```

L2TP SET WINDOW

Enhances traffic performance in a tunneling environment. The command's options affect the way incoming payload packets are processed. The router is configured with the following default options: sequencing, required, and size 10.

l2tp set window sequencing|pacing|nosequencing|optional|required|size <*TunnelName*>

sequencing Sequence numbers are placed in the L2TP payload packets. With this option, one end instructs the other end to send sequence packets. No acknowledgments are issued for received packets.

pacing Sequence numbers are placed in the L2TP payload packets. When a session is created, the router specifies a window size. Acknowledgments for received packets are issued.

nosequencing No sequence numbers are placed in the L2TP payload packets carrying the PPP packets. If the remote end carries out sequencing or pacing, the router can still send and receive sequenced packets.

optional Allows dynamic switching of a session from pacing or sequencing to nosequencing.

required Disables dynamic switching from pacing or sequencing to nosequencing.

size Controls the size of the receive window for receiving packets for sequencing or pacing, when a session is created. Size can be 0 for packet sequencing. Must be a non-zero value for window pacing. Size must be less than or equal to 30.

TunnelName Name of the tunnel (character string). This name is case-sensitive.

Example: `l2tp set window sequencing PacingAtWork`

REMOTE SETL2TPCLIENT

With this command, this remote is the path to the L2TP client and accepts tunnel calls. Use this command if your router acts as an LNS. You must also specify PPP authentication and IP routes for this remote.

remote setl2tpclient <*TunnelName*><*remoteName*>

TunnelName Name of the tunnel (character string) associated with the remote LAC. The name is case-sensitive.

remoteName Name of the remote entry (character string). The name is case sensitive.

Example: `remote setl2tpclient PacingAtWork Router2`

REMOTE SETLNS

With this command, this remote is the path to the LNS, and it will forward the incoming call (which matches this remote entry) through the tunnel named <*TunnelName*> if your router is the client.

Note: The remote entry must also have appropriate information such as PPP authentication, IP routing, IPX routing, bridging, or Caller ID.

remote setLNS <*TunnelName*><*remoteName*>

TunnelName Name of the tunnel (character string). The name is case-sensitive.

RemoteName Name of the remote entry (character string).

Example: `remote setLNS PacingAtWork lnsServer`

Bridge Filtering Commands

Bridge filtering allows you to control the packets transferred across the router. This feature can be used to enhance security or improve performance. Filtering is based on matched patterns within the packet at a specified offset. Two filtering modes are available.

- Deny mode will discard any packet that matches the deny filter database and let all other packets pass.
- Allow mode will only pass the packets that match the allow filter database and discard all others.

Up to 40 deny and 40 allow filters can be activated from the filter database.

FILTER BR ?

Lists the supported keywords.

filter br ?

Response:

```
Bridge filter commands:
?                    add                  del
use                  list
```

FILTER BR ADD

Adds a bridging filter to the filtering database. The filter can allow or deny the forwarding of packets based on the contents of the packets. The command specifies the position within the packet that is checked and the data that must appear in that location in order for the packet to match this filter.

filter br add [*pos*] [*data*] allow | deny

pos Byte offset within a packet; number from 0-127.

data Hexadecimal number up to 6 bytes.

Example:

This filter prevents forwarding of RARP packets across the bridge. The data at byte offset 12 in each packet is checked and, if the data is hex 8035, the packet is denied forwarding.

```
filter br add 12 8035 deny
```

FILTER BR DEL

Deletes a bridging filter from the filtering database. The parameters on the command identify the filter to be deleted.

filter br del [*pos*] [*data*] allow | deny

pos Byte offset within a packet; number from 0-127.

data Hexadecimal number up to 6 bytes.

Example:

This command deletes the filter which denies the forwarding of packets that have the hex value 8035 at byte offset 12.

```
filter br del 12 8035 deny
```

FILTER BR LIST

Lists the bridging filters in the filtering database.

filter br list

Example: `filter br list`

Response:

```
Allow Filter:

Deny Filter:
pos:12, len=2, <80><35>
```

FILTER BR USE

Sets the mode of filtering to either deny, allow, or none.

filter br use none | deny | allow

Example: `filter br use allow`

PPPoE Commands

This section contains the commands that are specific to PPPoE (PPP over Ethernet). To learn more about PPPoE configuration and management, see page 103.

The commands in this section are:

remote setPPPoEservice	Used when configuring a PPPoE client entry.
pppoe close	Ends a PPPoE session.
pppoe list	Lists information about PPPoE sessions.

REMOTE SETPPPOESERVICE

Defines the remote router entry as a PPPoE remote entry. It also specifies the service to which PPPoE users connect through this remote entry.

Note: Enter this command immediately after the **remote add** command that defines the remote router entry.

remote setPPPoEservice <*service*> | ***** | **-** <*remoteName*>

service Name of the PPPoE service to which this remote connects PPPoE users. The service provider defines the name of its service. The name is case-sensitive.

Specify ***** if the remote can be used to connect to any PPPoE service.

Specify **-** to clear the setting.

remoteName Name of the remote router entry (string).

Example:

The following commands define the remote router used to connect to the PPPoE service DialUpPPP.net. Note that the **remote setPPPoEservice** command is entered immediately after the **remote add** command.

```
remote add PPPoEremote
remote setPPPoEservice DialUpPPP.net PPPoEremote
```

PPPOE CLOSE

Closes a currently active PPPoE session. To see the currently active PPPoE sessions, enter **pppoe list**.

pppoe close <ifsnumber>

ifsnumber Session to be closed. Specify the PPPoE/Ifs number for the session as shown in the **ifs** or **pppoe list** command output.

Example:

The first command lists information about a PPPoE session, including its Ifs number (1); the second command closes that session.

```
# pppoe list
PPPoE Client Session ...... DialUpPPP.net
      PPPoE/Ifs number..... 1
      Access Concentrator.. 15021109931568-efficient
      Peer MAC Address .... 00:10:67:00:66:E2
      Session ID .......... 2
      State ............... 2
      Flags ............... 1
# pppoe close 1
```

PPPOE LIST

Lists information about the currently active PPPoE sessions.

pppoe list

Output Fields:

PPPoE Client Session . . . . . Service name.

PPPoE/Ifs number. . . . . Number identifying the session. It is used on the **pppoe close** command.

Access Concentrator. . . Identification of the access concentrator to which the session is connected.

Peer MAC Address. . . . MAC address to which the session is connected.

Session ID. . . . . Identification of the session by the access concentrator. The service provider needs this ID to find the access concentrator log for the session.

State . . . . . Internal value indicating the state of the session:

- 0 Initial state.
- 1 Session has finished.
- 2 Session has been established and can send and receive user data.
- 3 Reserved.
- 4 Session is attempting to discover an access concentrator to provide the requested service.
- 5 Reserved.
- 6 Session has selected an access concentrator to provide the service and is waiting for it to set up the session.

Flags . . . . . Internal flag. Currently, the only defined value is 1, indicating that this is a client PPPoE session.

Example:

```
# pppoe list
PPPoE Client Session ...... DialUpPPP.net
      PPPoE/Ifs number..... 1
      Access Concentrator.. 15021109931568-efficient
      Peer MAC Address .... 00:10:67:00:66:E2
      Session ID .......... 2
```

```
State ............... 2
Flags ............... 1
```

IKE (Internet Key Exchange) Commands

The IKE software option and the IKE commands are described in IPSec (Internet Protocol Security), on page 149. Additional IKE debug commands are listed in IKE Debug Commands, on page 208.

IKE COMMIT

Determines whether the IKE commit bit is set. By default, the commit bit is not set (**off**). To see the current setting, enter **ike commit** with no parameter.

If packets are not being processed correctly across an IPSec tunnel, try **ike commit on** so that the commit bit is set. Setting the commit bit makes sure that no IPSec traffic arrives at the router before the router is ready for it.

For more information about IKE, see IPSec (Internet Protocol Security), on page 149.

ike commit [on | off | help]

on Commit bit is set.

off Commit bit is not set. The default value is off.

help Displays help message.

Example: `ike commit on`

IKE FLUSH

Clears all IKE configuration information from the router. For more information about IKE, see IPSec (Internet Protocol Security), on page 149.

ike flush

IKE IPSEC POLICIES ADD

Defines the name of an IPsec policy to be used for filtering. Other IPSec Policy commands define the filtering parameters (see IKE IPSec Policy Commands, on page 157).

ike ipsec policies add <*PolicyName*>

PolicyName New name for an IPsec policy. To see the policy names in use, use the **ike ipsec policies list** command.

Example: `ike ipsec policies add mypolicy`

IKE IPSEC POLICIES DELETE

Deletes an existing IPSec policy. To define IPSec Policies, see IKE IPSec Policy Commands, on page 157.

ike ipsec policies delete <*PolicyName*>

PolicyName Name of an existing IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies delete mypolicy`

IKE IPSEC POLICIES DISABLE

Disables an IPSec policy. The policy can be re-enabled using the **enable** command (see IKE IPSec Policy Commands, on page 157).

ike ipsec policies disable <*PolicyName*>

PolicyName Name of an existing IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies disable mypolicy`

IKE IPSEC POLICIES ENABLE

Enables an IPSec policy. An enable command is required for each new policy; the enable command indicates that the specification of the policy is complete and the policy is ready to be used. The enable command can also be used to re-enable a disabled policy. For more information, see IKE IPSec Policy Commands, on page 157.

ike ipsec policies enable <*PolicyName*>

PolicyName Name of the IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies enable mypolicy`

IKE IPSEC POLICIES LIST

Lists the IPSec policies. For more information, see IKE IPSec Policy Commands, on page 157.

ike ipsec policies list

Example:

```
# ike ipsec policies list
IKE IPSec policies:
mypolicy (enabled)
  Source address/mask: 192.168.16.0/255.255.255.0
  Destination address/mask: 192.168.23.0/255.255.255.0
```

```
Protocol: *
Source port: *
Destination port: *
Tunnel mode
Peer: my_aggressive_peer (0.0.0.0)
Proposals: myproposal
```

IKE IPSEC POLICIES SET DEST

Defines a *destination* filtering parameter value for the policy. The destination parameter requires that the data be intended for the specified destination IP address and mask. The destination is the device or network that finally receives the packet, not the router that routes the packet.

ike ipsec policies set dest <*IPaddress*> <*IPmask*> <*PolicyName*>

IPaddress IP address allowed to be the destination of the data (4 decimals separated by periods).

IPmask IP network mask (4 decimals separated by periods).

PolicyName Name of the IPsec policy to which the destination parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set dest 192.168.16.0 255.255.255.0 mypolicy`

IKE IPSEC POLICIES SET DESTPORT

Defines a *destination port* filtering parameter value for the policy. The destination port parameter requires a specific destination port for the data or allows any destination port (*). (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

ike ipsec policies set destport <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

PortNumber Destination port whose data is allowed by the policy. The port can be specified by one of the listed names or by its number. To allow data through for any destination port, specify an asterisk (*).
TELNET
HTTP
SMTP
TFTP
*

PolicyName Name of the IPsec policy to which the destination port parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Examples: `ike ipsec policies set destport * mypolicy`

`ike ipsec policies set destport http webpolicy`

IKE IPSEC POLICIES SET INTERFACE

Defines an *interface* filtering parameter value for the policy. The policy is only used when the specified interface is connected. For example, if the policy is to be used only when the Dial Backup remote is connected, you would specify the remote name as the interface for the policy. (To read about Dial Backup, see page 109.)

Note: The specified interface must be the interface to the IKE peer.

Otherwise, if the policy can be used regardless of the connected interface, specify the string **none**.

This command is intended to allow the user to choose when to apply IPSec/IKE filters and incur the resulting encryption and authentication costs. With this command, you can limit a policy to a specific interface.

ike ipsec policies set interface <*interface* | all > <*PolicyName*>

interface Interface that must be connected when the policy is used. This is usually referenced by a remote name, although it could be another interface such as “ethernet/0”. If no interface restriction is to be set for this policy, specify the string **all**.

PolicyName Name of the IPsec policy to which the interface parameter value is added. To see the policies, use the **ike ipsec policies list** command.

Examples:

This command requires that, when the remote interface **backup** comes up, IKE is enabled for packets described by policy **corporate**. The specified interface (**backup**) must be the interface to the IKE peer.

```
ike ipsec policies set interface backup corporate
```

This command specifies that IKE is enabled for packets described by policy **mypolicy** regardless of the interface the peer is on.

```
ike ipsec policies set interface all mypolicy
```

IKE IPSEC POLICIES SET MODE

Defines the *mode* filtering parameter value for the policy. The mode parameter specifies the encapsulation mode (tunnel or transport) that may be used for the connection (see Transport and Tunnel Encapsulation Modes, on page 149). If no value is set for the mode parameter, tunnel mode is assumed.

ike ipsec policies set mode <TUNNEL | TRANSPORT> <*PolicyName*>

TUNNEL
TRANSPORT Encapsulation method required for the connection. The default is TUNNEL.

PolicyName Name of the IPsec policy to which the encapsulation mode parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set mode transport rtr2rtrpolicy`

IKE IPSEC POLICIES SET PEER

Defines a *peer* filtering parameter value for the policy. The peer parameter specifies an IKE peer that may be used for the connection. (The peer must have been defined by IKE peer commands; see IKE Peer Commands, on page 154.)

ike ipsec policies set peer <*PeerName*> <*PolicyName*>

PeerName	Name of an IKE peer. To see the IKE peer names, use the **ike peers list** command.
PolicyName	Name of the IPsec policy to which the peer parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set peer my_aggressive_peer mypolicy`

IKE IPSEC POLICIES SET PFS

Defines the *pfs* filtering parameter value for the policy. The pfs parameter specifies the Perfect Forward Secrecy negotiation used for the connection.

If you specify **1** or **2**, Perfect Forward Secrecy is performed using the specified Diffie-Hellman group (1 or 2). If you specify **none**, then Perfect Forward Secrecy is not required for this connection and no Diffie-Hellman group is used to encrypt the keys during rekey. To read more about PFS, see IKE Management, on page 151

ike ipsec policies set pfs <1 | 2 | none > <*PolicyName*>

ike ipsec policies set pfs <1 | 2 | none > <*PolicyName*>

1	Use Diffie-Hellman group 1 for the Perfect Forward Secrecy negotiation.
2	Use Diffie-Hellman group 2 for the Perfect Forward Secrecy negotiation.
none	Perfect Forward Secrecy negotiation is not required for this connection.
PolicyName	Name of the IPsec policy to which the pfs parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set pfs 2 mypolicy`

IKE IPSEC POLICIES SET PROPOSAL

Defines a *proposal* filtering parameter value for the policy. The proposal parameter specifies an IKE IPSec proposal that may be used for the connection. (It must have been defined by IKE IPSec proposal commands; see IKE IPSec Proposal Commands, on page 156.)

Unlike the other filtering parameters, the policy may allow more than one value for the *proposal* parameter. For example, two **set proposal** commands could specify two proposals, either of which could be used by the connection. See IKE IPSec Policy Commands, on page 157.

ike ipsec policies set proposal <*ProposalName*> <*PolicyName*>

ProposalName Name of an IKE proposal. To see the IKE proposal names, use the **ike proposals list** command.

PolicyName Name of the IPsec policy to which the proposal parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set proposal myproposal mypolicy`

IKE IPSEC POLICIES SET PROTOCOL

Defines a *protocol* filtering parameter value for the policy. The protocol parameter requires a specific protocol that must be used or allows any protocol (*).

ike ipsec policies set protocol <*ProtocolNumber* | TCP | UDP | *> <*PolicyName*>

ProtocolNumber Protocol required by the policy. The protocol can be specified by number or by name (TCP or UDP). To allow data through for any protocol, specify an asterisk (*).
TCP
UDP
*

PolicyName Name of the IPsec policy to which the protocol parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set protocol * mypolicy`

`ike ipsec policies set protocol tcp webpolicy`

IKE IPSEC POLICIES SET SOURCE

Defines a *source* filtering parameter value for the policy. The source parameter requires the data come from the specified source IP address and mask. The source is the device or network that sent the packet, not the router that routes the packet.

ike ipsec policies set source <*IPaddress*> <*IPmask*> <*PolicyName*>

IPaddress IP address allowed to be the source of the data (4 decimals separated by periods).

IPmask IP network mask (4 decimals separated by periods).

PolicyName Name of the IPsec policy to which the source parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example: `ike ipsec policies set source 192.168.16.0 255.255.255.0 mypolicy`

IKE IPSEC POLICIES SET SOURCEPORT

Defines a *source port* filtering parameter value for the policy. The source port parameter requires a specific source port for the data or allows any source port (*) (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

ike ipsec policies set sourceport <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

PortNumber TELNET HTTP SMTP TFTP *	Source port whose data is allowed by the policy. The port can be specified by one of the listed names or by its number. To allow data through for any source port, specify an asterisk (*).
PolicyName	Name of the IPsec policy to which the source port parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Examples:

```
ike ipsec policies set sourceport * mypolicy

ike ipsec policies set sourceport http webpolicy
```

IKE IPSEC POLICIES SET TRANSLATE

Defines a *translate* filtering parameter value for the policy. The *translate* option determines whether the router applies NAT (network address translation) before the packets are encrypted by IPSec.

Note: The remote must have IP address translation enabled (see NAT on page 95 and the **remote setIpTranslate** command on page 313).

Note: The address that NAT translates to should be the source or destination address for the policy (use the **set source** or **set dest** commands).

Use this option when several remote sites have the same IP subnet, making it impossible to tunnel those sites unchanged to the corporate network.

When the router's public IP address is not the desired choice for the network address translation, you can define a virtual Ethernet interface. A virtual Ethernet interface can be created to translate to an arbitrary IP address (see IP Subnets, on page 79). Again, be sure that the virtual Ethernet interface has IP address translation enabled (**eth ip translate**, page 277), and use the virtual Ethernet interface as the gateway to the other end of the protected network. (See the example below.) You can use the **eth ip addhostmapping** command (page 263) to map a range of NAT addresses to private addresses so the IKE tunnel can be initiated from either end.

ike ipsec policies set translate on | off <*PolicyName*>

on | off	Sets the translate option on or off. If translate is set to **on**, translation is applied before encryption, and the packets are sent using the host router's public IP address.
PolicyName	Name of the IPsec policy to which the source port parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

Example:

The following commands suggest how a virtual interface could be defined for use with Network Address Translation and an IPSec tunnel.

```
# The address of the corporate LAN is 192.168.0.0, but the desired
# NAT address is 10.0.0.1 so you create a virtual interface (0:99),
# turn off RIP for the interface, and assign it the address 10.0.0.1/24.
eth add 0:99
eth ip opt txrip off 0:99
eth ip opt rxrip off 0:99
```

```
eth ip addr 10.0.0.1 255.255.255.0 0:99
#
# Next, enable NAT for the virtual interface and route traffic to the
# the corporate backbone (192.168.0.0/16) through the virtual interface.
eth ip translate on 0:99
eth ip addroute 192.168.0.0 255.255.0.0 10.0.0.0.1 0:99


# Later, when you set up the IKE tunnel, include these commands
# when defining a policy. (The policy name is corporate.)
# The source address must be the virtual interface address.
# The destination address must be the corporate backbone address.
# ike ipsec policies set source 10.0.0.1 255.255.255.255 corporate
# ike ipsec policies set dest 192.168.0.0 255.255.0.0 corporate
# ike ipsec policies set translate on corporate
```

IKE IPSEC PROPOSALS ADD

Defines the name of an IKE IPSec proposal. The proposal commands define the proposals exchanged to set up an IPSec security association (SA), that is, an SA to be used for the user data transfer. See IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals add *<ProposalName>*

ProposalName New name for an IPsec proposal. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals add myproposal`

IKE IPSEC PROPOSALS DELETE

Deletes an existing IKE IPSec proposal. For more information, see IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals delete *<ProposalName>*

ProposalName Name of the IPsec proposal to be deleted. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals delete myproposal`

IKE IPSEC PROPOSALS LIST

Lists the IKE IPSec proposals. For more information, see IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals list

Example:

```
# ike ipsec proposals list
IKE IPSEC PROPOSALS:
myproposal
  ESP encryption: 3DES
  ESP authentication: SHA1
  IPComp: None
  Lifetime 600
  Lifedata 50000
```

IKE IPSEC PROPOSALS SET AHAUTH

Sets the proposal parameter that determines whether AH message authentication is requested and, if it is requested, the hash algorithm used.

Note: The proposal must select either the AH or ESP encapsulation methods. It cannot request AH authentication if it requests ESP encryption and/or ESP authentication.

For more information, see ESP and AH Security Protocols, on page 150 or IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set ahauth <MD5 | SHA1 | NONE> <*ProposalName*>

One of the following:

MD5 Use AH encapsulation and authenticate using hash algorithm Message Digest 5.

SHA1 Use AH encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE No AH encapsulation and no AH message authentication. (If you select this option, ESP encapsulation must be requested by a **set espenc** or **set espauth** command.)

ProposalName Name of the IPsec proposal to which the AH authentication parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set ahauth sha1 myproposal`

IKE IPSEC PROPOSALS SET ESPAUTH

Sets the proposal parameter that determines whether ESP message authentication is requested and, if it is requested, the hash algorithm used.

For more information, see ESP and AH Security Protocols, on page 150 or IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set espauth <MD5 | SHA1 | NONE> <*ProposalName*>

One of the following:

MD5 Use ESP encapsulation and authenticate using hash algorithm Message Digest 5.

SHA1 Use ESP encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE No ESP encapsulation and no ESP message authentication. (If you select this option, the

encapsulation method must be requested by a **set espenc** or **set ahauth** command.)

ProposalName Name of the IPsec proposal to which the ESP authentication parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set espauth sha1 myproposal`

IKE IPSEC PROPOSALS SET ESPENC

Sets the proposal parameter that determines whether ESP encryption is requested and, if it is requested, the encryption method used.

For more information, see ESP and AH Security Protocols, on page 150 or IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set espenc <DES | 3DES | NULL | NONE> <*ProposalName*>

One of the following:

DES Use ESP encapsulation and 56-bit encryption

3DES Use ESP encapsulation and 168-bit encryption (if 3DES is enabled in the router; see Software Option Keys, on page 124.)

NULL No encryption, but use ESP encapsulation. Headers are inserted as though the data was encrypted. This allows verification of the source, but sends the data in the clear, increasing throughput.

NONE No encryption and no ESP encapsulation. (If you select this option, the encapsulation method must be requested by a **set espauth** or **set ahauth** command.)

ProposalName Name of the IPsec proposal to which the ESP encryption parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set espenc 3des myproposal`

IKE IPSEC PROPOSALS SET IPCOMP

Sets the proposal parameter that requests either no compression or LZS compression. For more information, see IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set ipcomp <NONE | LZS> <*ProposalName*>

ike ipsec proposals set > <*ProposalName*>

One of the following:

NONE No compression.

LZS Compress using the LZS algorithm.

ProposalName Name of the IPsec proposal to which the IP compression parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set ipcomp none myproposal`

IKE IPSEC PROPOSALS SET LIFEDATA

Sets the proposal parameter that specifies the maximum number of kilobytes for the IPSec SA; 0 means unlimited. After the maximum data is transferred, IKE renegotiates the connection. By limiting the amount of data that can be transferred, you reduce the likelihood of the key being broken.

For more information on proposal parameters, see IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set lifedata <*kbytes*> <*ProposalName*>

kbytes Maximum number of kilobytes transferred before renegotiation; 0 means unlimited.

ProposalName Name of the IPsec proposal to which the lifedata parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set lifedata 50000 myproposal`

IKE IPSEC PROPOSALS SET LIFETIME

Sets the proposal parameter that specifies the length of time (in seconds) before the IPSec SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

For more information on proposal parameters, see IKE IPSec Proposal Commands, on page 156.

ike ipsec proposals set lifetime <*seconds*> <*ProposalName*>

seconds Maximum number of seconds before renegotiation; 0 means unlimited.

ProposalName Name of the IPsec proposal to which the lifetime parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

Example: `ike ipsec proposals set lifetime 600 myproposal`

IKE PEERS ADD

Defines the name of a new IKE peer. Other commands specify the address, secret, and mode of the peer connection; see IKE Peer Commands, on page 154.

ike peers add <*PeerName*>

PeerName New name for an IKE peer. To see the peer names in use, use the **ike peers list** command.

Example: `ike peers add my_aggressive_peer`

IKE PEERS DELETE

Deletes an existing IKE peer entry. For more information, see IKE Peer Commands, on page 154.

ike peers delete <*PeerName*>

PeerName Name of the IKE peer to be deleted. To see the peer names in use, use the **ike peers list** command.

Example: `ike peers delete my_aggressive_peer`

IKE PEERS LIST

Lists the defined IKE peers. For more information, see IKE Peer Commands, on page 154.

ike peers list

Example:

```
# ike peers list
IKE Peers:
IKE Peers:
my_aggressive_peer
  IP address = 0.0.0.0
  preshared secret = "confidential_hushhush"
  aggressive, peer id = example.efficient.com (Domain name)
            local id = test.efficient.com (Domain name)
my_main_peer
  IP address = 1.2.3.4
  preshared secret = "Shipsailsatmidnight"
  main mode
```

IKE PEERS SET ADDRESS

Sets the IP address of the other endpoint of the secure IKE peer connection. The address specified depends on the mode of the peer connection, which can be either main mode or aggressive mode. (See IKE Management, on page 151.)

If the mode is main mode, the other endpoint of the peer connection is constant, and you specify its IP address.

If the mode is aggressive mode, one end of the connection, the gateway, has a fixed IP address. The other end, the client, has a changing address. When configuring the client, set the peer IP address to the fixed gateway address. When configuring the gateway for an aggressive mode connection, set the peer IP address to **0.0.0.0**.

ike peers set address *<IPaddress> <PeerName>*

IPaddress IP address (4 decimals separated by periods). When configuring a gateway for an aggressive mode connection, set the IP address to **0.0.0.0**.

PeerName Name of the IKE peer whose address is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set address 0.0.0.0 my_aggressive_peer`

IKE PEERS SET LOCALID

Sets the local ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The local ID must match the peer ID on the other end of the connection. The local ID can be an IP address, domain name, or e-mail address as specified by the **set localidtype** command. For more information, see IKE Peer Commands, on page 154.

ike peers set localid <*AggressiveModeID*> <*PeerName*>

AggressiveModeID IP address (4 decimals separated by periods), domain name, or e-mail address.

PeerName Name of the IKE peer whose local ID is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set localid test.efficient.com my_aggressive_peer`

IKE PEERS SET LOCALIDTYPE

Sets the type of the local ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The local ID type must match the peer ID type on the other end of the connection. The possible ID types are IP address, domain name, or e-mail address. For more information, see IKE Peer Commands, on page 154.

ike peers set localidtype <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

One of the following:

IPADDR The local ID must be an IP address.
DOMAINNAME The local ID must be a domain name.
EMAIL The local ID must be an e-mail address.

PeerName Name of the IKE peer whose local ID type is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set localidtype domainname my_aggressive_peer`

IKE PEERS SET MODE

Sets the IKE peer connection mode to either main mode or aggressive mode. Main mode is used when the IP addresses of both ends are known and constant. Aggressive mode is used when the address of one end can change, as with a typical modem or DSL connection. (See Main Mode and Aggressive Mode, on page 152.)

ike peers set mode <MAIN | AGGRESSIVE> <*PeerName*>

One of the following:

MAIN Selects main mode (both ends constant).
AGGRESSIVE Selects aggressive mode (one end can change).

PeerName Name of the IKE peer whose mode is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set mode aggressive my_aggressive_peer`

IKE PEERS SET PEERID

Sets the peer ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The peer ID must match the local ID on the other end of the connection. The peer ID can be an IP address, domain name, or e-mail address as specified by the **set peeridtype** command. For more information, see IKE Peer Commands, on page 154.

ike peers set peerid <*AggressiveModeID*> <*PeerName*>

AggressiveModeID IP address (4 decimals separated by periods), domain name, or e-mail address.

PeerName Name of the IKE peer whose peer ID is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set peerid example.efficient.com my_aggressive_peer`

IKE PEERS SET PEERIDTYPE

Sets the type of the peer ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The peer ID type must match the local ID type on the other end of the connection. The possible ID types are IP address, domain name, or e-mail address. For more information, see IKE Peer Commands, on page 154.

ike peers set peeridtype <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

One of the following:

IPADDR The peer ID must be an IP address.
DOMAINNAME The peer ID must be a domain name.
EMAIL The peer ID must be an e-mail address.

PeerName Name of the IKE peer whose peer ID type is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set peeridtype domainname my_aggressive_peer`

IKE PEERS SET SECRET

Sets the shared secret for the IKE peer connection. The secret must be identical for both ends. For more information, see IKE Peer Commands, on page 154.

ike peers set secret <*secret*> <*PeerName*>

secret Secret (up to 256 characters; do not use spaces or non-printable characters).

PeerName Name of the IKE peer whose secret is specified. To see the peer names, use the **ike peers list** command.

Example: `ike peers set secret confidential_hushhush my_aggressive_peer`

IKE PROPOSALS ADD

Defines the name of a new IKE proposal. The IKE proposal commands define the proposals exchanged during the Phase 1 SA. For more information, see IKE Management, on page 151.

ike proposals add <*ProposalName*>

ProposalName Name for the new IKE proposal. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals add my_ike_proposal`

IKE PROPOSALS DELETE

Deletes an existing IKE proposal. See IKE Proposal Commands, on page 155.

ike proposals delete <*ProposalName*>

ProposalName Name of the IKE proposal to be deleted. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals delete my_ike_proposal`

IKE PROPOSALS LIST

Lists the IKE proposals. See IKE Proposal Commands, on page 155.

ike proposals list

Example:

```
# ike proposals list
IKE proposals:
my_ike_proposal
  Session authentication: Preshared key
  Encryption: DES
  Message authentication: MD5
  DH Group 2
  Lifetime 86400
  Lifedata 0
```

IKE PROPOSALS SET DH_GROUP

Sets the IKE proposal parameter that specifies the Diffie-Hellman (DH) key generation group used (no group or group 1 or 2). See IKE Proposal Commands, on page 155.

ike proposals set dh_group <NONE | 1 | 2 > <*ProposalName*>

One of the following:

NONE	No DH group is used.
1	Use DH group 1.
2	Use DH group 2.

ProposalName Name of the IKE proposal to which the DH group parameter is added. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals set dh_group 2 my_ike_proposal`

IKE PROPOSALS SET ENCRYPTION

Sets the IKE proposal parameter that requests *ESP encryption* and specifies the encryption method used. (See IKE Proposal Commands, on page 155.)

ike proposals set encryption <DES | 3DES > <*ProposalName*>

One of the following:

DES Use DES (56-bit) encryption.

3DES Use 3DES (168-bit) encryption (if 3DES is enabled in the router; see Software Option Keys, on page 124).

ProposalName Name of the IKE proposal to which the encryption parameter is added. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals set encryption des my_ike_proposal`

IKE PROPOSALS SET LIFETIME

Sets the IKE proposal parameter that specifies the length of time (in seconds) before the Phase 1 SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection. See IKE Management, on page 151.

ike proposals set lifetime <*seconds*> <*ProposalName*>

seconds Maximum number of seconds before renegotiation; 0 means unlimited.

ProposalName Name of the IKE proposal to which the lifetime parameter is added. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals set lifetime 86400 my_ike_proposal`

IKE PROPOSALS SET MESSAGE_AUTH

Sets the IKE proposal parameter that specifies the message authentication done. It can propose no message authentication, or it can propose authentication using the hash algorithm Message Digest 5 (MD5) or Secure Hash Algorithm-1 (SHA1).

ike proposals set message_auth <NONE | MD5 | SHA1> <*ProposalName*>

One of the following:

NONE	No authentication.
MD5	Authenticate using the MD5 algorithm.
SHA1	Authenticate using the SHA1 algorithm.

ProposalName Name of the IKE proposal to which the message authentication parameter is added. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals set message_auth md5 my_ike_proposal`

IKE PROPOSALS SET SESSION_AUTH

Sets the IKE proposal parameter that specifies the session authentication; preshared key is currently the only option. For more information on IKE proposals, see IKE Management, on page 151.

ike proposals set session_auth <PRESHARE> <*ProposalName*>

PRESHARE Preshared key.

ProposalName Name of the IKE proposal to which the session authentication parameter is added. To see the proposal names in use, use the **ike proposals list** command.

Example: `ike proposals set session_auth preshare my_ike_proposal`

IPSec Commands

The following commands allow you to define an IPSec connection without IKE. To read about IPSec Security, see IPSec (Internet Protocol Security), on page 149.

Note: If you define a tunnel using IPSec commands, the keys will remain static. This could pose a security risk and is not recommended. Use of IKE for key management is recommended.

IPSEC ADD

Defines an IPSec security association (SA) name.

ipsec add <*SAname*>

SAname Name for the new IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec add show_rx`

IPSEC DEL

Deletes an existing IPSec security association (SA) name.

ipsec del <*SAname*>

SAname Name for the IPSec SA to be deleted.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec del show_rx`

IPSEC DISABLE

Disables a defined IPSec SA entry.

ipsec disable <*SAname*>

SAname Name for the IPSec SA to be disabled.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec disable show_rx`

IPSEC ENABLE

Enables a defined IPSec SA entry, indicating it is complete and ready to be used. The command can also re-enable a disabled SA entry.

ipsec enable <*SAname*>

SAname Name for the IPSec SA to be enabled.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec enable show_rx`

IPSEC FLUSH

Clears all IPSec definitions.

ipsec flush

IPSEC LIST

Lists one or all IPSec SA entries.

ipsec list [<*SAname*>]

SAname Optional name for the IPSec SA to be listed.

Example:

```
# ipsec list
IPSec security associations:
```

```
show_rx
  Gateway: 207.135.89.233
  Inbound
  Tunnel
  BOTH
  3DES
   key=111111111122222222223333333333444444444455555555
  SHA1
   key=aaaaaaaaaabbbbbbbbbbccccccccccddddddddddd (20)
  No compression
  id =424242
  seq=1, bitmap=ffffffff
show_tx
  Gateway: 207.135.89.233
  Outbound
  Tunnel
  BOTH
  3DES
   key=012345678901234567890123456789012345678901234567
  SHA1
   key=abcdefabcdefabcdefabcdefabcdefabcdefabcd (20)
No compression
id =123456
seq=6734
```

IPSEC SET AUTHENTICATION

Selects authentication for the IPSec SA using either SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5).

ipsec set authentication <MD5 | SHA1> <*SAname*>

One of the following:

MD5	Authenticate using the MD5 algorithm.
SHA1	Authenticate using the SHA1 algorithm.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set authentication sha1 show_rx`

IPSEC SET AUTHKEY

Specifies the authentication key (hexadecimal) for the IPSec SA.

ipsec set authkey <*key*> <*SAname*>

key Hexadecimal authentication key.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set authkey aaaaaaaaaabbbbbbbbbbccccccccccddddddddddd show_rx`

IPSEC SET COMPRESSION

Selects either LZS compression or no compression for the IPSec SA.

ipsec set compression <NONE | LZS> <*SAname*>

One of the following:
- NONE — No compression.
- LZS — Compress using the LZS algorithm.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set compression none show_rx`

IPSEC SET DIRECTION

Defines the direction of the IPSec SA.

ipsec set direction <INBOUND | OUTBOUND> <*SAname*>

One of the following:
- INBOUND
- OUTBOUND

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set direction inbound show_rx`

IPSEC SET ENCKEY

Specifies the encryption key.

ipsec set enckey <*key*> <*SAname*>

key Hexadecimal encryption key (64 bits for DES or 192 bits for 3DES).

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set enckey 1111111111222222222233333333334444444444555555555 show_rx`

IPSEC SET ENCRYPTION

Selects the encryption used for the IPSec SA: no encryption, DES (56-bit) encryption, or 3DES (168-bit) encryption.

ipsec set encryption <NULL | DES-CBC | 3DES> <*SAname*>

One of the following:

NULL Use no encryption.

DES-CBC Use DES (56-bit) encryption.

3DES Use 3DES (168-bit) encryption.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set encryption 3des show_rx`

IPSEC SET GATEWAY

Defines the IP address of the gateway of the IPSec SA.

ipsec set gateway <*IPaddress*> <*SAname*>

IPaddress IP address (4 decimals separated by periods).

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set gateway 207.135.89.233 show_rx`

IPSEC SET IDENT

Specifies the identifier (SPID) for the IPSec tunnel. It must match the SPID at the other end of the tunnel, that is, the tx SPID on this end must match the rx SPID on the other end.

ipsec set ident <*ident*> <*SAname*>

ident SPID for the IPSec tunnel.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set ident 424242 show_rx`

IPSEC SET MODE

Selects the encapsulation mode (tunnel or transport) for the SA. The default is tunnel mode.

ipsec set mode <TUNNEL | TRANSPORT> <*SAname*>

One of the following:
TUNNEL Tunnel encapsulation mode.
TRANSPORT Transport encapsulation mode.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set mode transport rtr2rtr`

IPSEC SET SERVICE

Selects the authentication and/or encryption services used for the SA.

ipsec set service <ESP | AH | BOTH> <*SAname*>

One of the following:

AH	AH authentication.
ESP	ESP encryption.
BOTH	Both ESP encryption and authentication.

SAname Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

Example: `ipsec set service both show_rx`

Appendix A. Network Information Worksheets

To configure the target (local) router, fill out the blank worksheet(s) that corresponds to the desired Link Protocol and Network Protocol:

If you are connecting to more than one remote router:

Fill out one set of information for each remote router in the *Remote Routers* section of the worksheet.

If you are setting up both ends of the network:

Use a *mirror image* of the information listed in your target router worksheet to configure the router on the other end of the WAN link.

Note: You may want to review the Sample Configurations, on page 65.

Configuring PPP with IP Routing

PPP with IP Routing		
Steps	**Commands**	**Your settings**
System Settings		
System Name	**system name** <*name*>	.....
System Message	**system msg** <*message*>	.....
Authentication Passwd	**system passwd** <*password*>	.....
Ethernet IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver**<*ipaddr*>	.....
Change Login	**system admin** <*password*>	.....
Remote Routers		
New Entry	**remote add** <*remoteName*>	.....
Link Protocol	**remote setProtocol PPP** <*remoteName*>	.....
PVC or DLCI	**remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*> **remote setDLCI** <*number*><*remoteName*>	.....
Security	**remote setAuthen** <*protocol*> <*remoteName*>	.....
Remote's Password	**remote setOurPasswd** <*passwd*> <*remoteName*>	.....
Bridging On/Off	**remote disBridge** <*remoteName*>	.....
TCP/IP Route Address	**remote addIproute** <*ipnet*> <*ipnetmask*> <*hops*> <*remoteName*>	.....
If NAT is enabled: To enable NAT -and- You may need to enter a Source WAN Port Address	**remote setIpTranslate** on <*remoteName*> **remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*>	.....
If NAT is OFF: You may need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*>	.....
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	**eth ip enable** **eth ip firewall** <on | off>	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

Configuring PPP with IPX Routing

PPP with IPX Routing		
Steps	**Commands**	**Your Settings**
System Settings		
System Name	**system name** <*name*>	.....
System Message	**system msg** <*message*>	.....
Authentication Passwd	**system passwd** <*password*>	.....
Ethernet IP Address	**eth ip addr** <*ipnet*> <*ipnetmask*> [<*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*>	.....
Change Login	**dhcp set valueoption domainnameserver** <*ipaddr*>	.....
Ethernet IPX Network #	**system admin** <*password*>	.....
	eth ipx addr <*ipxnet*> [<*port#*>]	
	eth ipx frame <*type*>	
Remote Routers		
New Entry	**remote add** <*remoteName*>	.....
Link Protocol	**remote setProtocol PPP** <*remoteName*>	.....
PVC or DLCI	**remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>	.....
	remote setDLCI <*number*> <*remoteName*>	
Security	**remote setAuthen** <*protocol*> <*remoteName*>	.....
Remote's Password	**remote setPasswd** <*password*> <*remoteName*>	.....
Bridging On/Off	**remote disBridge** <*remoteName*>	.....
IPX Routes Add	**remote addIpxroute** <*ipxNet*> <*metric*> <*ticks*> <*remoteName*>	.....
IPX SAPs Add	**remote addIpxsap** <*servicename*> <*ipxNet*> <*ipxNode*> <*socket*> <*type*> <*hops*> <*remoteName*>	.....
	remote setIpxaddr <*ipxNet*> <*remoteName*>	
IP and IPX Routing		
TCP/IP Routing	**eth ip disable**	.....
IPX Routing	**eth ipx enable**	.....
Store **Reboot**	**save** **reboot**	

Configuring PPP with Bridging

PPP with Bridging		
Steps	**Commands**	**Your Settings**
	System Settings	
System Name	**system name** <*name*>	.....
System Message	**system msg** <*message*>	.....
Authorization Password	**system passwd** <*password*>	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoptiondomainnameserver** <*ipaddr*>	.....
Change Login	**system admin** <*password*>	.....
	Remote Routers	
New Entry	**remote add** <*remoteName*>	.....
Link Protocol	**remote setProtocol PPP** <*remoteName*>	.....
PVC or DLCI	**remote setPVC** <*vpi* number>*****<*vci number*> <*remoteName*> **remote setDLCI** <*number*> <*remoteName*>	.....
Security	**remote setAuthen** <*protocol*> <*remoteName*>	.....
Remote's Password	**remote setOurPasswd** <*password*> <*remoteName*>	.....
Bridging On/Off	**remote enaBridge** <*remoteName*>	.....
	IP and IPX Routing	
IP Routing	**eth ip disable**	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

Configuring RFC 1483 / RFC 1490 with IP Routing

RFC 1483 / RFC 1490 with IP Routing		
Steps	**Commands**	**Your Settings**
	System Settings	
System Message	**system msg** <*message*>	.....
Ethernet IP Address	**eth ip addr** <*ipnet*> <*ipnetmask*> [*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*>	.....
	dhcp set valueoption domainnameserver < *ipaddr*>	.....
Change Login	**system admin** < *password*>	.....
	Remote Routers	
New Entry	**remote add** <*remoteName*>	.....
Link Protocol/PVC[a] (for ATM routers)	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <*vpi number*> ***** <*vci number*> <*remoteName*>	.....
Link Protocol /DLCI[b] (for Frame Relay Routers)	**remote setProtocol FR** <*remoteName*> **remote setDLCI** <*number*><*remoteName*>	.....
Bridging On/Off	**remote disBridge** <*remoteName*>	.....
TCP/IP Route Address	**remote addiproute** <*ipnet*> <*ipnetmask*> <*hops*> <*remoteName*>	.....
If NAT is enabled: To enable NAT -and- You must enter a Source WAN Port Address	**remote setIpTranslate on** <*remoteName*> **remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*>	.....
If NAT is OFF: You may need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*>	.....
	IP and IPX Routing	
TCP/IP Routing (Internet Firewall)	**eth ip enable** **eth ip firewall** <on | off>	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring RFC 1483 / RFC 1490 with IPX Routing

RFC 1483 / RFC 1490 with IPX Routing		
Steps	**Commands**	**Your Settings**
	System Settings	
System Message	**system msg** <*message*>	.....
Ethernet IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>	.....
Ethernet IPX Network #	**eth ipx addr** <*ipxnet*> [>*port#*>] **eth ipx frame** <*type*>	.....
Change Login	**system admin** <*password*>	.....
	Remote Routers	
New Entry	**remote add** <*remoteName*>	.....
Link Protocol/PVC[a] (for ATM routers)	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>	.....
Link Protocol/DLCI[b] (for Frame Relay Routers)	**remote setProtocol FR** <*remoteName*> **remote setDLCI** <*number*><*remoteName*>	.....
Bridging On/Off	**remote disBridge** <*remoteName*>	.....
IPX Routes Add	**remote addIpxroute** <*ipxNet*> <*metric*> <*ticks*> <*remoteName*>	.....
IPX SAPs Add	**remote addIpxsap** <*servicename*> <*ipxNet*> <*ipxNode*> <*socket*> <*type*> <*hops*> <*remoteName*> **remote setIpxaddr** <*ipxNet*> <*remoteName*>	.....
	IP and IPX Routing	
TCP/IP Routing (Internet Firewall)	**eth ip disable** **eth ip firewall** <on | off >	.....
IPX Routing	**eth ipx enable**	.....
Store **Reboot**	**save** **reboot**	

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring RFC 1483 / RFC 1490 with Bridging

RFC 1483 / RFC 1490 with Bridging		
Steps	**Commands**	**Your Settings**
System Settings		
System Message	**system msg** <*message*>	.....
DHCP Settings	**dhcp set valueoption domainname** <*domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>	.....
Change Login	**system admin** <*password*>	.....
Remote Routers		
New Entry	**remote add** <*remoteName*>	.....
Link Protocol/PVC[a] (for ATM routers)	**remote setProtocol RFC1483** <*remoteName*> **remote setPVC** <vpi *number*>*****<*vci number*> <*remoteName*>	.....
Link Protocol /DLCI[b] (for Frame Relay Routers)	**remote setProtocol FR** <*remoteName*> **remote setDLCI** <*number*><*remoteName*>	.....
Bridging On/Off	**remote enaBridge** <*remoteName*>	.....
IP and IPX Routing		
IP Routing	**eth ip disable**	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

Configuring RFC 1483MER / RFC 1490MER with IP Routing

RFC 1483MER/RFC 1490MER with IP Routing		
Steps	**Commands**	**Your Settings**
System Settings		
System Message	**system msg** <*message*>	.....
Ethernet IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*>[<*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** < *domainname*> **dhcp set valueoption domainnameserver** <*ipaddr* >	.....
Change Login	**system admin** <*password*>	.....
Remote Routers		
New Entry	**remote add** <*remoteName*>	.....
Link Protocol/PVC[a] (for ATM routers)	**remote setProtocol RFC1483MER** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>	.....
Link Protocol /DLCI[b] (for Frame Relay Routers)	**remote setProtocol MER** <*remoteName*> **remote setDLCI** <*number*><*remoteName*>	.....
Bridging On/Off	**remote disBridge** < *remoteName*>	.....
TCP/IP Route Address	**remote addIproute** <*ipnet*> <*ipnetmask*> <*ipgateway*> <*hops*> <*remoteName*>	.....
If NAT is enabled: To enable NAT,enter: *and* enter a Source WAN Port Address	**remote setIpTranslate on** <*remoteName*> **remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*> [c]	.....
If NAT is not enabled: You may need to enter a Source WAN Port Address	**remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*> <*remoteName*> [d]	.....
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	**eth ip enable** **eth ip firewall** <on | off>	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.
c The mask is the mask of the remote network.
d The mask is the mask of the remote network.

Configuring FRF8 with IP Routing

RFC 1483FR with IP Routing		
Steps	**Commands**	**Your Settings**
System Settings		
System Message	**system msg** <*message*>	.....
Ethernet IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]	.....
DHCP Settings	**dhcp set valueoption domainname** < *domainname*> **dhcp set valueoption domainnameserver** <*ipaddr*>	.....
Change Login	**system admin** <*password*>	.....
Remote Routers		
New Entry	**remote add** <*remoteName*>	.....
Link Protocol/PVC	**remote setProtocol FRF8** <*remoteName*> **remote setPVC** <*vpi number*>*****<*vci number*> <*remoteName*>	.....
Bridging On/Off	**remote disBridge** <*remoteName*>	.....
TCP/IP Route Address	**remoteaddIproute** <*ipnet*> <*ipnetmask*> <*hops*><*remoteName*>	.....
If NAT is enabled: To enable NAT -AND- You must enter a Source WAN Port Addr	**remote setIpTranslate on** <*remoteName*> **remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*> [a]	.....
If NAT is not enabled: You may need to enter a Source WAN Port Addr	**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*> [b]	.....
IP and IPX Routing		
TCP/IP Routing (Internet Firewall)	**eth ip enable** **eth ip firewall** <on | off>	.....
IPX Routing	**eth ipx disable**	.....
Store **Reboot**	**save** **reboot**	

a The mask is the mask of the remote network
b The mask is the mask of the remote network

Configuring a Dual-Ethernet Router for IP Routing

This table outlines commands used to configure a Dual-Ethernet router for IP Routing.

Dual-Ethernet Router - IP Routing		
Steps	**Commands**	**Your Settings**
System Settings		
System Name	**system name** <*name*>	.....
Message	**system msg** <*message*>	.....
Ethernet Settings		
Routing/ Bridging Controls	**eth ip enable** **eth br disable**	.....
ETH/0 IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]	.....
ETH/1 IP Address	**eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>]	.....
TCP/IP default route address	**eth ip addroute** <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*port#*>]	.....
DHCP Settings		
Define DHCP network for ETH/1	**dhcp add** [<*net*> <*mask*> | <*ipaddr*> | <*code*> <*min*> <*max*> <*type*>	.....
Create an address pool for ETH/1	**dhcp set addresses** <*first ipaddr*> <*last ipaddr*>	.....
DNS Domain Name	**dhcp set valueoption domainname** <*domainname*>	.....
DNS Server	**dhcp set valueoption domainnameserver** <*ipaddr*>	.....
WINS Server Address	**dhcp set valueoption winsserver** <*ipaddr*>	.....
Store	**save**	
Reboot	**reboot**	

Appendix B. Configuring IPX Routing

IPX Routing Concepts

To establish IPX Routing, you will need to enter all remote routers in the remote router database to which your router will connect.

1. For each remote router, enter the network addresses and services that may be accessed beyond the remote router.
2. Also enter a network number for the WAN link.
3. After you have specified the route addressing and services, you can then enable IPX routing across the Ethernet LAN.

Static Seeding: When IPX traffic is destined for network segments and servers beyond the remote router, the target router's routing information table must be statically seeded. Static seeding ensures that the target router connects to the appropriate remote router. After the link is established, RIP broadcast packets will dynamically add to the target router's routing table. Seeding the routing table is not necessary for target routers that never connect; they will discover remote networks beyond the calling router as soon as RIP updates arrive (provided the remote router supports RIP). However, for watchdog spoofing to work, you will need to define the remote IPX routes for network segments and servers.

Configure IPX Routing

Configuring your router for IPX routing can be rather complex. The following section will guide you through the configuration process. Remember that PPP authentication configuration must be completed *before* you attempt IPX routing configuration. The full router configuration for simple IPX routing includes the following:

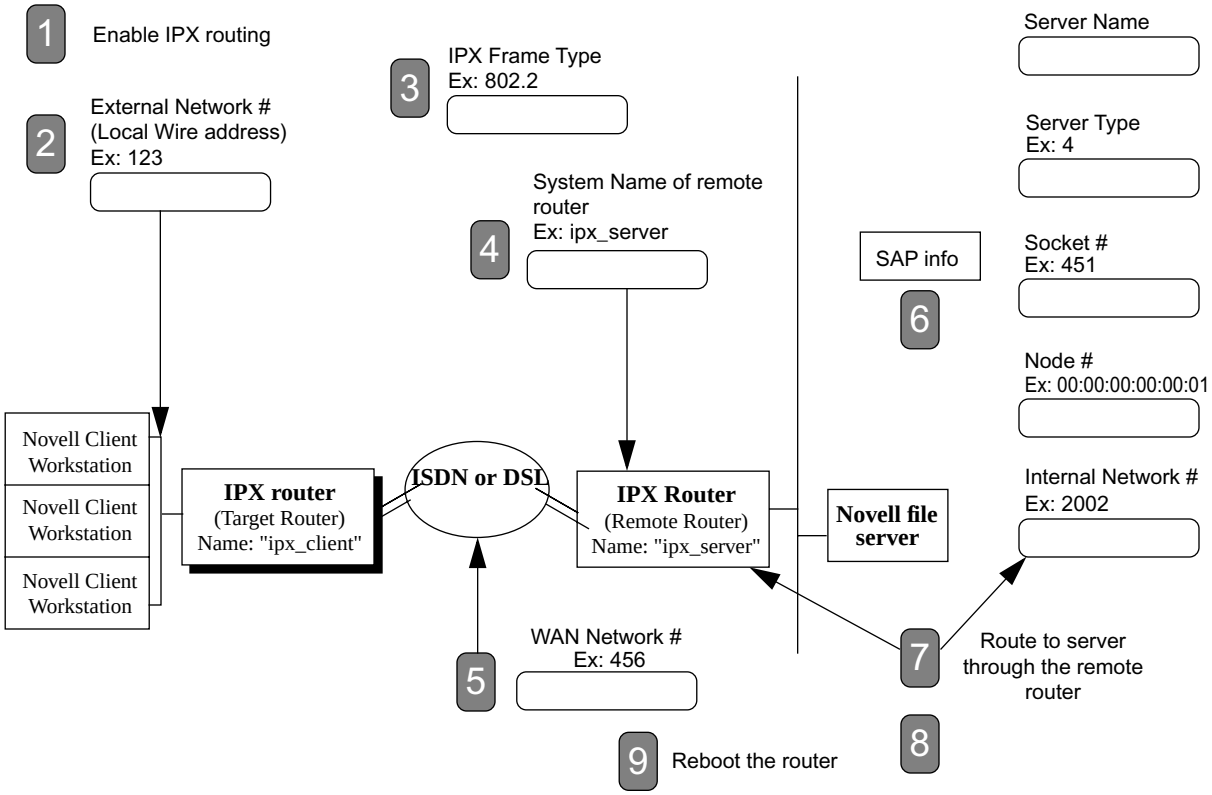
- PPP authentication
- IPX routing (this section)

The following section, Step 1: Collect Your Network Information for the Target (Local) Router, on page 409, provides a configuration diagram and a command table to assist you with the configuration of the target router.

Step 2: Review your Settings, on page 410 lists the commands used to review the IPX configuration and provides a configuration example.

Step 1: Collect Your Network Information for the Target (Local) Router

The remote side of the WAN link has all of the file and print services.
Enter the needed network information in the blank boxes of the diagram. Then match the boxes' numbers with the numbers in the Command Table below to configure the target router for IPX.



Command Table

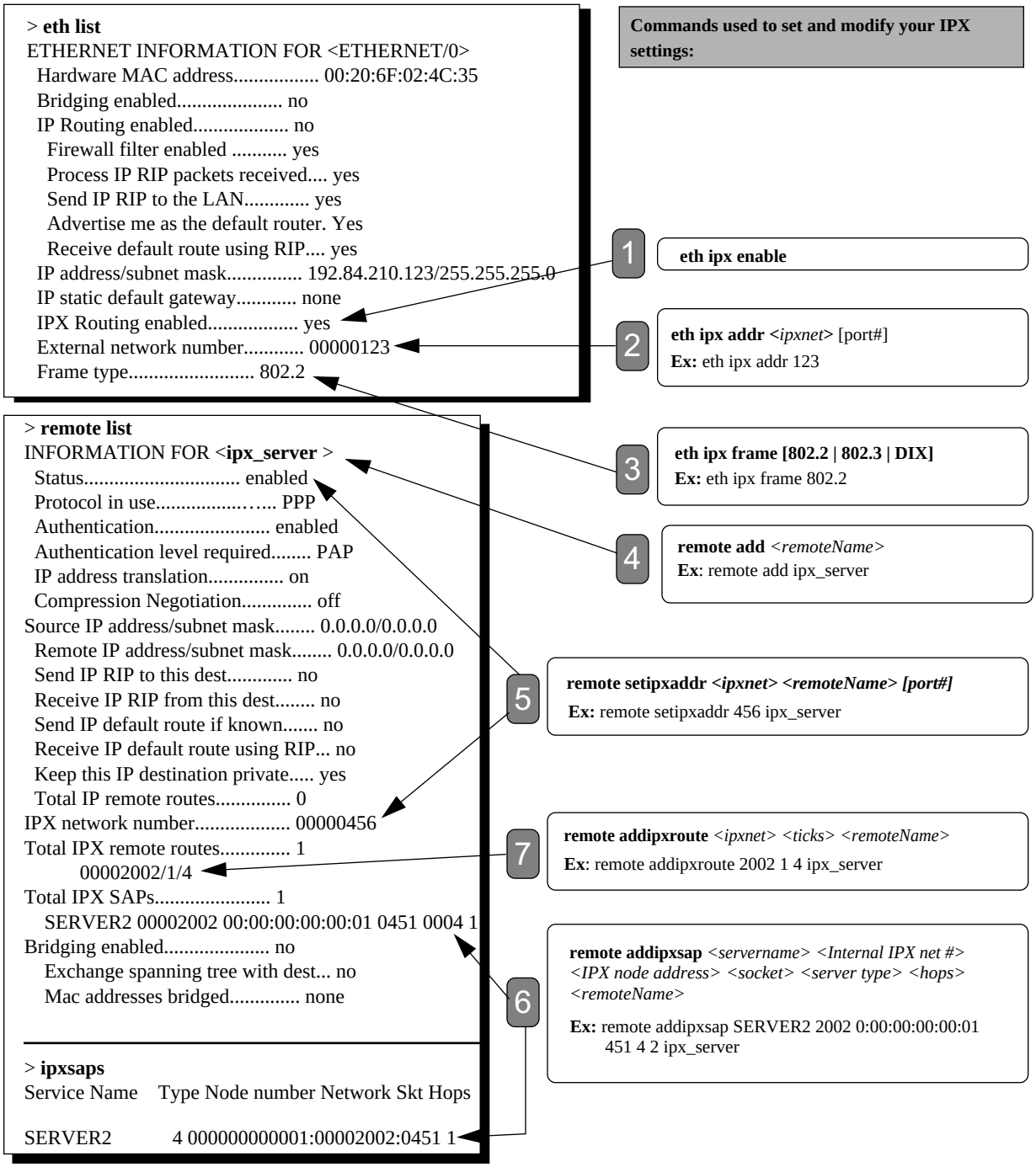
These commands are used to configure the Target (client-side) router (**ipx_client**). Log in with the password **admin.**

IPX Commands with examples	Ref #	Comments
eth ipx enable	**1**	Enable IPX Routing
eth ipx addr 123	**2**	Set the local 'wire' address
eth ipx frame 802.2	**3**	Set the Frame Type
remote add ipx_server	**4**	Add a connection name
remote setIpxaddr 456 ipx_server	**5**	Set the WAN network # (common to both sides)
remote addIpxsap SERVER2 2002 00:00:00:00:00:01 0451 4 1 ipx_server	**6**	Add a file server (SAP)
remote addIpxroute 2002 1 4 ipx_server	**7**	Add a route to the server
save	**8**	Save your settings
reboot	**9**	Reboot for changes to take effect

Step 2: Review your Settings

Commands used to review your IPX configuration:

- **eth list**
- **remote list**
- **ipxsaps**



Command Index

Symbols

A

B

C

D

E

F

H

I

K

L

M

P

R

S

T

V

Topic Index

D

E

F

G

H

I

J

K

L

M

N

P

R

S

T

U

V

W

Y