L-244067-52

AT&T UNIX® PC ENHANCED TCP/IP WIN/3B LAN INTERFACE

ADMINISTRATOR's GUIDE

July 1986

AT&T UNIX PC ENHANCED TCP/IP WIN/3B LAN INTERFACE

ADMINISTRATOR's GUIDE

CONTENTS

1. INTRODUCTION

The AT&T UNIX® PC Enhanced TCP/IP WIN/3B LAN Interface for the AT&T UNIX PC (WIN[1]/3B LAN) brings to the UNIX PC a complete networking environment. In addition to the primary administrative task of installation, there are many other tasks that need to be performed from time to time.

WIN/3B LAN has two user interfaces through which you can perform these administrative tasks: the standard UNIX System V shell, and the UNIX user agent or window interface.

1.1 PURPOSE OF THIS GUIDE

The purpose of this Guide is two-fold: to educate you on the elements of WIN/3B LAN and to orient you to good network administration practices. The emphasis is on building an understanding of what your network is and how to operate it, rather than on a recipe approach to network problem solving.

1.1.1 Prerequisites

This Guide assumes familiarity with administration of the UNIX[2] operating system on the UNIX PC. It also assumes some familiarity with WIN/3B LAN at the user level. It is recommended that before reading this Guide, you acquaint yourself with these other WIN/3B LAN documents:

WIN/3B LAN User's Guide
This Guide discusses *ftp*, *telnet* and other user

1. WIN is a trademark of The Wollongong Group, Inc.
2. UNIX is a registered trademark of AT&T.

commands. It contains an introduction to networking and a glossary.

WIN/3B LAN Programmer's Reference Manual
This Manual is the complete reference for WIN/3B LAN. Special attention should be focused on *intro(4)*, "Introduction to Networking Facilities", and Section 1M, "Network Maintenance and Operation Commands".

1.1.2 Conventions

A number of conventions are employed throughout this Guide.

Where samples of interactive sessions are given, user input is shown in **boldface** while system responses are shown in plain print.

Filenames, commands, and option names are shown in *italics* unless they are part of a chapter or section heading or sample session. In headings they may appear partially or completely capitalized when in practice they are not. Filenames are always shown with their pathnames.

The user's home directory is referenced by a ~ (tilde). The pathname *~/filename* indicates that *filename* is in that user's home directory.

The first occurrence of a new term is enclosed in quotation marks unless it is used in a chapter or section heading. Subsequent references are not so marked.

1.1.3 Contents

This Guide is organized into eight chapters and three appendices. Chapters 1 and 2 provide background information on the nature of the tasks WIN/3B LAN performs and on WIN/3B LAN structure.

Chapter 3 discusses user tools provided by WIN/3B LAN, explaining how they operate and giving examples of their use.

Chapter 4 discusses the network databases. These files contain information critical to the operation of WIN/3B LAN. The purpose and format of each is discussed.

Chapter 5 discusses the automated procedure for installing WIN/3B LAN software. Using this procedure with the WIN/3B LAN menu interface makes the installation process simple and straightforward.

Chapter 6 discusses network configuration and maintenance procedures. The configuration and maintenance operations are located in a single menu for quick access and easy execution.

Chapter 7 discusses the procedure to deinstall software on the WIN/3B LAN. This procedure is simplified through use of a single menu option.

Chapter 8 contains several sections of WIN/3B LAN error messages along with brief explanations of their possible causes.

Appendix A points you to the documentation for hardware installation.

Appendix B contains three examples of how to go about resolving network problems.

Appendix C lists some good sources of networking information.

1.2 NETWORKING WITH WIN/3B LAN

This chapter provides context information for the rest of the Guide: a brief introduction to some of the basic networking concepts embodied in WIN/3B LAN. If you are already familiar with networking or WIN/3B LAN in particular, you may wish to skip to the next chapter.

1.2.1 Computer Networks: Layers, Protocols, and Interfaces

Interconnecting computers allows an organization to share data and computing resources. As the number of computers within an organization increases, the value of interconnecting them increases. Computers that are interconnected to form a network are called "hosts".

Networking is made possible through the interaction of hierarchically organized "layers" of functionality. The layer furthest from the user is the physical; the others are all software layers. Each succeeding layer provides more and more sophisticated services to the layer directly above it until finally the user is presented with the most sophisticated layer, called the "application" layer, through which he readily and without any networking expertise may make use of the facilities provided by the underlying layers. This is analogous to the way in which UNIX shell commands provide a user with readily accessible operating system services without requiring any knowledge of how those services are provided.

Each layer carries out a specific subset of related functions. These functions taken together implement a "protocol", which is a set of rules governing the actions within a given layer. The protocol implemented in a layer is specific to that layer, and for that reason a layer on one host can communicate only with the corresponding layer on another host, called a "peer". This characteristic of protocols, that they communicate only horizontally, is called "peer-to-peer" communication. In some cases a layer contains more than one protocol, but the principle of peer-to-peer communication still pertains.

But communication requires all layers working together. When communication takes place, data is passed vertically across layer boundaries according to rules known to the two interacting layers. These rules comprise the "interface" between the two layers. Layers interact vertically only with adjacent layers.

A user entering a networking command causes the passage of data down through the succeeding layers until it eventually makes its way to the hardware or "physical" layer. On its way down it gets successively packaged by the different layers, each protocol usually adding a "header" or "leader" to what it receives. This is referred to as the "packetizing" of the data; the data packages are called "packets". When a packet comes out the hardware at the other end, it is passed up through the succeeding layers, each layer stripping off the header attached by the corresponding layer on the source host and acting on the packet according to the information contained in the header.

WIN/3B LAN supports a set of U.S. Department of Defense (DoD) standard protocols collectively known as TCP/IP (Transmission Control Protocol/Internet Protocol) and thus enables AT&T UNIX PC computers to interconnect with other computers that support these protocols. Wollongong and other companies supply TCP/IP for many UNIX computers; UNIX PC computers can interconnect with these as well as with other UNIX PC computers by way of WIN/3B LAN.

For more information on hierarchical network architecture, see the references listed in Appendix C.

1.2.2 Ethernet Physical Transport Medium

Ethernet, originally developed by the Xerox Corporation in 1972, but since adopted as a standard by many companies, is currently one of the physical network media available for UNIX PC computers. The physical components of an Ethernet network are cable, transceivers, and a network interface board. The Ethernet cable is the actual transmission medium. The transceiver taps onto the Ethernet cable and is connected to the interface board by another length of cable. The Ethernet interface board resides in the backplane of the computer and enables it to formulate outgoing data and interpret incoming data according to established rules.

Installation of these physical components enables a computer to communicate with other computers over the Ethernet cable. This physical configuration is shown in Figure 1-1.

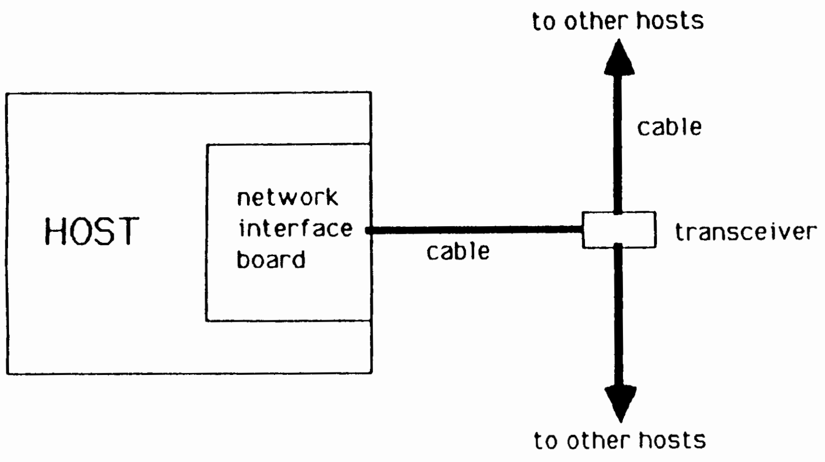


Figure 1-1. Ethernet Physical Configuration

1.2.3 Ethernet Addressing

Once the physical means for communication are in place, a further concept is required. This is the notion of "addressing". Addressing is simply the assignment of a unique designation to

each host on a network. WIN/3B LAN supports both "broadcast" and "point-to-point" addressing. Broadcast addressing enables a host to transmit a packet to all hosts on its network. Point-to-point addressing enables a host to transmit a packet to a specific host.

Every Ethernet interface board has a unique address, called an Ethernet address, assigned to it at the time of its manufacture; being interface board specific, it is a "physical" address. This address contains six bytes. The first three bytes are a unique manufacturer identifier assigned to the manufacturer by the Xerox Corporation. The remaining three bytes are determined by the manufacturer, who must ensure that these three bytes are unique for each interface board it manufactures. This part of the address is analogous to a serial number.

1.2.4 Internetworking

The interconnection of networks extends the concept of computing resource sharing a step beyond simple networking. It is useful to any organization that wants to gain access to computers that are already on existing networks. A common scenario is an organization wanting to interconnect their local area network with the ARPANET[3].

1.2.4.1 Gateways

A "gateway" is the physical mechanism for interconnecting networks. A host computer acting as a gateway has a hardware connection to each of the networks being connected, as illustrated in Figure 1-2.

3. The network of the Department of Defense Advanced Research Projects Agency.

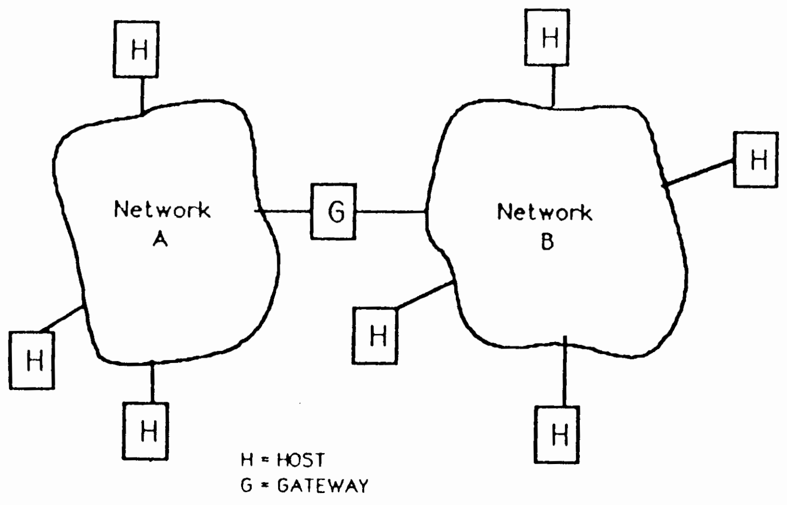


Figure 1-2. A Gateway Between Two Networks

Gateways provide access to hosts on other networks even though they are not directly connected to your network. For instance, gateways can connect your UNIX PC host to hosts on other private local area networks (LANs), or to public networks such as ARPANET.

1.2.4.2 Routing

But a physical gateway is not enough to enable Internetwork (often abbreviated to "Internet") communication. Hosts on an Internet also require information on how to get packets to one another: they require "routing" information. This is nothing more than a description of how to get from one host to another on an Internet. Routing is the software concept that

complements the hardware concept of a gateway; it enables hosts to utilize the hardware capability provided by the gateway. Currently, each host on a TCP/IP Internet maintains its own routing information.

1.2.4.3 Internet Addressing

Routing through a gateway implies the concept of "Internet addressing". An Internet address is the designation by which a host is known to other hosts, whether on its own or a connected network. Internet addresses are independent of the physical addresses (e.g. Ethernet addresses) in force within interconnected networks, and are therefore called "logical" addresses. In WIN/3B LAN, for example, the Internet address of a host is independent of its Ethernet address, and therefore will not change if the physical transmission medium changes: if the Ethernet interface board on a host fails and is replaced with another; its Ethernet address changes, but its Internet address remains the same.

All Internet addresses contain four bytes, but what each of these bytes signifies varies with the Internet address class it represents. There are three classes of Internet address: A, B and C. All hosts on a given network share the same Internet address type. The type chosen depends on, and thence determines, the way the network is organized.

Class A addressing reflects the simplest network organization. It allows a collection of hosts to be divided among 128 networks. This is by far the most common network organization in use today. If more than 128 networks are needed, then class B or C addresses must be used. In Class A addressing, the first byte of the Internet Address identifies the host's network. The last three bytes identify the individual host.

Class B allows only 64 networks, but each of these networks can be subdivided into 512 subnetworks. Converting a class A network into class B requires restructuring the network into 64 networks each containing up to 512 subnetworks. In Class B addressing, the first byte of the Internet address identifies the host's network; the second byte identifies the host's subnetwork. The last two bytes identify the individual host.

Class C allows only 32 networks, but each of these may be subdivided into 512 subnets. Each subnetwork may be further subdivided into 512 subhosts. Converting a class A network into class C requires a major redesign of your network topology. This network class is rarely, if ever, used today. In Class C addressing, the first byte of the Internet address identifies the host's network; the second byte identifies the host's subnetwork; and the third byte identifies the host's local area network (LAN). Finally, the last byte identifies the individual host.

There are many factors to consider when determining which network organization and Internet address class to implement; this subject is beyond the scope of this Guide. One salient point, however, is that while the organization chosen may reflect the actual, physical layout of the Internet, it may also reflect a purely logical view without reference to physical layout. For instance, one of the more sophisticated organizations, class B or C, may be implemented in order to facilitate a closer tracking of network usage for billing purposes. For more detailed information on the formulation of Internet addresses, see Section *inet(3N)* in the **WIN/3B LAN Programmer's Reference Manual.**

2. WIN/3B LAN SOFTWARE COMPONENTS

WIN/3B LAN adds new services to the UNIX operating system on your UNIX PC. Some are provided at the user level, while others are incorporated into and provided by the UNIX kernel.

2.1 USER LEVEL

The user level components of WIN/3B LAN fall into three categories: programs or commands, software function libraries, and databases. Virtually all the networking needs of general users, networking programmers, and network administrators can be satisfied by the components provided at this level. These components are illustrated by Figure 2-1.

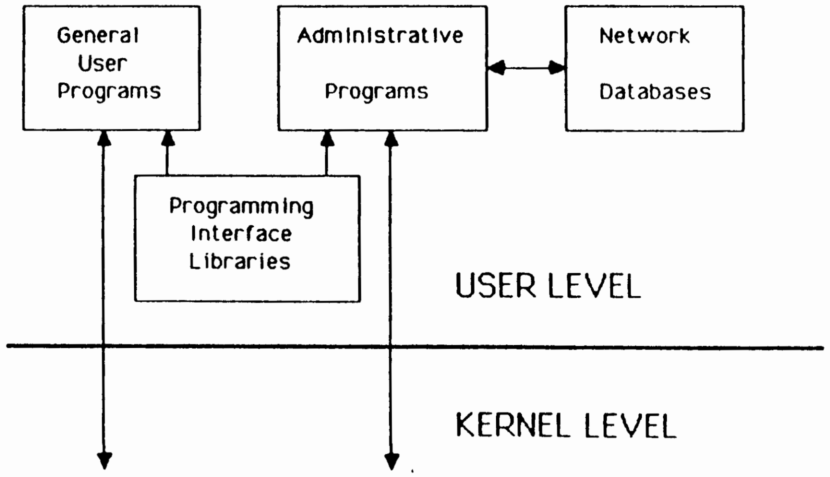


Figure 2-1. User Level WIN/3B LAN Components

2.1.1 User and Administrative Programs

The application programs are of two types: commands available for general use and those whose use is restricted to the superuser.

The general use commands provide users with the ability to login to remote hosts, to transfer files between networked hosts, and to get various data about the current status of the network and the hosts accessible through it. A facility for sending electronic mail is also provided. These commands are discussed in depth in the **WIN/3B LAN User Guide** and in Section 1 of the **WIN/3B LAN Programmer's Reference Manual**. Points of administrative interest regarding them are discussed in the chapter on "GENERAL USER SERVICES".

The restricted use commands perform a variety of administrative functions. Some used for maintaining network information tables kept in the UNIX kernel. These commands are discussed in detail in Section 1M of the **WIN/3B LAN Programmers's Reference Manual**, as well as in the chapters on "GENERAL USER SERVICES" and "CONFIGURATION AND MAINTENANCE".

2.1.2 Network Databases

In order to function, the network needs not only hardware and software but also information. For example, the network needs to know the names, addresses, and services available on each host. Some of this information is maintained in tables within kernel portions of WIN/3B LAN, but much of it is kept in files that can be modified only by the superuser. Setting up and maintaining these files are important network administration duties. These files are discussed in Section 5 of the **WIN/3B LAN Programmer's Reference Manual**, as well as in the chapter on "NETWORK DATABASES". Procedures for setting up and maintaining these files are discussed in the

chapters on "SOFTWARE INSTALLATION" and "CONFIGURATION AND MAINTENANCE".

2.1.3 Programming Interface Libraries

Two user level programming interface libraries are available through which networking services can be invoked. These are the Socket Compatability Interface (SCI) library and the Transport Level Interface (TLI) library. They provide mutually exclusive interfaces to the kernel level network services. All networking application programs rely on one or the other of these libraries.

The SCI library provides a family of functions based on the concept of a "socket." This library provides an interface that is compatible with the networking system call interface available in the 4.2 BSD version of the UNIX operating system. This library is defined in detail in Section 3W of the **WIN/3B LAN Programmer's Reference Manual**.

The TLI library is an implementation of a family of functions newly defined by AT&T. These functions will eventually become a standard part of UNIX and will be fully supported by AT&T. This library is defined in detail in Section 3T of the **WIN/3B LAN Programmer's Reference Manual**.

2.2 KERNEL LEVEL

WIN/3B LAN kernel level services are provided by a UNIX special file or loadable device driver. Device drivers are the software engines in UNIX whose job it is to manage i/o. Ordinarily, each device driver is tailored to one particular kind of hardware device, the capabilities of which it manages and makes accessible to the operating system and so to the user. Drivers normally contain entry points that allow a user to open, read from, write to, send UNIX ioctls to, and close the device. They also contain interrupt handlers through which they interact with the underlying hardware.

Some drivers are not associated with hardware. These are referred to as pseudo device drivers. A pseudo device driver simulates the action of a hardware driver and is sometimes used as a multiplexer to other device drivers.

The drivers discussed above and below are all contained within the single loadable device driver that is loaded at boot time.

2.2.1 Network Pseudo Driver

This pseudo driver is the network protocol engine, encapsulating support for the TCP/IP protocol family and providing the services advertised by both programming interface libraries (SCI and TLI). It accepts data from the user level, packages it according to the chosen protocols, and passes it on to another driver. Conversely, it accepts data from that other driver, unpackages it according to the appropriate protocols, and passes it up to the user level.

This driver, */dev/net*, is composed of the following, starting with what is closest to the user level and working down:

A. Transport Endpoint Interface (TEI);

B. Transmission Control Protocol (TCP) or User Datagram Protocol (UDP);

C. Internet Protocol (IP) and Internet Control Message Protocol (ICMP).

This driver consists of an interface and two protocol layers. In addition to the TEI there is also an interface between the TCP/UDP protocol layer and the IP/ICMP layer, but since it is private to these layers and is invisible to users and administrators alike, it receives no further mention here. The structure of */dev/net* is shown in Figure 2-2.

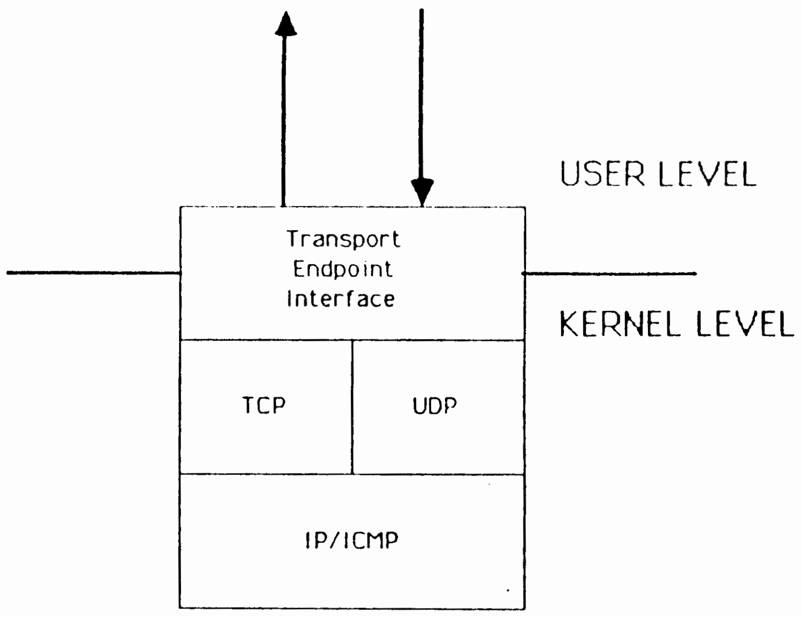


Figure 2-2. Network Pseudo Driver Structure

2.2.1.1 Transport Endpoint Interface (TEI)

The Transport Endpoint Interface is a body of kernel level entry points built around the concept of a "transport endpoint". These entry points correspond to the functions in the SCI and TLI libraries. They make the services of the TCP/UDP protocol layer available to those libraries.

A transport endpoint identifies an instantiation of the opened network pseudo device. When a given instantiation is created and opened, an identifier is returned to the calling user level process. This identifier is then referenced in order to carry out any further communication between the user level process and the TCP/UDP protocol layer.

The SCI and TLI libraries have different views and implementations of transport endpoints, although the facilities provided are essentially the same. In the terminology of the SCI library transport endpoints are sockets and in that of the TLI library they are simply standard UNIX file descriptors.

2.2.1.2 Transmission Control Protocol (TCP)

TCP and UDP provide users with different types of service, but reside in the same protocol layer. The primary responsibility of TCP is to provide user programs, through the TEI, with "virtual circuits". A virtual circuit is a reliable data pathway established between the TCP layers of two communicating hosts. Reliability means that the data sent out on a virtual circuit always arrives exactly as it was sent. TCP guarantees this reliable, sequential data transfer.

2.2.1.3 User Datagram Protocol (UDP)

UDP provides an unreliable datagram service to user programs. A datagram is approximately the same thing as a packet. UDP does not enforce reliability, that is, it does not provide a sender with any indication that a sent packet was successfully received. UDP also does not guarantee that data will be received in the same order in which it was sent. Reliability and data sequencing are the specialty of TCP.

2.2.1.4 Internet Protocol (IP) and Internet Message Control Protocol (ICMP)

IP is responsible for routing datagrams to their destination Internet addresses. Like UDP, it is a datagram service which does not guarantee reliable or sequenced data transfers. In fact it is the IP datagram service which UDP makes available to the user level through the TEI.

ICMP is an adjunct of IP used to control the behavior of IP. It informs IP of errors such as bad routes and also communicates general control information such as that necessary for data flow control.

2.2.2 Ethernet Driver

The Ethernet driver is responsible for managing the Ethernet interface board. It receives outgoing packets from IP and passes them to the interface board, which puts them out on the net. Conversely, incoming packets are taken off the interface board and passed upstream to IP.

2.2.3 Terminal Psuedo Driver

The *pty*, or psuedo *tty*, is a pseudo terminal driver that enables an interactive user to login to a remote host. This is known as a network virtual terminal (NVT) capability. In UNIX each user is connected to a shell via a *tty* driver. The *tty* driver handles all terminal control information like XON, XOFF, and tabs. The *pty* driver performs the same function when a user talks to a remote shell. All terminal control information is passed from the local host across the network to the remote *pty*. The *pty* processes the information and the remote shell never knows it is not connected to a real *tty*. There are eight ptys on each host. Hence each system can handle a maximum of eight remote logins.

2.2.4 Loopback Pseudo Driver

This pseudo driver utilizes networking facilities without ever actually sending a packet out onto the network. Instead, communication is set up with the local host as though it were a remote host: the communication is "looped back". This driver is useful for testing networking facilities.

3. GENERAL USER SERVICES

General user services typically involve the cooperative interaction of two separate processes, one on the local host and one on the remote. The local host process is referred to as a "client" because in initiating the interaction it is seeking the services of a corresponding process on a remote host. The remote host process is referred to as a "server" because its purpose is to serve the needs of the client process. This cooperative interaction, called the "client/server model", is the paradigm for the operation of most general user services. A corollary of this is that, for most general user services, if an appropriate daemon is not running on the chosen host the service is unavailable.

The normal life cycle of a client process starts when a user enters one of the networking commands. The client process then sends a request for service to the appropriate process on the host specified by the user. If the request is honored, a connection is established, or "opened", between the local client and remote server processes. The user transacts the remainder of his or her business over this open connection. Finally, the user enters the appropriate termination sequence, the connection is closed, and the client process dies. A server starts its life cycle as a background process started by the superuser. In this initial state the server is called a "daemon". The daemon "listens" at a known address for incoming service requests. When it receives a request, it establishes a connection with the requesting client, spawns a child process, and goes back to listening for more incoming requests. The child, the server, inherits the connection established by the parent and handles all further client business for the duration of the session. When the termination sequence sent by the client is received by the server, it closes the connection and dies.

One caveat on the above distinction drawn between daemon and server: in common parlance the distinction is often ignored, and the terms "daemon" and "server" are used

interchangeably. You should be able to tell from the context which process is meant.

3.1 FTP AND FTPD

The program that implements the DoD File Transfer Protocol is called *ftp*. It enables a user to transfer files between the local and a remote host.

When establishing a connection with *ftp*, a remote host can be specified by its Internet address, its official "hostname" or by an "alias". If the official hostname or an alias is supplied, *ftp* searches the file */etc/hosts* to find the corresponding Internet address. Using the Internet address, it makes a connection with the remote *ftp* daemon, called *ftpd*. The first step of the *ftpd* server is to validate the user making the request. It does so according to three rules:

1. The user's username must be in the standard UNIX file */etc/passwd*. When this check is passed, the user is prompted for the password before any file transfer operations can be performed. If there is no password associated with the login name, the user must still type something other than a carriage return when prompted for a password, to gain access to the remote host.
2. If the username is "ftp" and an account is present in */etc/passwd*, the user is allowed to login by specifying any password. Since anyone can login under "ftp", it is wise to restrict the access privileges of this account.
3. The username must not appear in the file */etc/ftpuser*. If it does, access is denied.

3.2 TFTP AND TFTPD

The DoD Trivial File Transfer Protocol user program, called *tftp*, is simpler than *ftp* but less versatile.

tftp differs from *ftp* in a number of ways. One is that it lacks any login facility. Another is that while *ftp* uses TCP, *tftp* uses UDP. For this reason *tftp* is considered an unreliable method of transferring files.

When *tftp* is invoked, it makes a connection with the remote *tftpd* daemon. *tftp* does not validate users and therefore can only be used to transfer publicly readable files.

3.3 TELNET AND TELNETD

The *telnet* program implements the DoD network virtual terminal protocol TELNET. It enables a user to login to a remote host from a local host, provided the user has an account on the remote host. It appears then to the user as though his or her terminal is directly attached to the remote host. *Telnet* can be invoked in either the input or the command mode. In the input mode, *telnet* is invisible: the user is logged into the remote host and goes about his or her business.

In the command mode, the user enters commands to *telnet*. One of these commands is the *negotiate* command. It can be used to negotiate the setting of various optional qualities of the opened connection. This command supports four actions through which negotiation proceeds: *do, dont, will*, and *wont*. (Note that *dont* and *wont* do not contain apostrophes.) When *do* or *dont* are used to initiate a negotiation, the client requests that the remote server set or unset the option specified. The remote server then responds that it either *will* or *wont*. A negotiation started with *will* or *wont* announces to the server that the client is willing or unwilling to set a specified option. The remote server will respond with *do* or *dont*. When the remote server makes the appropriate response, the negotiation is complete and the option is set or not, accordingly.

Six options can be negotiated:

binary	This option sets transmission to binary. By default, telnet transmits with seven bits, using the eighth bit for parity checking. Some applications require the full eight bits.
status	This command causes all options currently set on the server to be displayed.
echo	This option is used to set character echoing by either the client or the server.
sga	Suppress go ahead. Half duplex systems need a "go ahead" signal to begin transmitting once the other system is finished. This option is unnecessary for full duplex systems.
tm	Timing mark. This forces the server to flush all data in its queue and to send a timing mark. This command is useful if you have a connection that you believe is hung.
exopl	Extended options list. This list is not currently implemented but is included in the TELNET specification in anticipation of additional options.

Negotiation examples are shown below. The blank line after each successful response indicates how *telnet* command mode is exited after just one correct entry.

```
telnet> options
Will show option processing.

telnet> negotiate
negotiate> do status
SENT IAC DO STATUS
RCVD IAC WILL STATUS
SENT IAC SB STATUS SEND IAC SE
STATUS IS:
    WILL BINARY
    WILL ECHO

telnet> n
negotiate> dont echo
SENT IAC DONT ECHO
RCVD IAC WONT ECHO
```

3.4 RLOGIN AND RLOGIND

The *rlogin* command, like *telnet*, provides a way to login to a remote host. It differs from *telnet* in that it lacks any external standard by which the correctness of its implementation can be judged and it was designed and implemented specifically for UNIX systems.

rlogind is the daemon for *rlogin*. When the server is spawned it first attempts to validate and automatically log the user in by checking the file named */etc/hosts.equiv*. This file contains a list of hosts with which the current host has user accounts in common. If the incoming request is from a so called "equivalent" host, it then looks up the remote user's username in the local copy of */etc/passwd*. If it finds the username, the remote user is automatically logged in as if he or she were logging in on a terminal directly attached to the server's host. A special type of entry in */etc/hosts.equiv* can be used to allow a remote user to autologin using a username other than his or her own. See *hosts.equiv(5)* in the **WIN/3B LAN Programmer's Reference Manual** for more details.

If the incoming request is not coming from an equivalent host, *rlogind* looks for an account in */etc/passwd* that matches that used or specified by the requesting user. If such an account is found, the home directory associated with it is searched for a file named *~/.rhosts*. If found, *rlogind* tries to find an entry in it that matches the requesting username and the requesting host. If such is found, the user is automatically logged in. If

all autologins are unsuccessful, the user is prompted for username and password. See Section 4.6 of this manual for more information on *~/.rhost*.

3.5 RWHO, RUPTIME, AND RWHOD

The *rwho* command produces output similar to the UNIX *who* command but displays the users of all hosts on the local network. If a user has not typed input to the system for a minute or more, *rwho* reports this idle time. A user logged in but inactive for an hour or more is omitted from the *rwho* report.

The *ruptime* command displays the status of all hosts on the local area network. This command will print a table that contains the hostname, the current host status (up or down), amount of time that the host has been up or down, the number of users on the host, and the load average. IT includes all of the hosts on the local area network. The *ruptime* command normally sorts by hostname the list that it displays.

rwhod is the server used by both *rwho* and *ruptime* and should be running on bothe hosts for complete information. This server is both a producer and consumer of information. As a producer, every two minutes *rwhod* sends a broadcast packet over the network. This packet contains load averages, a list of users and other information about the host on which it resides. This packet is read by the *rwho* daemons on the other networked hosts. Those daemons demonstrate the consumer side of the *rwhod* character. When one of them gets such a broadcast packet, it stores the information from the packet in a file in the directory */usr/spool/rwho*. A file is maintained in that directory for every host on the network. When a new host is configured into the network and its *rwho* daemon emits a broadcast packet, a receiving *rwhod* makes a new file for this host in */usr/spool/rwho*.

If *rwhod* does not receive a broadcast message from a remote host it assumes that host is down. Of course, if a remote host is listed as down, it is possible that only the *rwhod* is not running. To remove the *ruptime* entry of a host that has been permanently removed from a network, just delete the file on that host in the */usr/spool/rwho* directory.

3.6 REMSH, RCP AND RSHD

The *remsh* or "remote shell" command enables a user to execute a command on a remote host. *remsh* passes INTERRUPT, QUIT and KILL signals to the remote host. Interactive commands, such as *vi* cannot be run using *remsh*. The connection normally terminates with the termination of the command. If no command is specified, the user is simply logged in.

The *rcp* command enables a user to transfer files between the local host and a remote host or between two remote hosts. Entries for both hosts must exist in the *etc/hosts.equiv* file for the transfer to be successful. It is essentially a special case of the more general *remsh* and operates in much the same manner.

rshd is the server for both *remsh* and *rcp*. The *rshd* server attempts to validate and automatically login the user in the same way as *rlogind*. If the autologin is successful and a shell is spawned, the shell inherits the opened connection.

and, upon termination of the command, the shell and the connection die.

3.7 REXEC, REXECD

The *rexec* function enables a program to execute commands on a remote host. This command differs from the *rsh* command in that it is invoked by a C program. If a username and

password are both specified, these are used to authenticate the user to the remote host. Otherwise the user's *.netrc* file on the local host is searched for the appropriate information. *rexecd* is the server for *rexec*. Once the user is authenticated on the remote host, the server spawns a shell, and this shell inherits the network connections established by the server.

3.8 FINGER, FINGERD

The *finger* command displays information about a user or users. It can also produce information about users on remote hosts. *fingerd* is the server for the finger command. This command operates in the same fashion on a remote host as it does on the local host.

3.9 /ETC/LDDRV/ETHER.RC

When the system is booted and the network is brought up, addresses must be assigned to the network interfaces, and the daemons must be activated. This is accomplished through the shell script */etc/lddrv/ether.rc*. By default, this shell script starts the following networking daemons:

```
ftpd
telnetd
fingerd
rlogind
rshd
sendmail
```

This file may also contain network-related booting tasks, such as setting up routes and assigning special network addresses. Network administrators will generally want to modify this file for their own sites.

4. NETWORK DATABASES

Information about the hosts, networks, services and protocols is maintained in a set of network database files. These files should be updated whenever the topology of the network changes. The procedures for changing the information in these files are discussed in Section 6 of this document, "Configuration and Maintenance."

The names used for networks, hosts and host aliases must conform to some restrictions:

1. The official *hostname* is the name of the host. This name is usually but not necessarily the same as that returned by the UNIX *uname* command. See *hostname(1)* in the **WIN/3B LAN Programmer's Reference Manual** for more information.
2. Network and host names are limited to 32 characters in length.
3. Names must not contain any metacharacters: $? { } [ ] * ! / \

Generally aliases are shorter than the official names. A network can have several aliases. For more information on these files, see Section 5 of the **WIN/3B LAN Programmer's Reference Manual**.

4.1 /ETC/HOSTS

/etc/hosts is an ASCII file containing the official hostnames, aliases, and Internet addresses of remote hosts. Networking applications programs reference this file to find the Internet addresses of hosts. On each line, the format is: Internet address, official hostname and aliases. These fields can be separated by spaces or tabs. The official hostname is the name

of the host.

Each byte of the Internet address should be separated by a period ".". The aliases can be anything, and are usually an abbreviation of the official name. Comments must begin with a pound sign "#".

```
##################################
# Local Hosts
##################################

85.34.09.01 officalname1   alias1a  alias1b  alias1c
85.34.09.02 officalname2   alias2a
85.34.09.03 officalname3   alias3a  alias3b
```

Figure 4-1. Example of */etc/hosts*

Both *ftp* and *telnet* sequentially search this file for the Internet address of the host. It is good practice to put the most frequently used hosts at the beginning of this file to shorten search time. If there are two entries for one remote host, the first entry is used. This file can only be edited by the superuser.

4.2 /ETC/NETWORKS

/etc/networks lists the names, network numbers, and aliases of each network to which the current host has a direct connection. From this file IP determines which network a packet should be routed to. The format of the file is: official network name, network number and aliases. Fields are separated by blanks or tab characters.

This file is supplied with distribution. You may need to update it for unofficial aliases and local changes. If your network will be connected to a larger network, for example the ARPANET, your network name and number will be assigned to you.

```
Loopback        127         loop lb
Ethernet        89          ether enet
```

Figure 4-2. Example of */etc/networks*

Like */etc/hosts* this file is searched sequentially, so for general efficiency place the most frequently accessed networks at the top of the file.

4.3 /ETC/PROTOCOLS

/etc/protocols is used by IP to determine which higher level protocol packets should be sent to. From the packet header, IP gets a protocol number, and finds the corresponding protocol using this table.

Protocol names can contain any printable character other than a space, tab, newline or comment character. The format is: official protocol name, protocol number and any aliases. Comments are preceded with a pound sign "#".

```
#
# Internet (IP) Protocols
#
ip      0    IP      # Internet protocol,
icmp    1    ICMP    # inet control message protocol
tcp     6    TCP     # transmission control protocol
udp     17   UDP     # user datagram protocol
```

Figure 4-3. Example of */etc/protocols*

4.4 /ETC/SERVICES

/etc/services lists which transport protocol (TCP or UDP) and standard port number each user level service uses.

The format of the file is: official service name, port number/protocol name and aliases of the services. The port number and protocol name are separated by a forward slash "/". New services can be added.

```
#
# Unix specific services
#
exec     512/tcp
login    513/tcp    log
shell    514/tcp    sh
who      513/udp
syslog   514/udp    sl
talk     517/udp
route    520/udp    rte
```

Figure 4-4. Example of */etc/services*

4.5 /ETC/HOSTS.EQUIV

The */etc/hosts.equiv* file is an optional file used by the servers *rlogind* and *rshd* to authenticate a request for login coming from a user on a remote machine. It contains a list of remote or equivalent hosts. A user on a remote host with the same username can login automatically if the remote host is listed in this file. The system searches for the username in */etc/passwd*, and if it finds it the user is logged in immediately.

A special entry in this file specifies a username in addition to a host. This username must be a valid username on the local host. When someone does a remote login specifying this username, the user is automatically logged onto the system.

```
rhost1
rhost2
rhost3
rhost9 alfredo
rhost33 wanda
```

Figure 4-5. Example of */etc/hosts.equiv*

4.6 ~/.RHOSTS

The *~/.rhosts* file is essentially a private version of */etc/hosts.equiv* and is used by the same servers. This optional file is located in a user's home directory. With it a user can permit his or her local account to be used by specific remote users on specific remote hosts for autologin to the local host. If a remote host is specified without a username, a user on the remote host must have the same username as the owner of the local *~/.rhosts*. Otherwise the user will be prompted for a login and a password. The format for *~/.rhosts* is the same as for */etc/hosts.equiv*.

4.7 /ETC/FTPUSER

This file contains the names of those to whom *ftp* access is specifically denied. Every time an *ftp* user attempts to login, the *ftp* server searches this file for his or her username. If the name is found, the user is denied access. The following is an example of an *ftpuser* file.

```
fred
wilma
barney
betty
dino
```

Figure 4-6. Example of */etc/ftpusers*

Changes to */etc/ftpusers* must be made by the superuser.

4.8 ~/.NETRC

The *~/.netrc* file contains information that enables a user to automatically login to a remote system as an *ftp* user. Located in a user's home directory, this file lists a hostname and valid username and password for the remote hosts on which that user wishes to have autologin privileges.

machine *threeb21* login *mpd* password *smurf*
machine *threeb22* login *mpd* password *fred*
machine *threeb23* login *mark* password *koala*

Figure 4-7. Example of *.netrc*

Only the official hostnames, and not aliases, can be used in this file. If either *~/.netrc* does not exist or it exists but contains no entry for the currently specified host, the user is prompted for username and password. Since this file contains sensitive information, *ftp* requires that it be owner-readable only. If it is not, an error message is printed and the user is prompted for username and password. If you do not want to include your password in this file, you can omit the password and the word "password" on each line. When a user initiates an ftp connection, the system prompts for the password.

5. SOFTWARE INSTALLATION

The installation of WIN/3B LAN is performed through the UNIX PC window interface.

5.1 PREREQUISITES

Installation requires the following:

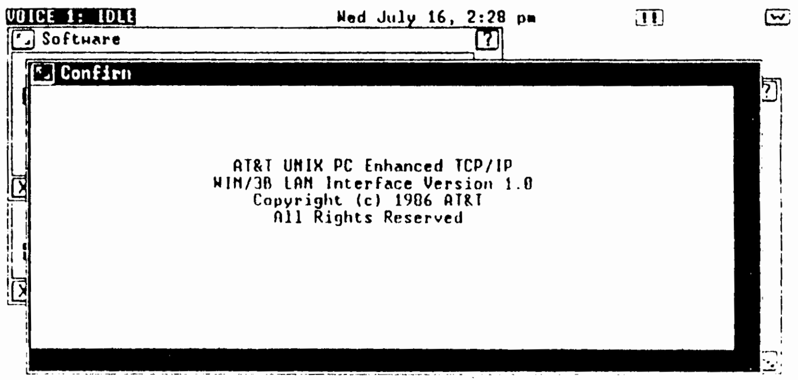
1. An Ethernet network interface board installed on your UNIX PC.
2. The five WIN/3B LAN diskettes, which come in the WIN/3B LAN package.
3. 2500 blocks (where each block is 512 bytes) available in the system.

5.2 INSTALLATION PROCEDURES

1. Log in as **install** using the system console.
2. When the Office of install menu appears, open the Administration window and open *Software Setup*.
3. When the Software window opens, open *Install Software from Floppy*.

   The Confirm window opens containing prompts to guide you through the installation procedure.
4. Insert the first floppy disk and press ENTER.

   When prompted to do so, insert the second floppy disk and press RETURN.
5. Continue this procedure with floppy disks 3, 4 and 5. When the last disk has been read, you will see this screen.



```
AT&T UNIX PC Enhanced TCP/IP
WIN/3B LAN Interface Version 1.0
Copyright (c) 1986 AT&T
All Rights Reserved
```

Figure 5-1. Installation Confirm Screen

6. Remove the floppy disk from the drive.

   When the Ethernet Setup window opens, answer the prompts by entering your network name and number, your host name and internet address, and any aliases for your host. The hostname should be the official name given by the UNIX *uname -n* command. Finally, answer whether or not the driver should be loaded at boot time. Press ENTER.

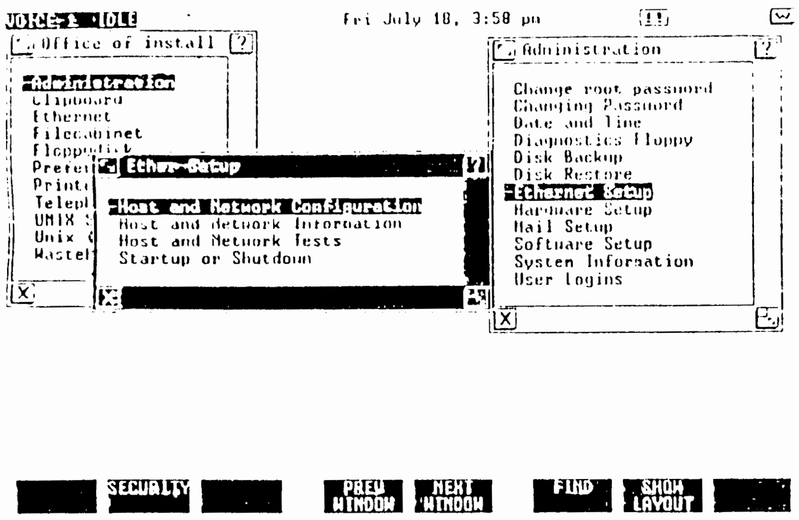


Figure 5-2. Ethernet Setup Screen

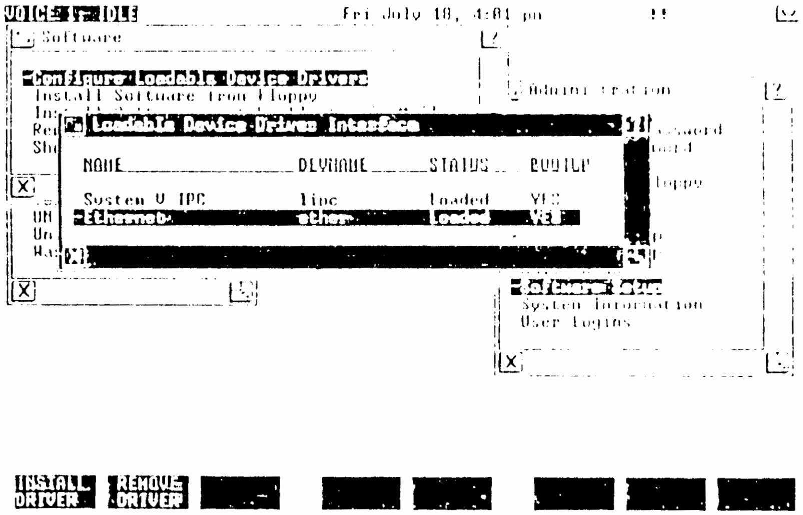
Names of hosts and networks must conform to the following restrictions:

1. Each name is limited to 32 characters in length.
2. Names must not contain any of the following characters: $? { } [ ] * !.

The Internet address myst be unique, if you assign an address that already exists on your network, change your host name internet address entry in /etc/hosts via Host Table Management to make it unique and select "Configure Network Interface."

The network driver will be loaded to complete the installation procedure, and a message will appear indicating that the installation is complete. Press ENTER to return to the Software window.

At this point, you might check to see if the Ethernet device driver has been installed by selecting *Configure Loadable Device Drivers*. Look at the status of the Ethernet driver. It should say "Loaded".



NAME	DEVNAME	STATUS	BUILTIN
System V IPC	ipc	loaded	YES
Ethernet	ether	Loaded	YES

Figure 5-3. Loadable Device Driver Interface Screen

To customize your network, select *Ethernet Setup* in the Administration window. The utilities in the Ether-Setup window automatically edit the database files associated with WIN/3B LAN. These utilities are described in the next chapter, "CONFIGURATION AND MAINTENANCE".

5.3 VERIFICATION

At this point, the network is installed and minimally configured. You will now be able to talk to yourself, but not to other hosts, since no other hosts have yet been made known to your machine. However, before proceeding further it is a good idea to verify that the installation and configuration was performed correctly. The *Host and Network Tests* option in the Ether-Setup window contains a sequence of five tests that exercise the installation. These tests are described in Chapter 6, "CONFIGURATION AND MAINTENANCE". Go through the tests described in Sections 6.3.1 through 6.3.5. Successful passage will assure you that your network is installed properly.

6. CONFIGURATION AND MAINTENANCE

Now that your WIN/3B LAN software has been loaded and tested, to be really useful it must be further configured with information about your network and other hosts accessible through it. Configuration and maintenance are performed by the superuser on the system console from the WIN/3B LAN Ether-Setup window. To get to this window:

1. Log in as **install**.
2. Open *Administration*
3. Open *Ethernet Setup*

All network configuration and maintenance can be performed from this window. All of these options first verify that the network has been initially set up.

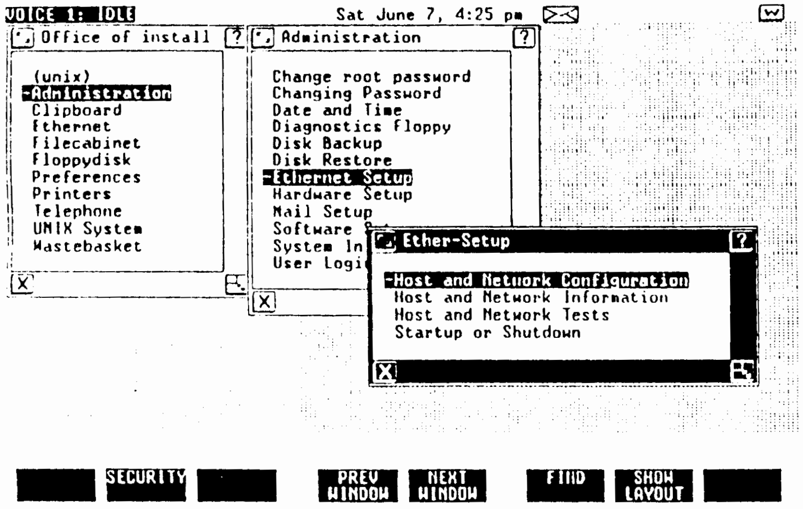


Figure 6-1. Ether-Setup Window

The following sections describe each of the Ether-Setup options. The operations described under each option can also be performed manually using shell commands. To get to the shell, select *UNIX System* from the Office window. The UNIX System window opens showing the shell prompt "$". Enter the appropriate shell command. Press < CTRL D> to return to the Office window.

6.1 HOST AND NETWORK CONFIGURATION

With the options in this window, you can

- add, delete, and display host table entries
- add, delete, and display network table entries
- add, delete, and display routing table entries
- add, delete, and display equivalent-host table entries
- get, set, and delete Ethernet addresses
- configure network interfaces
- start, stop, and show daemons
- enable and disable daemons
- start, stop, and show enabled daemons

Select *Host and Network Configuration* to open the Ether-Config window. Each option in this window is described below.

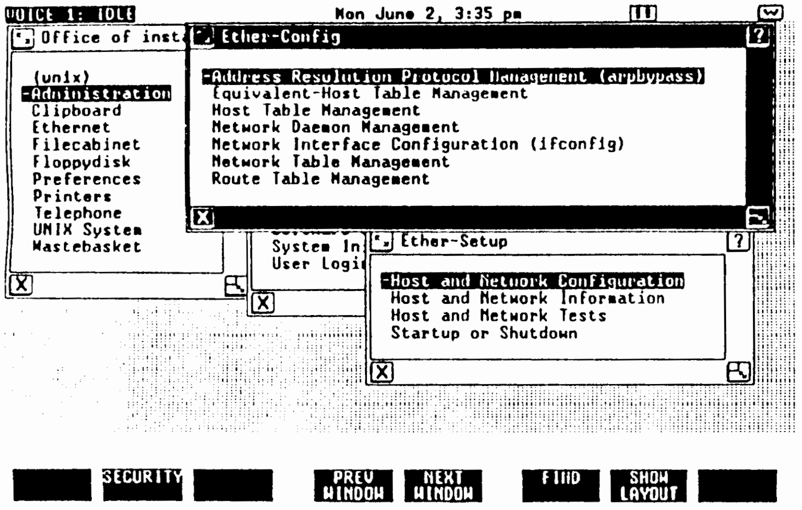


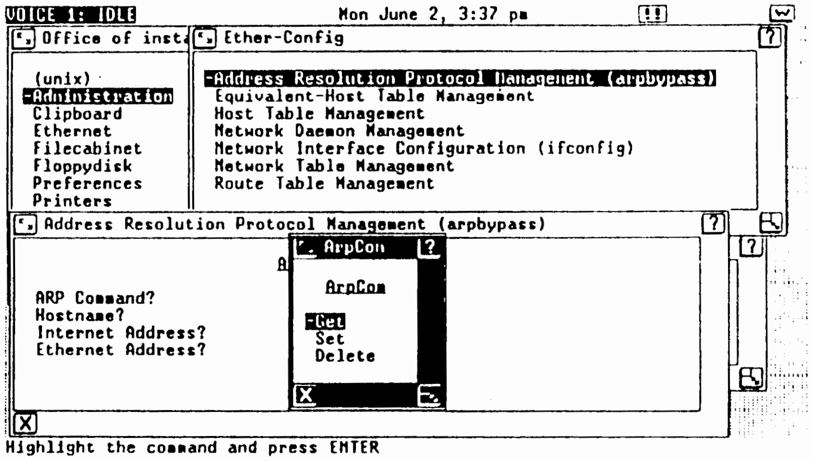
Figure 6-2. Ether-Config Window

6.1.1 Address Resolution Protocol Management (arpbypass)

This option is used to get, set, or delete the Ethernet or Internet address of a known host from the Address Resolution Protocol (ARP) table resident in the kernel. This command only works if ARP is enabled.

The translation of Internet to Ethernet address, and vice versa, is generally handled automatically by ARP, but not all Ethernet drivers support ARP. To communicate with hosts using such drivers it is necessary to manually enter their Ethernet addresses in the ARP table.

To get, set, or delete the Ethernet address of a host, select *Address Resolution Protocol Management (arpbypass)* to open the Address Resolution Protocol Management (arpbypass) window. The ARP Table Entry screen is used for the three *arpbypass* commands, *Get*, *Set*, and *Delete*. Select the appropriate command by pressing the <Cmd> key to open the ArpCom window; highlight the desired command, and press ENTER.

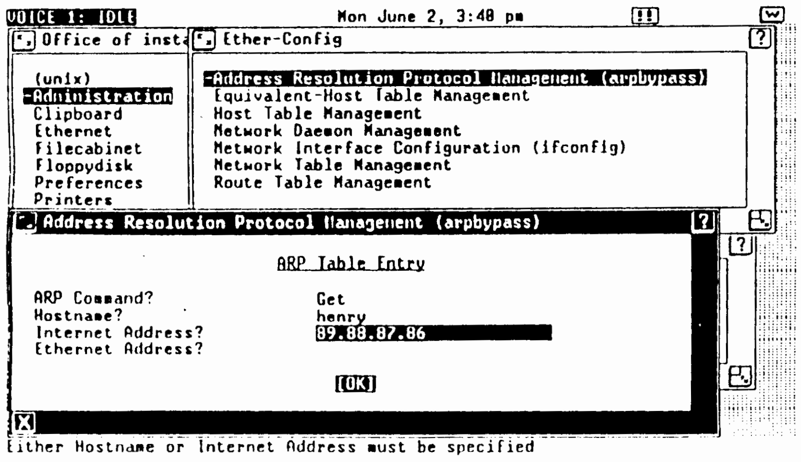


Highlight the command and press ENTER

Figure 6-3. Selecting Commands from the ArpCom Window

6.1.1.1 Get Ethernet Address

To get an Ethernet address for a host, select *Get* from the ArpCom window. Enter either the hostname or the Internet address, and press ENTER. The Internet and Ethernet addresses will appear in hex format next to the prompts.



```
                    ARP Table Entry

ARP Command?                Get
Hostname?                   henry
Internet Address?           89.88.87.86
Ethernet Address?

Either Hostname or Internet Address must be specified
```

Figure 6-4. Getting an Ethernet Address

Notice that the Internet address is four bytes, and the Ethernet address is six bytes long.

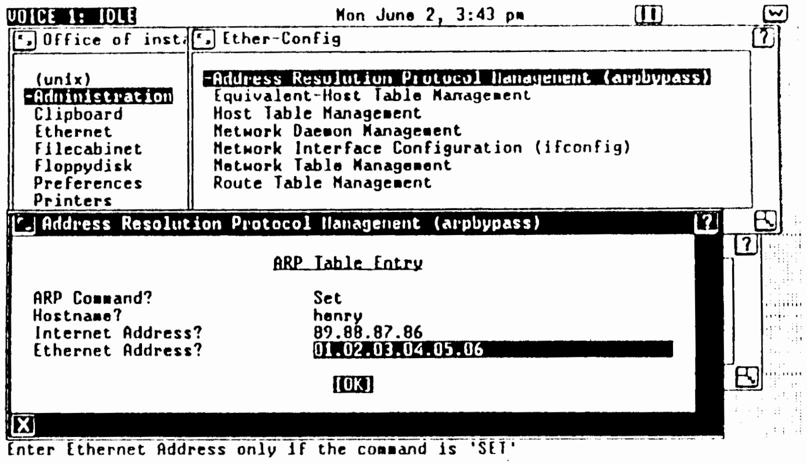
Sometimes this command produces the error: **get: not in ARP table.** Entries in the ARP table are volatile, lasting only twenty minutes. If you get this error, most likely the table entry for the host has been deleted. To make an entry in this table, simply attempt any network operation to this host and a new entry will be made in the table.

6.1.1.2 Set Ethernet Address

If an Ethernet address is incorrect in the ARP table, first

delete the incorrect entry, then use this option to set the correct address.

To set the Ethernet address of a host, select *Set* in the ArpCom window. The Address Resolution Protocol window opens showing the ARP Table Entry screen. Answer the prompts by entering either the hostname or Internet address and the Ethernet address, and press ENTER. The Internet and Ethernet addresses will appear in hex format when the process is complete.



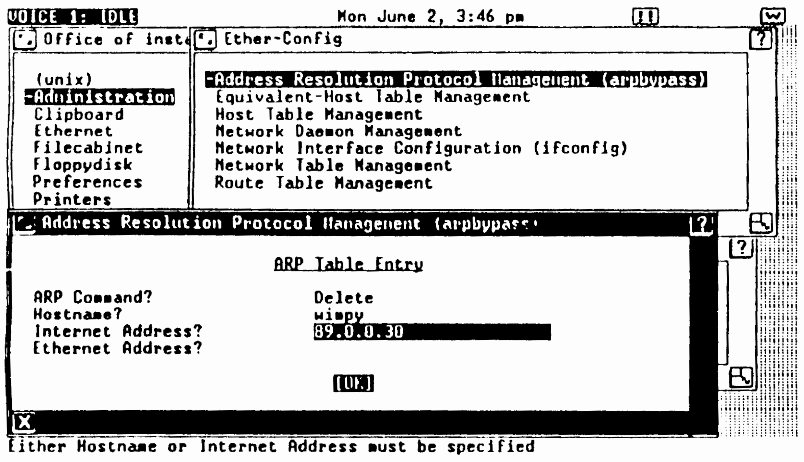
```
                    ARP Table Entry

ARP Command?                Set
Hostname?                   henry
Internet Address?           89.88.87.86
Ethernet Address?           01.02.03.04.05.06

Enter Ethernet Address only if the command is 'SET'
```

Figure 6-5. Setting an Ethernet Address

You can perform this function manually from the shell using the *arpbypass set* command.

6.1.1.3 Delete Ethernet Address

To delete the Ethernet address of a host, select *Delete* from the ArpCom window. Enter either the hostname or Internet address, and press ENTER. The Internet and Ethernet addresses will appear in hex format when the process is complete.



```
ARP Table Entry

ARP Command?          Delete
Hostname?             wimpy
Internet Address?     89.0.0.30
Ethernet Address?

Either Hostname or Internet Address must be specified
```

Figure 6-6. Deleting an Ethernet Address

You can perform this function manually from the shell using the *arpbypass delete* command.

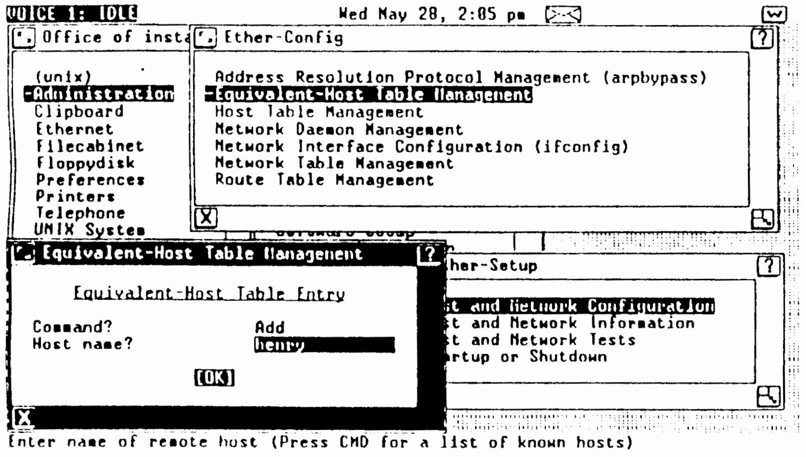
6.1.2 Equivalent-Host Table Management

This option is used to add or delete entries in the equivalent-host table, and to display the contents of the table. Entries in the equivalent-host table allow users on a remote host to log in to the local host automatically, as long as the user's username appears in the local host's */etc/passwd* file.

To add, delete, or display entries in the equivalent-host table, select *Equivalent-Host Table Management* to open the Equivalent-Host Table Management window. The Equivalent-Host Table Entry screen is used for the three equivalent-host table commands, *Add, Delete,* and *Show*. Select the appropriate command by pressing the CMD key to open the Commands window; highlight the desired command, and press ENTER.

6.1.2.1 Add an Equivalent-Host

To add an entry in the equivalent-host table, select *Add* in the Commands window. Enter the name of the remote host on which you have a login account, or press the CMD key to display a list of the current known hosts and select the host you wish to add. Press ENTER to add the equivalent-host to the equivalent-host table.



```
VOICE 1: IDLE          Wed May 28, 2:05 pm
Office of insta  Ether-Config
 (unix)           Address Resolution Protocol Management (arpbypass)
 -Administration  -Equivalent-Host Table Management
 Clipboard        Host Table Management
 Ethernet         Network Daemon Management
 Filecabinet      Network Interface Configuration (ifconfig)
 Floppydisk       Network Table Management
 Preferences      Route Table Management
 Printers
 Telephone
 UNIX System
Equivalent-Host Table Management      her-Setup

   Equivalent-Host Table Entry        t and Network Configuration
                                      t and Network Information
 Command?           Add               t and Network Tests
 Host name?         henry             rtup or Shutdown

               [OK]
Enter name of remote host (Press CMD for a list of known hosts)
```

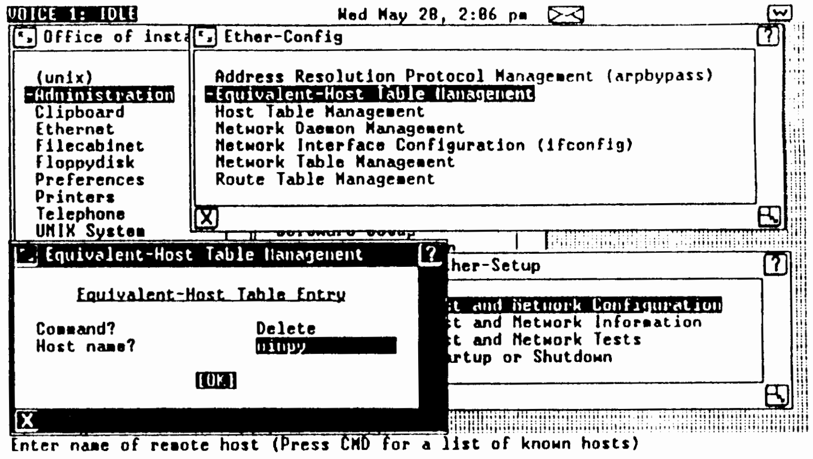
Figure 6-7. Adding an Entry to the Equivalent-Host Table

The Confirm window will display a message confirming that the host database has been updated. Press ENTER to return to the Equivalent-Host Table Entry screen, or CANCL to return to the Ether-Config window. You will receive an error message if you try to add an equivalent host that already exists in the equivalent-host table.

6.1.2.2 Delete an Equivalent-Host

To delete an entry in the equivalent-host table, select *Delete* in the Commands window. Enter the name of the remote host you wish to delete from the table, or press the CMD key to display the remote hosts currently in the equivalent-host table

and select the host you wish to delete. Press ENTER to delete the host.



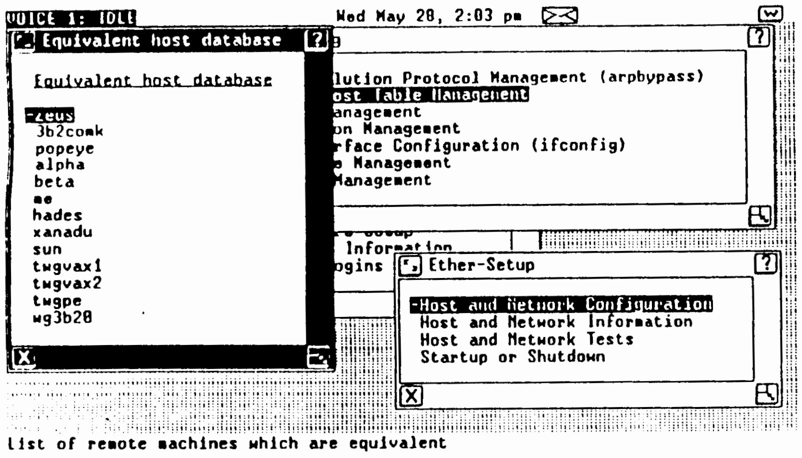
```
VOICE 1: IDLE          Wed May 28, 2:06 pm
Office of insta  Ether-Config
 (unix)           Address Resolution Protocol Management (arpbypass)
 -Administration  -Equivalent-Host Table Management
 Clipboard        Host Table Management
 Ethernet         Network Daemon Management
 Filecabinet      Network Interface Configuration (ifconfig)
 Floppydisk       Network Table Management
 Preferences      Route Table Management
 Printers
 Telephone
 UNIX System
Equivalent-Host Table Management      her-Setup

   Equivalent-Host Table Entry        t and Network Configuration
                                      t and Network Information
 Command?           Delete            t and Network Tests
 Host name?         dinpy             rtup or Shutdown

               [OK]
Enter name of remote host (Press CMD for a list of known hosts)
```

Figure 6-8. Deleting an Entry in the Equivalent-Host Table

The Confirm window will display a message confirming that the host database has been updated. Press ENTER to return to the Equivalent-Host Table Entry screen, or CANCL to return to the Ether-Config window. You will receive an error message if you try to delete an equivalent host that does not exist in the equivalent-host table.

6.1.2.3 Show the Equivalent-Host Table

To display the equivalent-host table, select *Show* in the Commands window and press ENTER. The Equivalent-host database window opens showing the list of remote hosts currently in the equivalent-host table.



```
VOICE 1: IDLE                     Wed May 28, 2:03 pm

Equivalent host database

 Equivalent host database

-zeus
 3b2comk
 popeye
 alpha
 beta
 me
 hades
 xanadu
 sun
 twgvax1
 twgvax2
 twgpe
 wg3b28

lution Protocol Management (arpbypass)
ost Table Management
anagement
on Management
rface Configuration (ifconfig)
o Management
Management

Information
ogins

Ether-Setup
 -Host and Network Configuration
  Host and Network Information
  Host and Network Tests
  Startup or Shutdown

list of remote machines which are equivalent
```

Figure 6-9. Showing the Equivalent-Host Table

6.1.3 Host Table Management

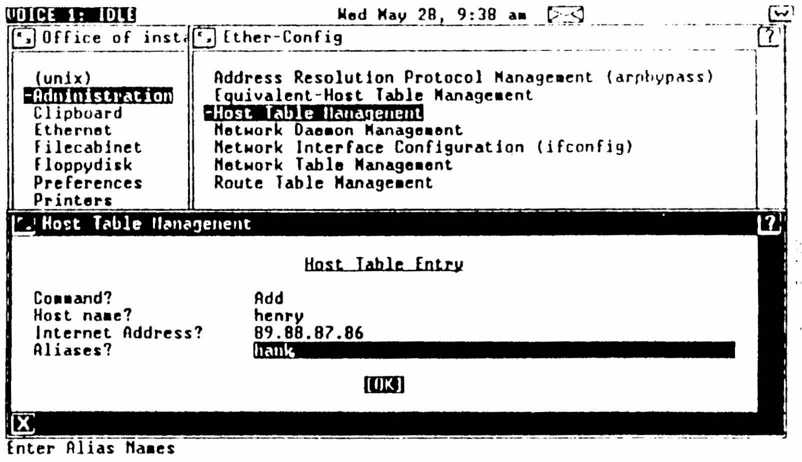
This option is used to add hosts to or delete hosts from the set of hosts known to your system, and to display the host table.

To add or delete a host, or to display the host table, select *Host Table Management* to open the Host Table Management window. The Host Table Entry screen is used for the three host table management commands, *Add*, *Delete*, and *Show*. Select the appropriate command by pressing the CMD key to open the Commands window; highlight the desired command,

and press ENTER.

6.1.3.1 Add a Remote Host

To add a remote host to the host table, select *Add* in the Commands window. Answer the prompts by entering the official host name which must correspond to the output of the *uname -n* command, the Internet address, and any aliases for the new host. (See the section on Internet Addressing in the first chapter for an explanation of the digits in the Internet address.) Press ENTER to add the host.



```
VOICE 1: IDLE                     Wed May 28, 9:38 am

Office of inst   Ether-Config

 (unix)            Address Resolution Protocol Management (arpbypass)
-Administration    Equivalent-Host Table Management
 Clipboard        -Host Table Management
 Ethernet          Network Daemon Management
 Filecabinet       Network Interface Configuration (ifconfig)
 Floppydisk        Network Table Management
 Preferences       Route Table Management
 Printers

Host Table Management

                        Host Table Entry

 Command?               Add
 Host name?             henry
 Internet Address?      89.88.87.86
 Aliases?               hank

                          [OK]

Enter Alias Names
```

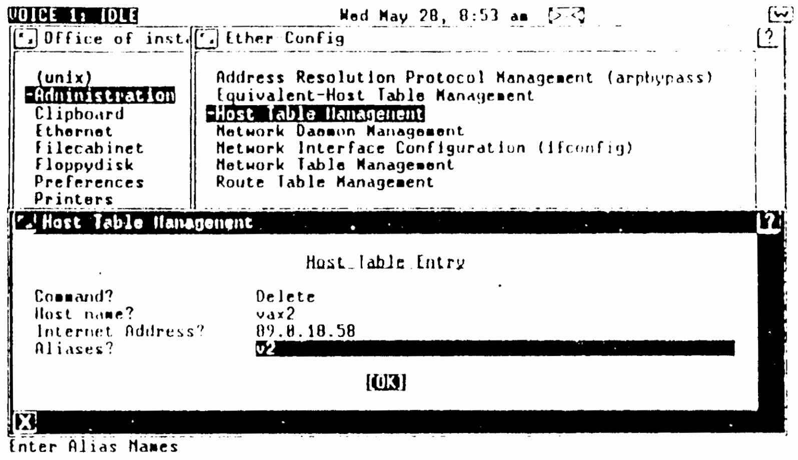
Figure 6-10. Adding an Entry to the Host Table

The Confirm window will display a message confirming that the host table has been updated. Press ENTER to return to the Host Table Entry screen, or CANCL to return to the Ether-Config window. You will receive an error message if you try to add a host that already exists in the host table.

This procedure adds the remote host to the */etc/hosts* file.

6.1.3.2 Delete a Remote Host

To delete a remote host from the set of hosts known to your system, select *Delete* in the Commands window. Enter the name and Internet address of the host to delete, or press the CMD key to display the list of known hosts and their Internet addresses and select the host you wish to delete. Press ENTER to delete the host from the host table.



```
                Host Table Entry

Command?               Delete
Host name?             vax2
Internet Address?      89.0.10.58
Aliases?               v2
```

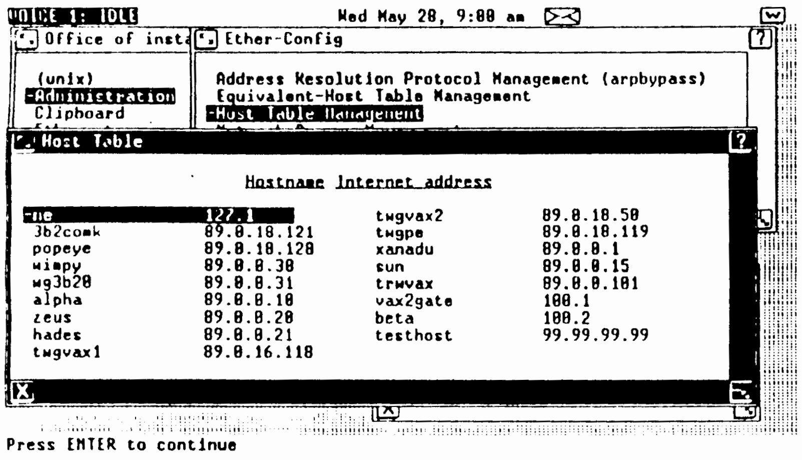
Enter Alias Names

Figure 6-11. Deleting an Entry in Host Table Entry

The Confirm window will display a message confirming that the host table has been updated. Press ENTER to return to the Host Table Entry screen, or CANCL to return to the Ether-Config window. You will receive an error message if you try to delete a host that does not exist in the host table.

6.1.3.3 Showing the Host Table

To display the list of remote hosts known to your system, select *Show* in the Commands window. Press ENTER to display the host table. The Host Table window opens showing the names and Internet addresses of all hosts currently known to your system. Selecting "Show" will cause all hosts entered via the User Agent to be displayed. If any hosts have been entered by editing /etc/hosts directly, they will not appear in user agent menus. Add them to the /usr/ethernet/ua/HostTable file.



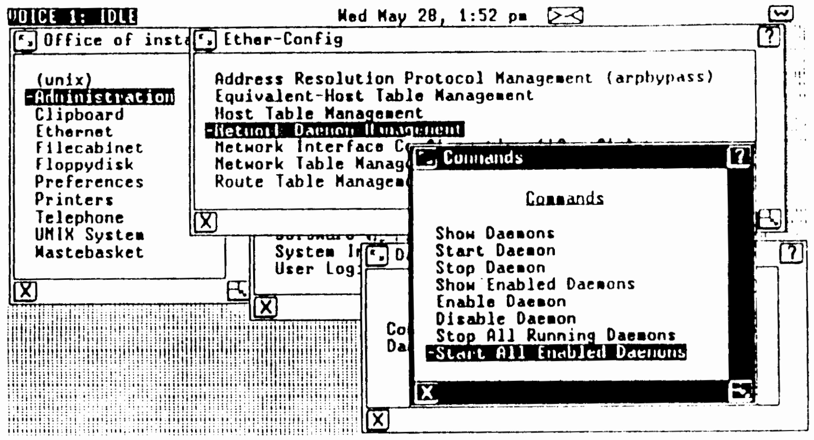
```
VOICE 1: IDLE                  Wed May 28, 9:00 am

Office of insta   Ether-Config

 (unix)            Address Resolution Protocol Management (arpbypass)
-Administration    Equivalent-Host Table Management
 Clipboard        -Host Table Management

Host Table

                  Hostname Internet address

-ne               127.1               twgvax2             89.0.10.50
 3b2comk          89.0.10.121         twgpe               89.0.10.119
 popeye           89.0.10.120         xanadu              89.0.0.1
 wimpy            89.0.0.30           sun                 89.0.0.15
 wg3b20           89.0.0.31           trwvax              89.0.0.101
 alpha            89.0.0.10           vax2gate            100.1
 zeus             89.0.0.20           beta                100.2
 hades            89.0.0.21           testhost            99.99.99.99
 twgvax1          89.0.16.110

Press ENTER to continue
```

Figure 6-12. Showing the Host Table

6.1.4 Network Daemon Management

This option is used to start, stop, enable, disable, and display network daemons. Select *Network Daemon Management* to open the Daemon Management window. The Daemon Management screen is used for all daemon management commands. Select the appropriate command by pressing the CMD key to open the Commands window; highlight the desired command, and press ENTER. By default the "rwho", "tftpd", "rexecd", and

"routed" daemons are not enabled on installation.

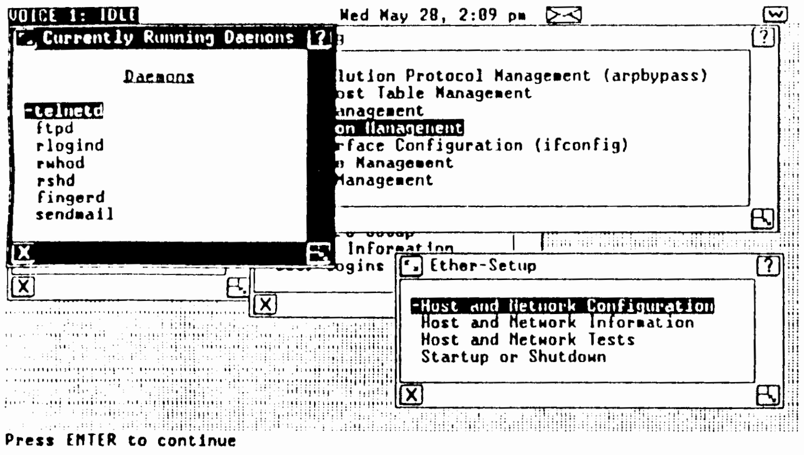


```
VOICE 1: IDLE                  Wed May 28, 1:52 pm

Office of insta   Ether-Config

 (unix)            Address Resolution Protocol Management (arpbypass)
-Administration    Equivalent-Host Table Management
 Clipboard         Host Table Management
 Ethernet         -Network Daemon Management
 Filecabinet       Network Interface Co
 Floppydisk        Network Table Manag
 Preferences       Route Table Managem
 Printers
 Telephone                   Commands
 UNIX System
 Wastebasket                 Commands

                   Show Daemons
                   Start Daemon
                   Stop Daemon
                   Show Enabled Daemons
                   Enable Daemon
                   Disable Daemon
                   Stop All Running Daemons
                  -Start All Enabled Daemons
```

Figure 6-13. Network Daemon Management Commands

6.1.4.1 Show Daemons

This command displays all currently running daemons. Select *Show Daemons* in the Commands window. Press ENTER to display the list of currently running daemons.

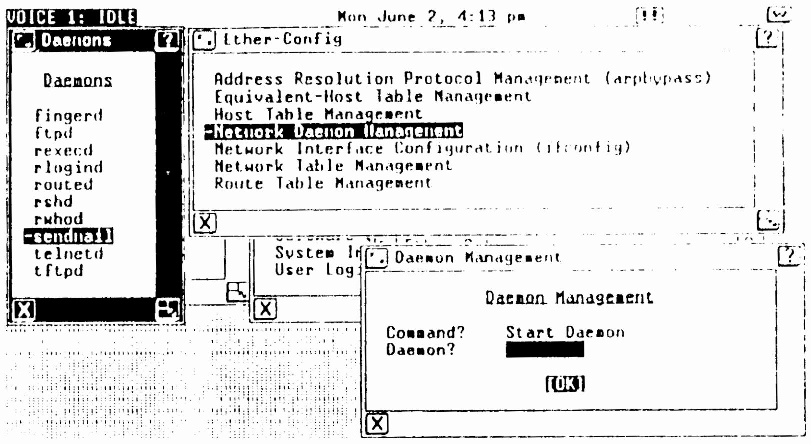


```
VOICE 1: IDLE                 Wed May 28, 2:09 pm
Currently Running Daemons
        Daemons
-telnetd
 ftpd
 rlogind
 rwhod
 rshd
 fingerd
 sendmail

lution Protocol Management (arpbypass)
ost Table Management
anagement
on Management
rface Configuration (ifconfig)
n Management
Management

 Information
ogins
Ether-Setup
-Host and Network Configuration
 Host and Network Information
 Host and Network Tests
 Startup or Shutdown
Press ENTER to continue
```

Figure 6-14. Showing the Currently Running Daemons

6.1.4.2 Start Daemon

To start a daemon, select *Start Daemon* in the Commands window. Enter the name of the daemon you wish to start, or press CMD to display a list of daemons and select the daemon you wish to start. Press ENTER to start the daemon.



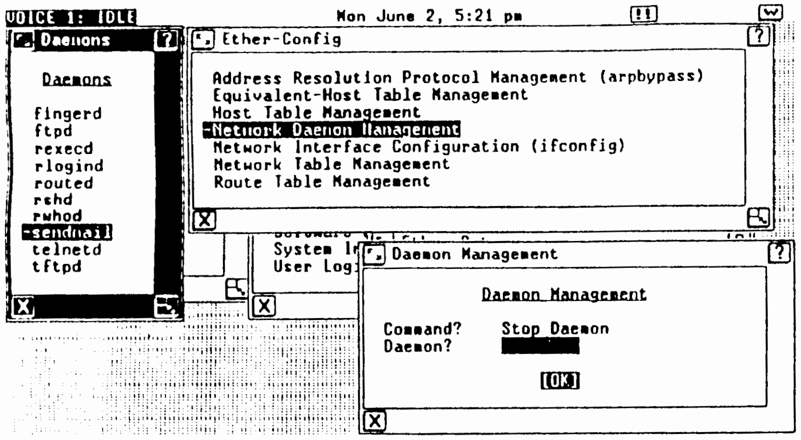
```
VOICE 1: IDLE                 Mon June 2, 4:13 pm
Daemons          Ether-Config
  Daemons          Address Resolution Protocol Management (arpbypass)
                   Equivalent-Host Table Management
 fingerd           Host Table Management
 ftpd             -Network Daemon Management
 rexecd            Network Interface Configuration (ifconfig)
 rlogind           Network Table Management
 routed            Route Table Management
 rshd
 rwhod
-sendmail          System In
 telnetd           User Log
 tftpd
                   Daemon Management
                        Daemon Management
                   Command?     Start Daemon
                   Daemon?
                               [OK]
```

Figure 6-15. Starting the *sendmail* Daemon

The Confirm window will display a message confirming that the daemon has been started. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to start a daemon that does not exist or is already started.

6.1.4.3 Stop Daemon

To stop a daemon, select *Stop Daemon* in the Commands window. Enter the name of the daemon you wish to stop, or press CMD to display a list of daemons and highlight the daemon you wish to stop. Press ENTER to stop the daemon.



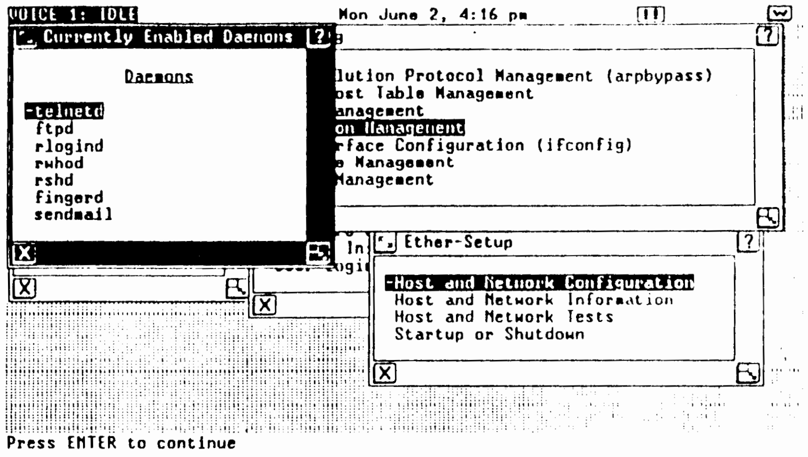
```
VOICE 1: IDLE                 Mon June 2, 5:21 pm
Daemons          Ether-Config

Daemons          Address Resolution Protocol Management (arpbypass)
                 Equivalent-Host Table Management
fingerd          Host Table Management
ftpd             -Network Daemon Management
rexecd           Network Interface Configuration (ifconfig)
rlogind          Network Table Management
routed           Route Table Management
rshd
rwhod
-sendmail        System In   Daemon Management
telnetd          User Log
tftpd                        Daemon Management

                             Command?   Stop Daemon
                             Daemon?

                                        [OK]
```

Figure 6-16. Stopping the *sendmail* Daemon

The Confirm window will display a message confirming that the daemon has been stopped. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to stop a daemon that does not exist or is already stopped.

6.1.4.4 Show Enabled Daemons

To display the list of currently enabled daemons, select *Show Enabled Daemons* in the Commands window. Press ENTER to display the currently enabled daemons.



```
VOICE 1: IDLE                 Mon June 2, 4:16 pm
Currently Enabled Daemons

     Daemons           lution Protocol Management (arpbypass)
                       ost Table Management
-telnetd               anagement
 ftpd                  on Management
 rlogind               rface Configuration (ifconfig)
 rwhod                 e Management
 rshd                  Management
 fingerd
 sendmail
                       Ether-Setup

                       -Host and Network Configuration
                       Host and Network Information
                       Host and Network Tests
                       Startup or Shutdown

Press ENTER to continue
```

Figure 6-17. Showing the Currently Enabled Daemons

6.1.4.5 Enable Daemon

To enable a daemon, select *Enable Daemon* in the Commands window. Enter the name of the daemon you wish to enable, or press the CMD key to display the list of daemons and select the daemon you wish to enable. Press ENTER to enable the daemon.

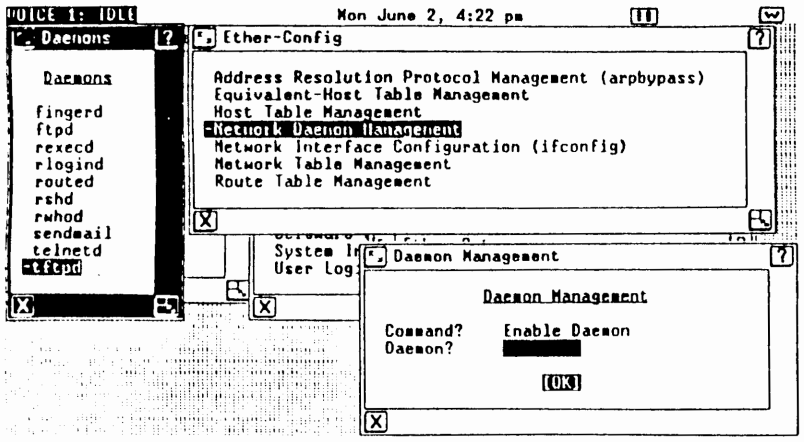


Figure 6-18. Enabling the *tfpd* Daemon

The Confirm window will display a message confirming that the daemon has been enabled. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to enable a daemon that does not exist or is already enabled.

6.1.4.6 Disable Daemon

To disable a daemon, select *Disable Daemon* in the Commands window. Enter the name of the daemon you wish to disable, or press the CMD key to display the list of daemons and select the daemon you wish to disable. Press ENTER to disable the daemon.

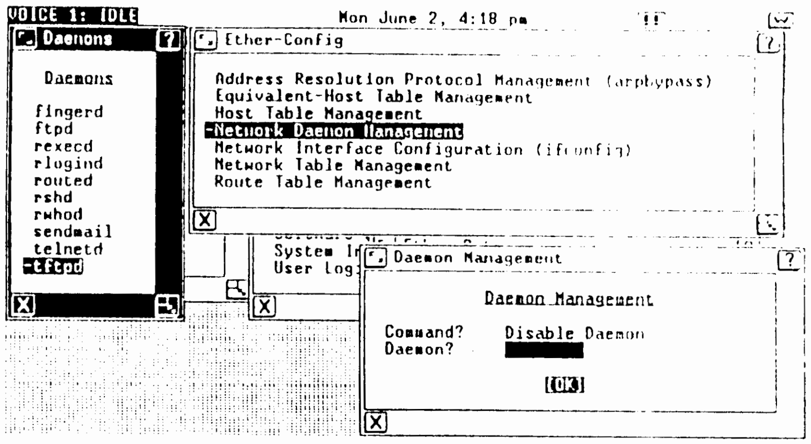
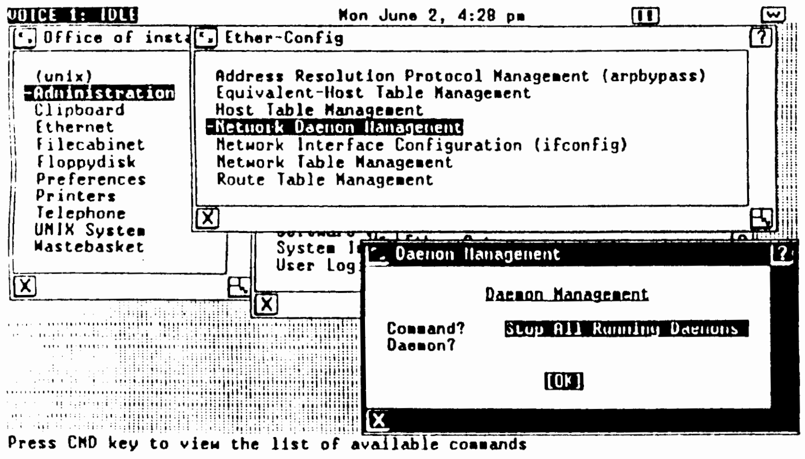


Figure 6-19. Disabling the *tfpd* Daemon

The Confirm window will display a message confirming that the daemon has been disabled. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to disable a daemon that does not exist or is already disabled.

6.1.4.7 Stop All Running Daemons

To stop all currently running daemons, select *Stop All Running Daemons* in the Commands window.



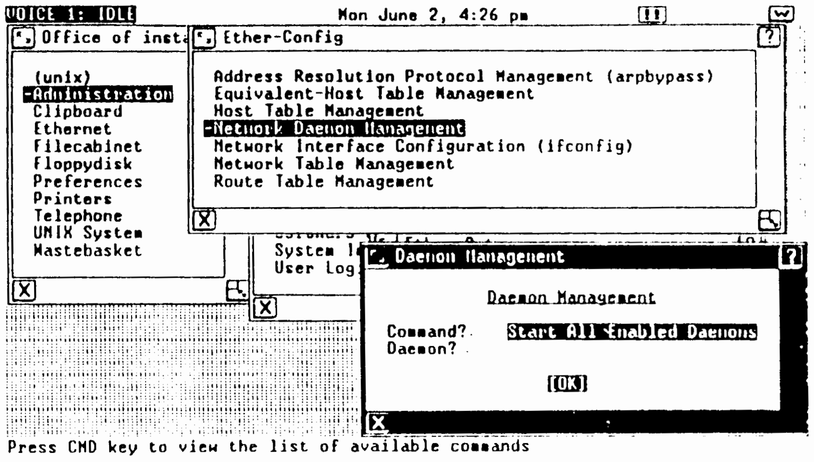
```
VOICE 1: IDLE                 Mon June 2, 4:28 pm
Office of inst   Ether-Config
 (unix)          Address Resolution Protocol Management (arpbypass)
-Administration  Equivalent-Host Table Management
 Clipboard       Host Table Management
 Ethernet        -Network Daemon Management
 Filecabinet     Network Interface Configuration (ifconfig)
 Floppydisk      Network Table Management
 Preferences     Route Table Management
 Printers
 Telephone
 UNIX System     System I   Daemon Management
 Wastebasket     User Log
                             Daemon Management
                  Command?   Stop All Running Daemons
                  Daemon?
                                   [OK]
Press CMD key to view the list of available commands
```

Figure 6-20. Stopping All Running Daemons

The Confirm window will display a message confirming that all network daemons have been stopped. Press ENTER to continue.

6.1.4.8 Start All Enabled Daemons

To start all enabled daemons, select *Start All Enabled Daemons* in the Commands window.



```
VOICE 1: IDLE                 Mon June 2, 4:26 pm
Office of inst   Ether-Config
 (unix)          Address Resolution Protocol Management (arpbypass)
-Administration  Equivalent-Host Table Management
 Clipboard       Host Table Management
 Ethernet        -Network Daemon Management
 Filecabinet     Network Interface Configuration (ifconfig)
 Floppydisk      Network Table Management
 Preferences     Route Table Management
 Printers
 Telephone
 UNIX System     System I   Daemon Management
 Wastebasket     User Log
                             Daemon Management
                  Command?   Start All Enabled Daemons
                  Daemon?
                                   [OK]
Press CMD key to view the list of available commands
```

Figure 6-21. Starting All Enabled Daemons

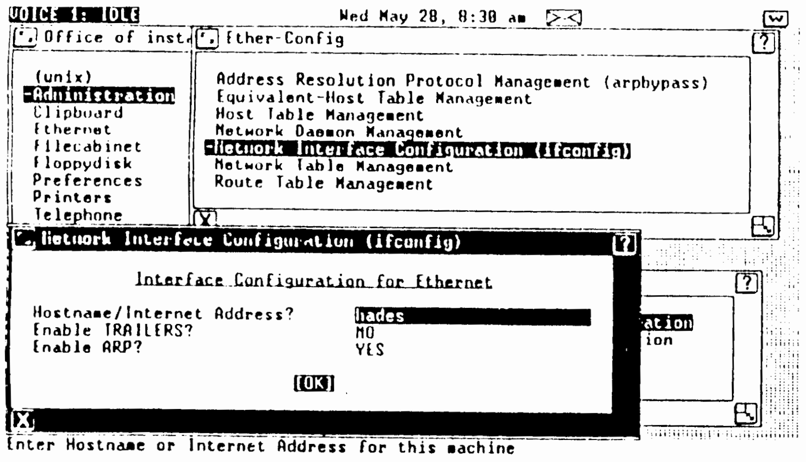
The Confirm window will display a message confirming that all network daemons have been started. Press ENTER to continue.

6.1.5 Network Interface Configuration (ifconfig)

This option is used to manipulate or configure a network interface. It can also be used to enable or disable a network interface, or to toggle ARP or trailer processing by an interface.

This option makes use of the *ifconfig* command which is invoked by */etc/lddrv/ether.rc* at boot time to bring up all network interfaces.

To configure an interface, select *Network Interface Configuration (ifconfig)* to open the Network Interface Configuration (ifconfig) window. Enter the hostname or Internet address of your machine. Answer the next two prompts to enable or disable TRAILERS and ARP. Press ENTER.



```
Interface Configuration for Ethernet

Hostname/Internet Address?      hades
Enable TRAILERS?                NO
Enable ARP?                     YES

Enter Hostname or Internet Address for this machine
```

Figure 6-22. Configuring Network Interfaces

The Internet address must be unique, if you assign an address that already exists on your network, change your host name internet address entry in /etc/hosts via Host Table Management to make it unique and select "Configure Network Interface."

6.1.5.1 Enabling/Disabling an Interface

The *Netstat -i* command shows which interfaces are down by placing an asterisk next to the name of the driver.

```
#netstat -i

Name   Mtu    Network    Address   Ipkts   Ierrs   Opkts   Oerrs   Colls
en0*   1500   Peg        3b2       7199    0       4431    0       0
lo0    1536   Loopback   me        0       0       0       0       0
```

Figure 6-23. Output of *netstat -i* Command

To turn the interface on, use *ifconfig* with the "up" parameter.

```
#/usr/bin/ifconfig en0 up
```

In some instances, you may wish to suspend a host from the network. There are a number of ways to accomplish this, but the quickest is to disable the interface to that network.

```
#/usr/bin/ifconfig en0 down
```

In this example the Ethernet driver is disabled, effectively suspending the host from the network.

6.1.5.2 Enabling/Disabling ARP

ARP is normally enabled when the network driver is loaded by an entry in the */etc/lddrv/ether.rc* file. For instance, to enable ARP processing, run the following command:

```
#/usr/bin/ifconfig en0 arp
```

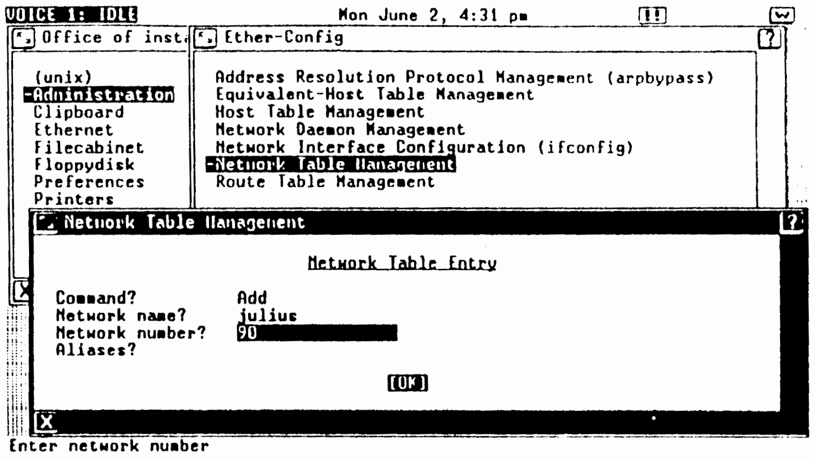
6.1.6 Network Table Managment

This option is used to add and delete entries in the network table, and to display the list of currently known networks.

To add or delete networks, or to display the currently known networks, select *Network Table Management* to open the Network Table Management window and show the Network Table Entry screen. This screen is used for the three Network Table Management commands, *Add*, *Delete*, and *Show*. Select the appropriate command by pressing the CMD key to open the Commands window; highlight the desired command, and press ENTER.

6.1.6.1 Add a Network

To add a network to the list of known networks, select *Add* in the Commands window. Answer the prompts on the Network Table Entry screen; enter the name and network number of the new network, and any aliases. Press ENTER to add the network.



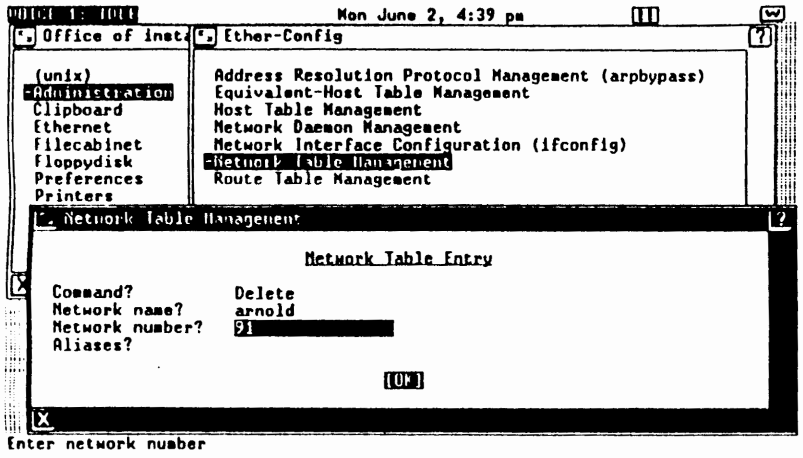
```
VOICE 1: IDLE                Mon June 2, 4:31 pm
Office of inst   Ether-Config
(unix)           Address Resolution Protocol Management (arpbypass)
-Administration  Equivalent-Host Table Management
Clipboard        Host Table Management
Ethernet         Network Daemon Management
Filecabinet      Network Interface Configuration (ifconfig)
Floppydisk       -Network Table Management
Preferences      Route Table Management
Printers
Network Table Management
                 Network Table Entry
Command?         Add
Network name?    julius
Network number?  90
Aliases?
                     OK
Enter network number
```

Figure 6-24. Adding a Network Table Entry

The Confirm window will display a message confirming that the network database has been updated. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to add a network that already exists.

6.1.6.2 Delete a Network

To delete a network from the list of known networks, select *Delete* in the Commands window. Enter the name and network number of the network to delete, and press ENTER.



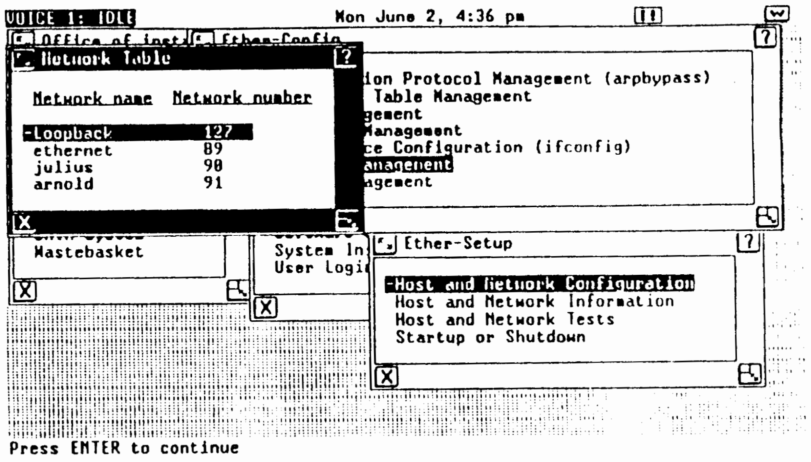
```
VOICE 1: IDLE                    Mon June 2, 4:39 pm
Office of inst  Ether-Config

 (unix)          Address Resolution Protocol Management (arpbypass)
 -Administration Equivalent-Host Table Management
 Clipboard       Host Table Management
 Ethernet        Network Daemon Management
 Filecabinet     Network Interface Configuration (ifconfig)
 Floppydisk      -Network Table Management
 Preferences     Route Table Management
 Printers

 Network Table Management

                     Network Table Entry

   Command?          Delete
   Network name?     arnold
   Network number?   91
   Aliases?

                          [OK]
Enter network number
```

Figure 6-25. Deleting a Network Table Entry

The Confirm window will display a message confirming that the network database has been updated. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to delete a network that does not exist.

6.1.6.3 Show Known Networks

To display the list of known networks, select *Show* in the Commands window, and press ENTER. The Network Table window opens displaying the list of networks and network numbers.



```
VOICE 1: IDLE                    Mon June 2, 4:36 pm
 Network Table
                                 ion Protocol Management (arpbypass)
  Network name   Network number   Table Management
                                 gement
 -Loopback        127            Management
  ethernet        89             ce Configuration (ifconfig)
  julius          90             anagement
  arnold          91             agement

  Wastebasket       System In  Ether-Setup
                    User Logi
                               -Host and Network Configuration
                                Host and Network Information
                                Host and Network Tests
                                Startup or Shutdown

Press ENTER to continue
```

Figure 6-26. Showing the Network Table

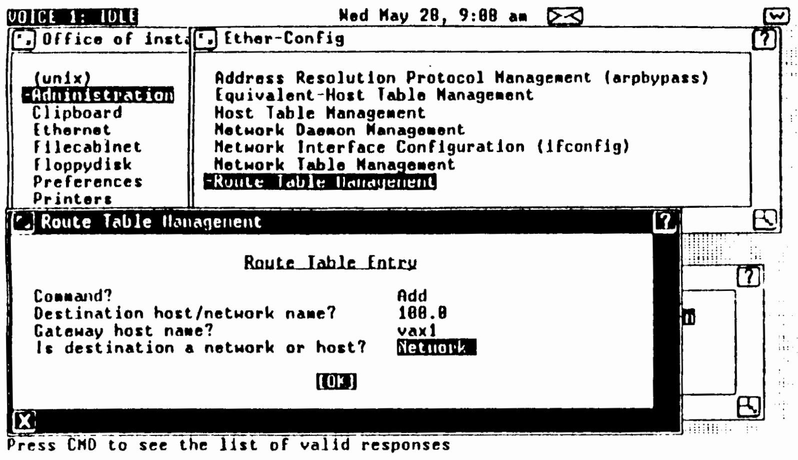
6.1.7 Route Table Management

This option enables you to add a route entry to, delete an entry from, or display the network routing tables resident in the UNIX kernel.

To add a route to or delete a route from the network, or display the route table, select *Route Table Management* to open the Route Table Management window. The Route Table Entry screen is used for the three route table management commands, *Add*, *Delete*, and *Show*. Select the appropriate command by pressing the CMD key to open the Commands window; highlight the desired command, and press ENTER.

6.1.7.1 Add a Routing Table Entry

To add a route, select *Add* in the Commands window. Answer the other prompts by entering the name of the destination host or network, the name of the gateway host, and whether the destination is a network or host. Press ENTER.

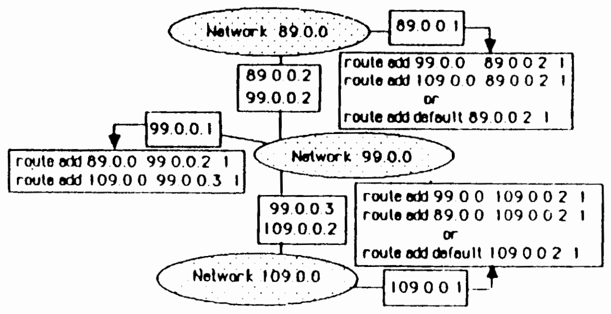


```
VOICE 1: IDLE                     Wed May 28, 9:08 am
Office of inst   Ether-Config

  (unix)           Address Resolution Protocol Management (arpbypass)
 -Administration   Equivalent-Host Table Management
  Clipboard        Host Table Management
  Ethernet         Network Daemon Management
  Filecabinet      Network Interface Configuration (ifconfig)
  Floppydisk       Network Table Management
  Preferences     -Route Table Management
  Printers
Route Table Management

                       Route Table Entry

 Command?                               Add
 Destination host/network name?         100.0
 Gateway host name?                     vax1
 Is destination a network or host?      Network

                          [OK]

Press CMD to see the list of valid responses
```

Figure 6-27. Adding a Routing Table Entry

The Confirm window will display a message confirming that the route table has been updated. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to add a route that already exists.

In the following example, these networks are interconnected via two gateways.



```
route add 99.0.0   89.0.0.2 1
route add 109.0.0  89.0.0.2 1
          or
route add default 89.0.0.2 1
```

```
route add 89.0.0   99.0.0.2 1
route add 109.0.0  99.0.0.3 1
```

```
route add 99.0.0  109.0.0.2 1
route add 89.0.0  109.0.0.2 1
          or
route add default 109.0.0.2 1
```

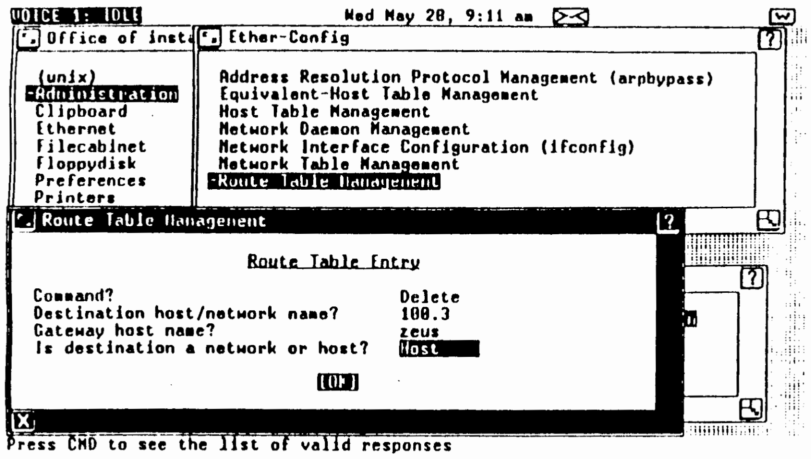
Figure 6-28. Adding Routes - an Example

Each gateway has two addresses, one for each network to which it's connected. Each network has a host which wishes to communicate with the hosts on the other networks. Shown in boxes are the commands the administrator of each host will need to enter to enable these hosts to communicate. Were this a regular requirement, these routing commands should be added to the network startup file */etc/lddrv/ether.rc*.

6.1.7.2 Delete a Routing Table Entry

This option enables you to delete a route from the network routing tables in the UNIX kernel.

To delete a route, select *Delete* in the Commands window. Enter the destination host or network name to delete, or press CMD to display the list of network routes and select the host or network name you wish to delete. Enter the gateway host name and whether the destination is a network or host. Press ENTER to delete the destination from the routing tables.



```
VOICE 1: IDLE              Wed May 28, 9:11 am

Office of inst   Ether-Config

 (unix)            Address Resolution Protocol Management (arpbypass)
-Administration    Equivalent-Host Table Management
 Clipboard         Host Table Management
 Ethernet          Network Daemon Management
 Filecabinet       Network Interface Configuration (ifconfig)
 Floppydisk        Network Table Management
 Preferences      -Route Table Management
 Printers

Route Table Management

                      Route Table Entry

Command?                                  Delete
Destination host/network name?            100.3
Gateway host name?                        zeus
Is destination a network or host?         Host

                           [OK]

Press CMD to see the list of valid responses
```

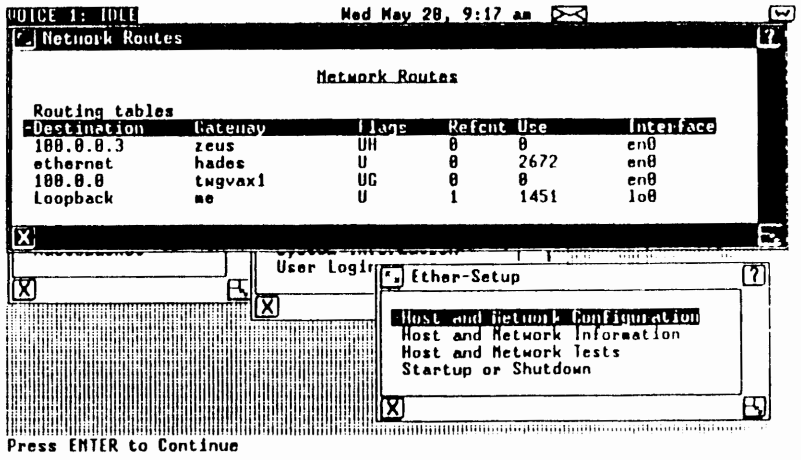
Figure 6-29. Deleting a Routing Table Entry

The Confirm window will display a message confirming that the route table has been updated. Press ENTER to continue, or CANCL to return to the Ether-Config window. You will receive an error message if you try to delete a route that does not exist in the route table.

You can manually delete a route using the *route* command.

6.1.7.3 Show the Routing Table

To display the network routes, select *Show* in the Commands window. Press ENTER to display the current network routes.



```
VOICE 1: IDLE              Wed May 28, 9:17 am

Network Routes

                        Network Routes

Routing tables
-Destination     Gateway          Flags    Refcnt  Use      Interface
 100.0.0.3       zeus             UH       0       0        en0
 ethernet        hades            U        0       2672     en0
 100.0.0         twgvax1          UG       0       0        en0
 Loopback        me               U        1       1451     lo0

User Logi   Ether-Setup

           -Host and Network Configuration
            Host and Network Information
            Host and Network Tests
            Startup or Shutdown

Press ENTER to Continue
```

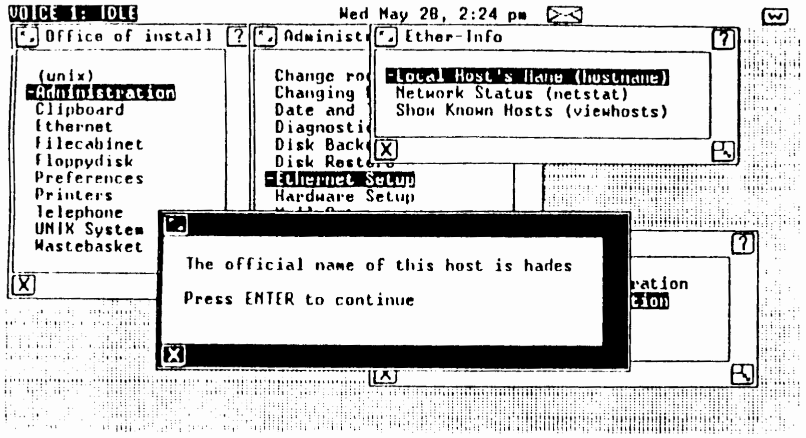
Figure 6-30. Showing the Routing Table

6.2 HOST AND NETWORK INFORMATION

The options in this window provide information about the local host's name, the status of various network-related data structures, and the hosts on the network. These options are used primarily to monitor and maintain your network.

6.2.1 Local Host's Name (hostname)

This option prints the name of the current host. This is the official hostname reported by the UNIX command *uname -n*.



```
The official name of this host is hades

Press ENTER to continue
```

Figure 6-31. Show the Local Host's Name

6.2.2 Network Status (netstat)

This option opens the NetStat window, which contains options for displaying the contents of various network-related data structures. It is the best source of information about the status of your network.

Select *Network Status (netstat)* to open the NetStat window. Each option in this window is described in the sections below.

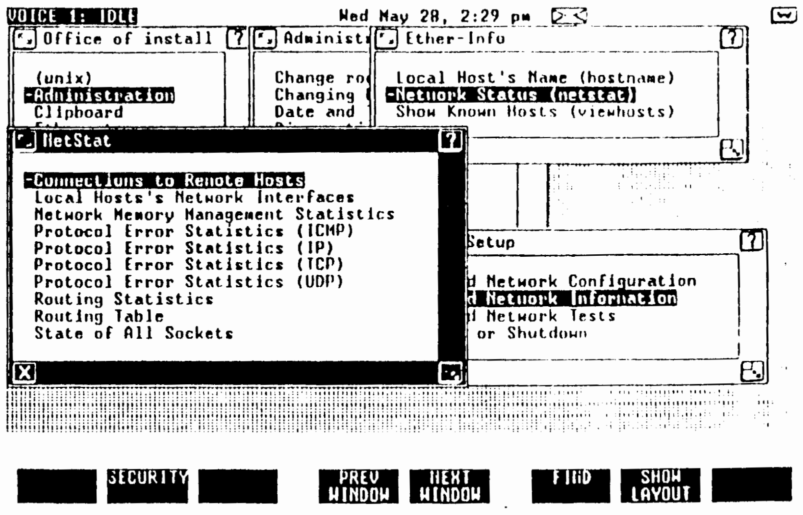
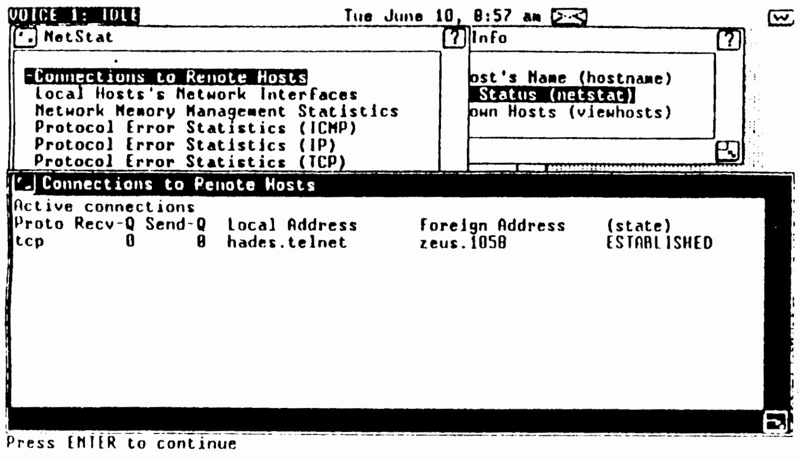

VOICE 1: IDLE
Wed May 28, 2:29 pm
Office of install
(unix)
-Administration
Clipboard
Administ
Change ro
Changing
Date and
Ether-Info
Local Host's Name (hostname)
-Network Status (netstat)
Show Known Hosts (viewhosts)
NetStat
-Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Protocol Error Statistics (UDP)
Routing Statistics
Routing Table
State of All Sockets
Setup
Network Configuration
Network Information
Network Tests
or Shutdown
SECURITY
PREV WINDOW
NEXT WINDOW
FIND
SHOW LAYOUT

Figure 6-32. NetStat Window

6.2.2.1 Connections to Remote Hosts

This option displays the status of the active connections to remote hosts. Servers are not shown. It displays network addresses symbolically, that is, by their associated names, rather than in Internet address format.

The first column indicates the protocol associated with a given socket. The second and third columns indicate the current length of the socket queues. The fourth and fifth columns show the local and remote addresses of the connection with which the socket is associated. The sixth column shows the state of the connection. Note that all the daemons are in the "listen" state and that their foreign addresses are specified by the shell wildcard "*".



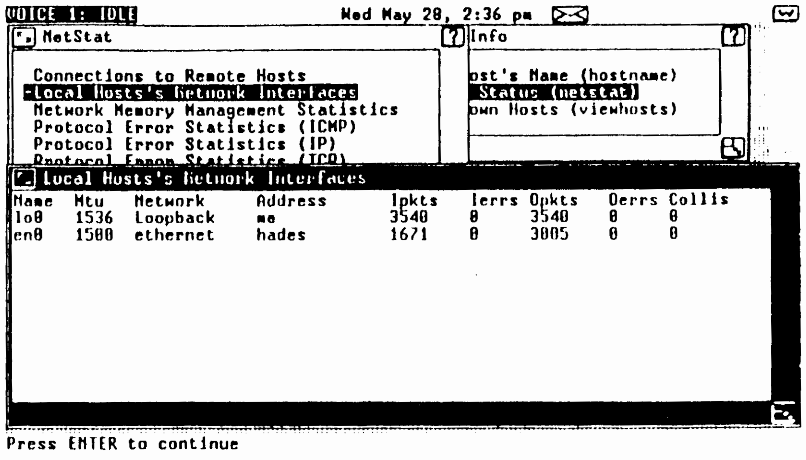
```
Connections to Remote Hosts
Active connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp        0      0  hades.telnet           zeus.1058              ESTABLISHED
```

VOICE 1: IDLE
Tue June 10, 8:57 am
NetStat
-Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Info
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Press ENTER to continue

Figure 6-33. Connections to Remote Hosts

6.2.2.2 Local Hosts's Network Interfaces

This option shows which media drivers, or network interfaces, are configured into the local host kernel.



```
Local Hosts's Network Interfaces
Name   Mtu    Network     Address      Ipkts   Ierrs Opkts   Oerrs Collis
lo0    1536   Loopback    me           3540    0     3540    0     0
en0    1500   ethernet    hades        1671    0     3005    0     0
```

VOICE 1: IDLE
Wed May 28, 2:36 pm
NetStat
Connections to Remote Hosts
-Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Info
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Press ENTER to continue

Figure 6-34. Local Host's Network Interfaces

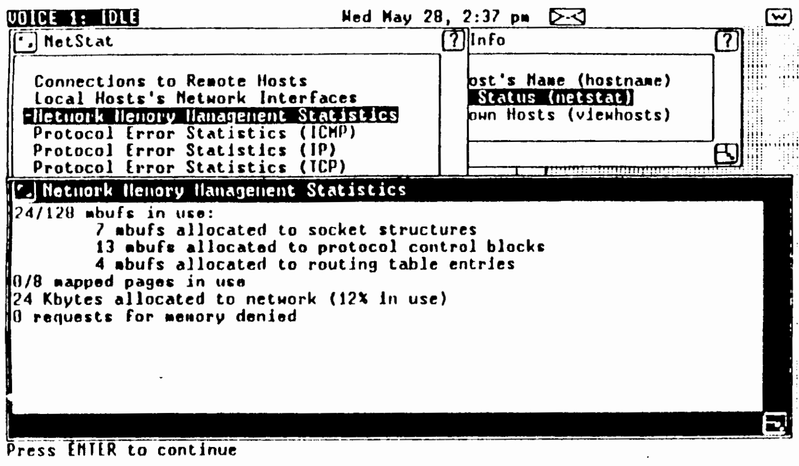
The first column lists the name of the interface. The second column indicates the size in bytes of the maximum transmission unit for that interface. The third and fourth columns show the official names of the network and the host to which the interface is attached. The fifth through eighth columns list the number of incoming and outgoing packets over the last hour and their associated errors. The last column lists the number of Ethernet collisions detected. Collisions do not imply errors, since Ethernet employs a collision detection and recovery system.

An asterisk after a value in the first column indicates that the interface has been disabled. A disabled interface can be enabled using the *ifconfig* command from the shell.

You can also perform this function from the shell by executing the *netstat -i* command.

6.2.2.3 Network Memory Management Statistics

This option lists statistics produced by memory management routines with which the network manages its own private share of kernel memory. It allocates and frees parcels of that memory in units called "mbufs". An mbuf is ordinarily 128 bytes. When messages of more than 128 bytes are formed, mbufs are chained until enough memory is supplied. When messages exceed 1024 bytes, memory may be allocated in "click mbufs," which are 1024 bytes in length.


VOICE 1: IDLE
Wed May 28, 2:37 pm
NetStat
Info
Connections to Remote Hosts
Local Hosts's Network Interfaces
-Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Network Memory Management Statistics

```
24/128 mbufs in use:
        7 mbufs allocated to socket structures
        13 mbufs allocated to protocol control blocks
        4 mbufs allocated to routing table entries
0/8 mapped pages in use
24 Kbytes allocated to network (12% in use)
0 requests for memory denied
```

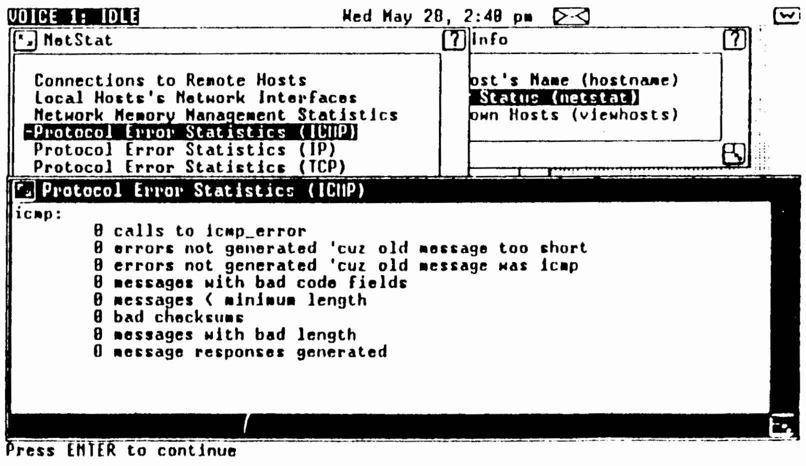
```
Press ENTER to continue
```

Figure 6-35. Network Memory Management Statistics

You can also perform this function from the shell by executing the *netstat -m* command.

6.2.2.4 Protocol Error Statistics (ICMP)

The *Protocol Error Statistics* options display error statistics for each protocol. This option displays error statistics for the ICMP protocol.


VOICE 1: IDLE
Wed May 28, 2:40 pm
NetStat
Info
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
-Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Protocol Error Statistics (ICMP)

```
icmp:
        0 calls to icmp_error
        0 errors not generated 'cuz old message too short
        0 errors not generated 'cuz old message was icmp
        0 messages with bad code fields
        0 messages < minimum length
        0 bad checksums
        0 messages with bad length
        0 message responses generated
```

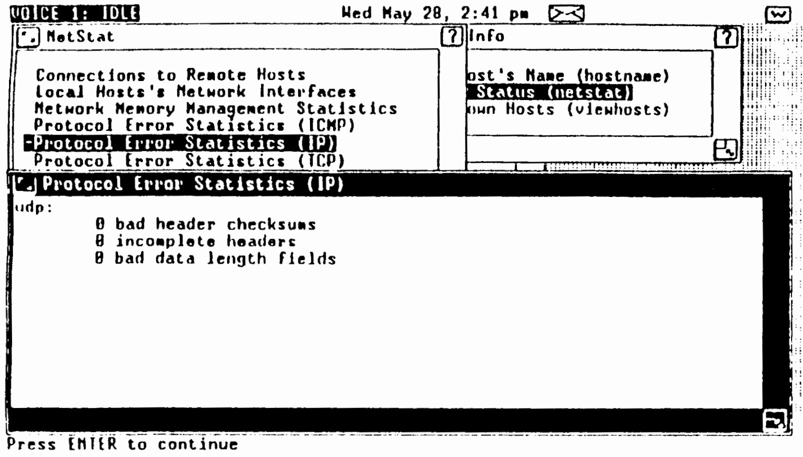
```
Press ENTER to continue
```

Figure 6-36. Protocol Error Statistics for ICMP

You can also perform this function from the shell by executing the *netstat -s -p icmp* command.

6.2.2.5 Protocol Error Statistics (IP)

The *Protocol Error Statistics* options display error statistics for each protocol. This option displays error statistics for the IP protocol.


VOICE 1: IDLE
Wed May 28, 2:41 pm
NetStat
Info
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
-Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Protocol Error Statistics (IP)

```
udp:
        0 bad header checksums
        0 incomplete headers
        0 bad data length fields
```

Press ENTER to continue

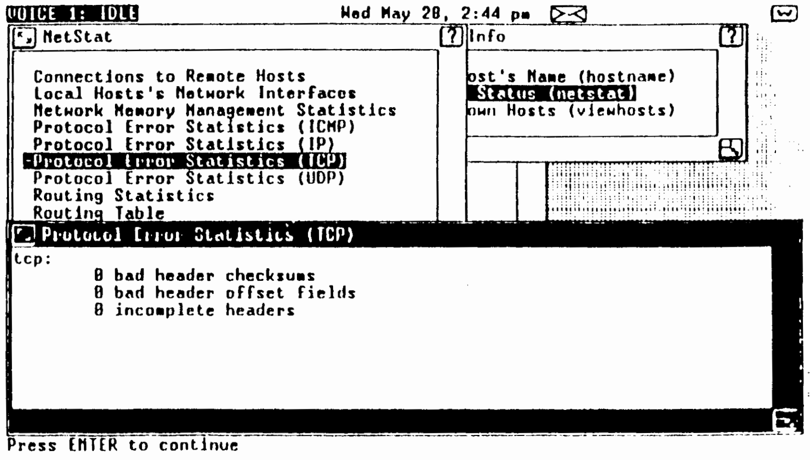
Figure 6-37. Protocol Error Statistics for IP

You can also perform this function from the shell by executing the *netstat -s -p ip* command.

6.2.2.6 Protocol Error Statistics (TCP)

The *Protocol Error Statistics* options display error statistics for each protocol. This option displays error statistics for the TCP

protocol. TCP has built in error detection and correction for reliability; any errors recorded here have been corrected, and should cause no concern.


VOICE 1: IDLE
Wed May 28, 2:44 pm
NetStat
Info
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
-Protocol Error Statistics (TCP)
Protocol Error Statistics (UDP)
Routing Statistics
Routing Table
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Protocol Error Statistics (TCP)

```
tcp:
        0 bad header checksums
        0 bad header offset fields
        0 incomplete headers
```

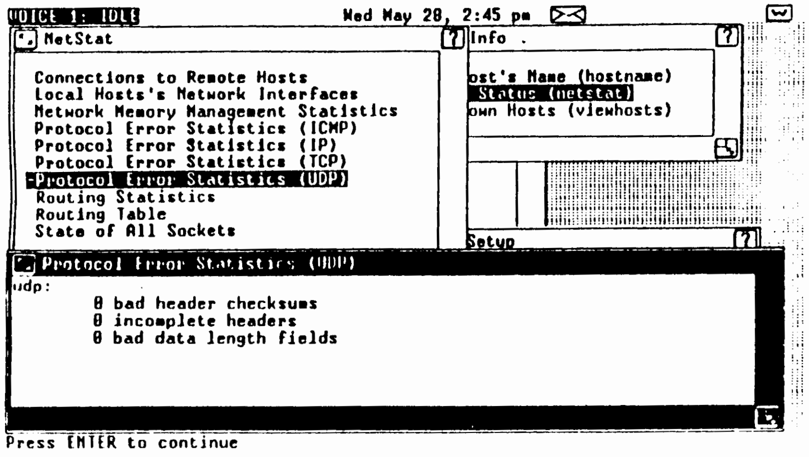
Press ENTER to continue

Figure 6-38. Protocol Error Statistics for TCP

You can also perform this function from the shell by executing the *netstat -s -p tcp* command.

6.2.2.7 Protocol Error Statistics (UDP)

The *Protocol Error Statistics* option displays error statistics for each protocol. This option displays error statistics for the UDP protocol.


VOICE 1: IDLE
Wed May 28, 2:45 pm
NetStat
Info
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Setup
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
-Protocol Error Statistics (UDP)
Routing Statistics
Routing Table
State of All Sockets
Protocol Error Statistics (UDP)

```
udp:
        0 bad header checksums
        0 incomplete headers
        0 bad data length fields
```

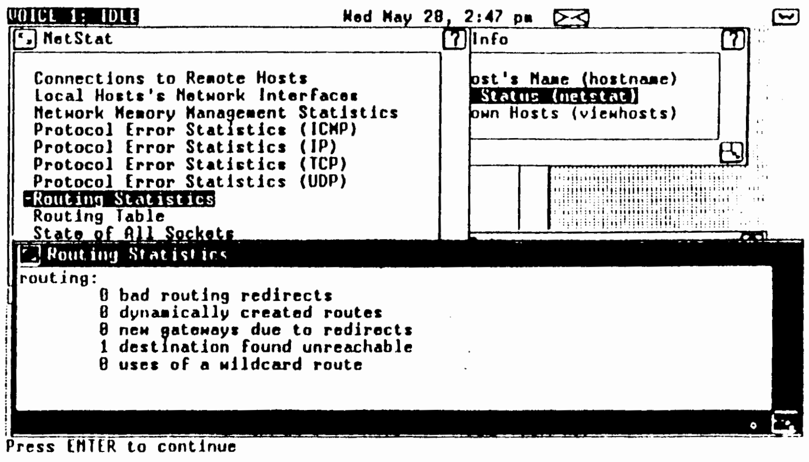
Press ENTER to continue

Figure 6-39. Displaying Protocol Error Statistics for UDP

You can also perform this function from the shell by executing the *netstat -s -p udp* command.

6.2.2.8 Routing Statistics

This option displays routing error statistics. Look here if your system performance is poor. If lots of errors are reported here, the poor performance is probably caused by the time-consuming retransmits caused by the errors.


VOICE 1: IDLE
Wed May 28, 2:47 pm
NetStat
Info
ost's Name (hostname)
Status (netstat)
own Hosts (viewhosts)
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Protocol Error Statistics (UDP)
-Routing Statistics
Routing Table
State of All Sockets
Routing Statistics

```
routing:
        0 bad routing redirects
        0 dynamically created routes
        0 new gateways due to redirects
        1 destination found unreachable
        0 uses of a wildcard route
```

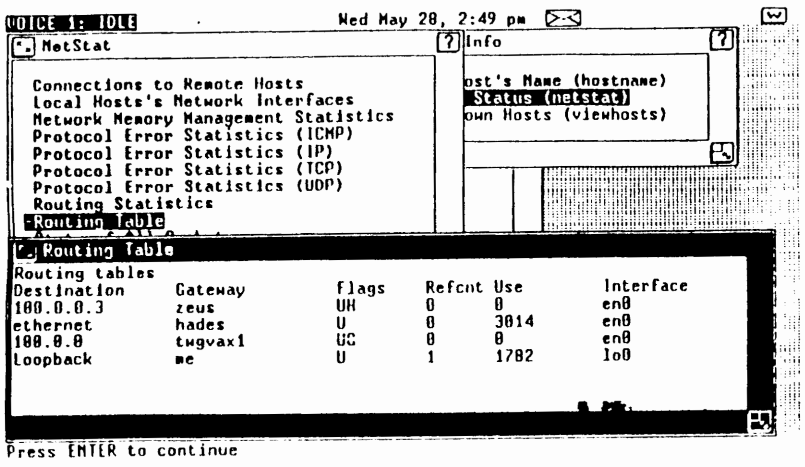
Press ENTER to continue

Figure 6-40. Displaying the Routing Statistics

You can also perform this function from the shell by executing the *netstat -rs* command.

6.2.2.9 Routing Table

This option displays the the Internet Protocol routing table. This table lists the available routes and their status. Each route consists of a destination host and a known gateway to that host.



```
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Protocol Error Statistics (UDP)
Routing Statistics
-Routing Table
```

```
Routing tables
Destination      Gateway          Flags      Refcnt Use         Interface
100.0.0.3        zeus             UH         0      0           en0
ethernet         hades            U          0      3014        en0
100.0.0          twgvax1          UG         0      0           en0
Loopback         me               U          1      1782        lo0

Press ENTER to continue
```

Figure 6-41. Displaying the Routing Table

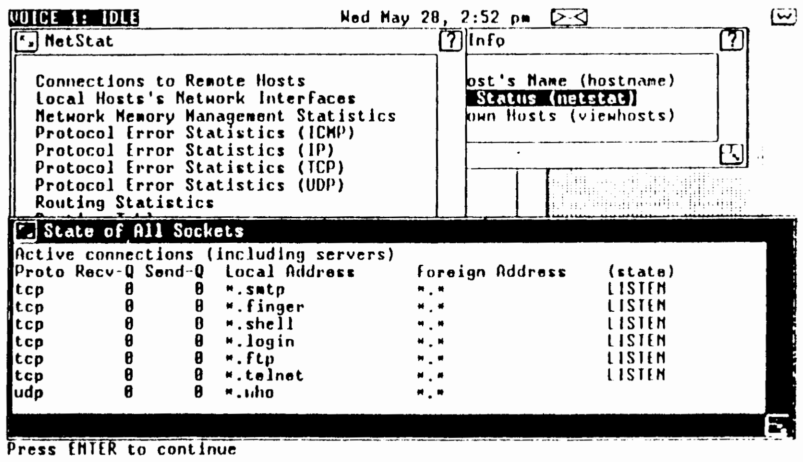
The first column shows the name of a desired destination for a packet. The second column lists the appropriate gateway. The third column indicates whether the gateway is up or down. An "H" in this column indicates that the destination is a host; a "G" indicates that the destination is a gateway. The fourth column shows the number of times the route has been used in the last hour. The fifth column shows the number of packets that have been routed through this gateway within the last hour. The last column indicates the interface used for the network.

You can perform this function from the shell by executing the *netstat -r* command.

This command provides a quick way of determining which networks are up and which interfaces these networks use. This table is normally updated manually with the *route* command. It can also be maintained by the *routed* daemon.

6.2.2.10 State of All Sockets

This option shows the state of all sockets; those used by servers and those occupied by the network daemons.



```
Connections to Remote Hosts
Local Hosts's Network Interfaces
Network Memory Management Statistics
Protocol Error Statistics (ICMP)
Protocol Error Statistics (IP)
Protocol Error Statistics (TCP)
Protocol Error Statistics (UDP)
Routing Statistics
```

```
Active connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address      (state)
tcp        0      0  *.smtp                 *.*                  LISTEN
tcp        0      0  *.finger               *.*                  LISTEN
tcp        0      0  *.shell                *.*                  LISTEN
tcp        0      0  *.login                *.*                  LISTEN
tcp        0      0  *.ftp                  *.*                  LISTEN
tcp        0      0  *.telnet               *.*                  LISTEN
udp        0      0  *.who                  *.*

Press ENTER to continue
```

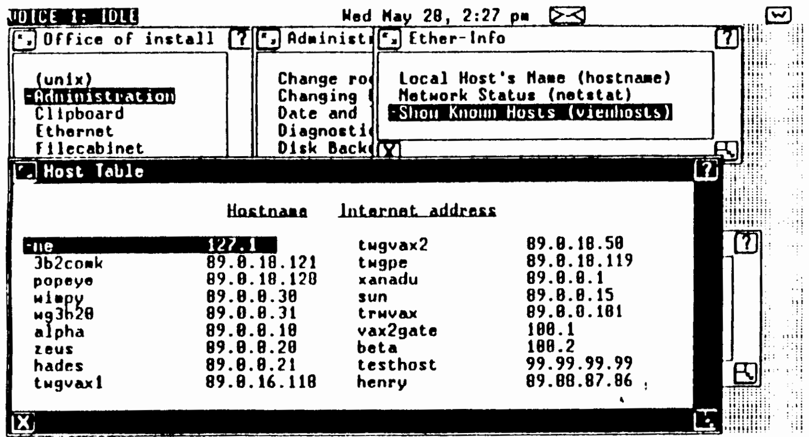
Figure 6-42. Displaying the State of All Sockets

The first column indicates the protocol associated with a given socket. The second and third columns indicate the current length of the socket queues. The fourth and fifth columns show the local and remote addresses of the connection with which the socket is associated. The sixth column shows the state of the connection. Note that all the daemons are in the "listen" state and that their foreign addresses are specified by the shell wildcard character "*".

6.2.3 Show Known Hosts (viewhosts)

This option displays the contents of the */etc/hosts* file.

To display a list of the hosts on your network, select *Show Known Hosts (viewhosts)*. The Host Table window opens showing the host names and Internet addresses of the current known hosts. If there are many known hosts, the table may appear in two columns.



```
Wed May 28, 2:27 pm

                    Hostname    Internet address

-ne            127.1            twgvax2        89.0.10.50
 3b2comk       89.0.10.121      twgpe          89.0.10.119
 popeye        89.0.10.120      xanadu         89.0.0.1
 wimpy         89.0.0.30        sun            89.0.0.15
 wg3b20        89.0.0.31        trwvax         89.0.0.101
 alpha         89.0.0.10        vax2gate       100.1
 zeus          89.0.0.20        beta           100.2
 hades         89.0.0.21        testhost       99.99.99.99
 twgvax1       89.0.16.110      henry          89.00.87.06

Press ENTER to continue
```

Figure 6-43. Displaying the Known Hosts

6.3 HOST AND NETWORK TESTS

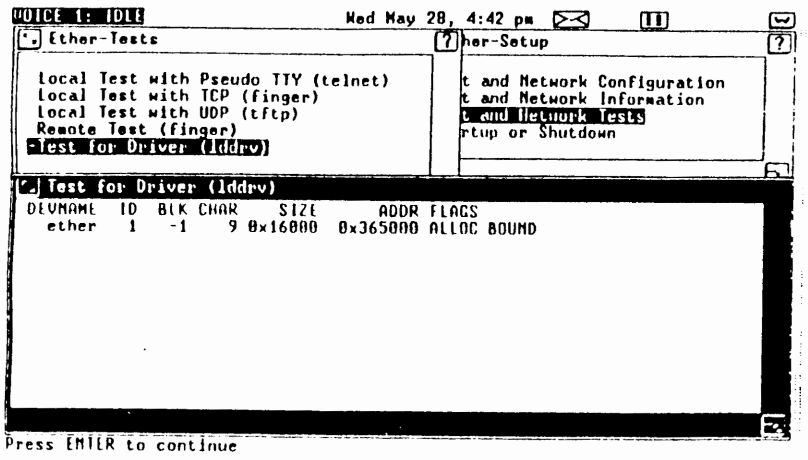
The sequence of tests contained in this window are intended for use after installing the network or when some sort of network failure seems to have occurred. Each test exercises a different feature of the network. The sequence is divided into three groups. The first, consisting of one test, determines whether the network driver is loaded or not. The second, consisting of three tests, exercises different protocols in the loaded network driver, as well as the pseudo terminal driver. The third, with one test, accesses both the Ethernet software and the hardware.

While the tests are alphabetized in the window, they should be performed in the order presented below. By performing the tests in the following sequence, you will be able to verify the correct or incorrect functioning of these different network features.

6.3.1 Test for Driver (lddrv)

This test simply checks the status of the network driver. Results resembling those shown below indicate that the driver is loaded. If the flags ALLOC and BOUND are not shown, the driver is not ready for operation. Ordinarily this will never happen unless the driver has been intentionally unloaded. To reload the driver, select *Software Setup* in the Administration window. Load the driver with the *Configure Loadable Device Drivers* option and rerun this test. If it passes then select [Start-up Ethernet on this Host] under the [Start-up or Shutdown] entry in the [Ethernet Setup] menu.

]



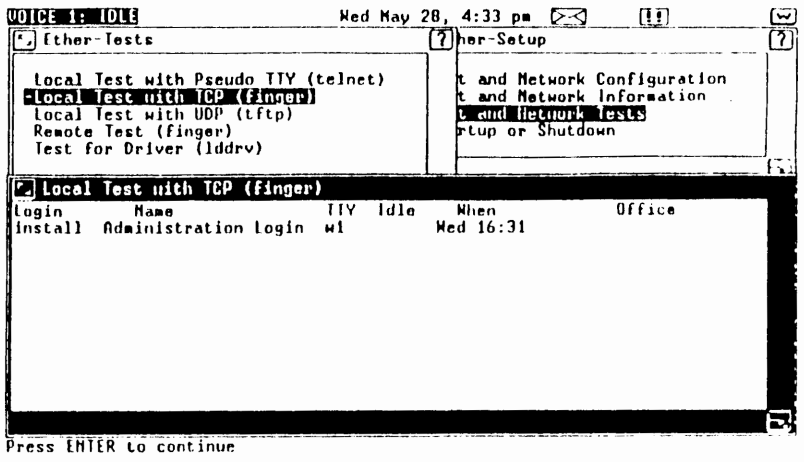
```
DEVNAME  ID  BLK CHAR    SIZE      ADDR FLAGS
  ether   1   -1    9 0x16000  0x365000 ALLOC BOUND
```

Press ENTER to continue

Figure 6-44. Testing the Network Driver

6.3.2 Local Test with TCP (finger)

This and the other two "local" tests exercise the network driver without accessing the Ethernet hardware. This test particularly exercises TCP using *finger* to do a loopback test. Results resembling those below indicate that the TCP protocol is functioning properly.


VOICE 1: IDLE
Wed May 28, 4:33 pm
Ether-Tests
Local Test with Pseudo TTY (telnet)
Local Test with TCP (finger)
Local Test with UDP (tftp)
Remote Test (finger)
Test for Driver (lddrv)
her-Setup
t and Network Configuration
t and Network Information
t and Network Tests
rtup or Shutdown
Local Test with TCP (finger)

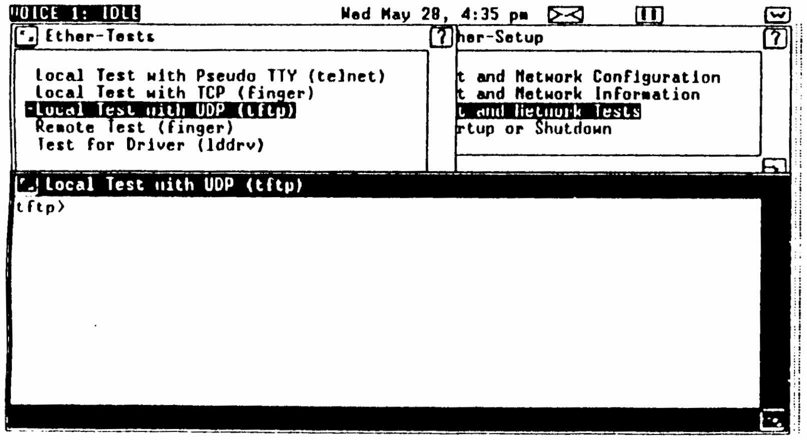
```
Login      Name                 TTY  Idle   When         Office
install  Administration Login   w1          Wed 16:31
```

Press ENTER to continue

Figure 6-45. Testing with TCP

6.3.3 Local Test with UDP (tftp)

This local test exercises the UDP portion of the network driver using *tftp* for a loopback connection. Receiving the tftp> prompt indicates that you are running the tftp command interpreter. To verify that UDP is functioning properly, execute one of the tftp command "get" or "put". The tftp daemon must be, running for this test to complete successfully. When the transfer has been completed, exit tftp by entering "quit", then close the window by pressing ENTER.

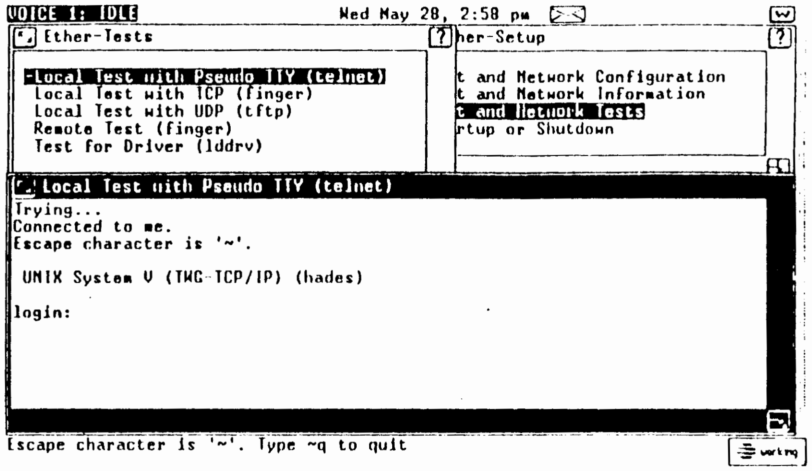


```
VOICE 1: IDLE                 Wed May 28, 4:35 pm
Ether-Tests                              her-Setup

  Local Test with Pseudo TTY (telnet)    t and Network Configuration
  Local Test with TCP (finger)           t and Network Information
 -Local Test with UDP (tftp)             t and Network Tests
  Remote Test (finger)                   rtup or Shutdown
  Test for Driver (lddrv)

Local Test with UDP (tftp)
tftp>
```

Figure 6-46. Testing with UDP

6.3.4 Local Test with Pseudo TTY (telnet)

In addition to TCP, this test also exercises the pseudo terminal portion of the network driver using *telnet* to perform a loopback test. Appearance of the login prompt indicates that this portion of the driver is functioning properly. Type quit to exit the test.

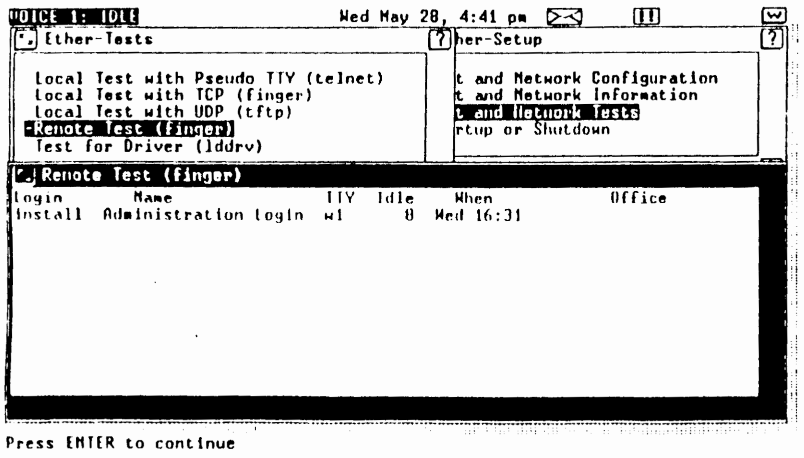


```
VOICE 1: IDLE                 Wed May 28, 2:58 pm
Ether-Tests                              her-Setup

 -Local Test with Pseudo TTY (telnet)    t and Network Configuration
  Local Test with TCP (finger)           t and Network Information
  Local Test with UDP (tftp)             t and Network Tests
  Remote Test (finger)                   rtup or Shutdown
  Test for Driver (lddrv)

Local Test with Pseudo TTY (telnet)
Trying...
Connected to me.
Escape character is '~'.

 UNIX System V (TWG-TCP/IP) (hades)

login:

Escape character is '~'. Type ~q to quit
```

Figure 6-47. Testing with Pseudo TTY

6.3.5 Remote Test (finger)

With this test you utilize the Ethernet software and hardware using *finger* to display the users on a remote host, Exit the test window by pressing <Enter>. If all these tests have been passed, you should have no difficulty using your network.



```
VOICE 1: IDLE                    Wed May 28, 4:41 pm
Ether-Tests                              her-Setup

  Local Test with Pseudo TTY (telnet)    t and Network Configuration
  Local Test with TCP (finger)           t and Network Information
  Local Test with UDP (tftp)             t and Network Tests
 -Remote Test (finger)                   rtup or Shutdown
  Test for Driver (lddrv)

Remote Test (finger)
login     Name                  TTY  Idle   When          Office
install   Administration login  w1      0  Wed 16:31

Press ENTER to continue
```

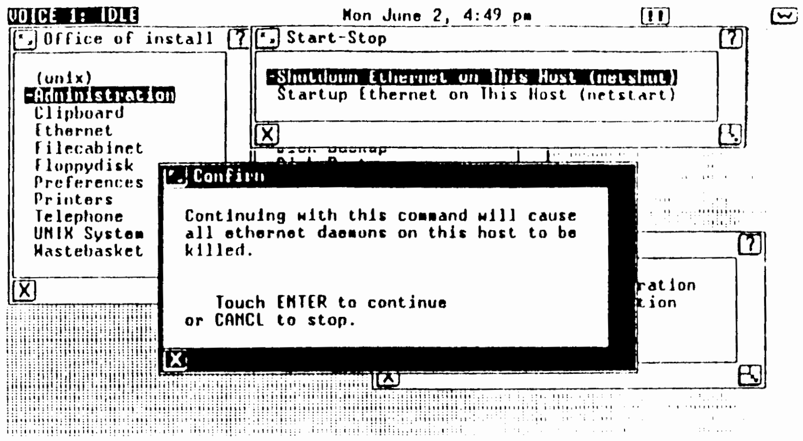
Figure 6-48. Remote Test of the Ethernet Software and Hardware

6.4 STARTUP OR SHUTDOWN

This option starts or stops all daemons and starts up or shuts down Ethernet on the local host.

6.4.1 Shutdown Ethernet

To shut down Ethernet, select *Startup or Shutdown* to open the Start-Stop window. Select *Shutdown Ethernet on This Host (netshut)*. The Confirm window opens with the following message.



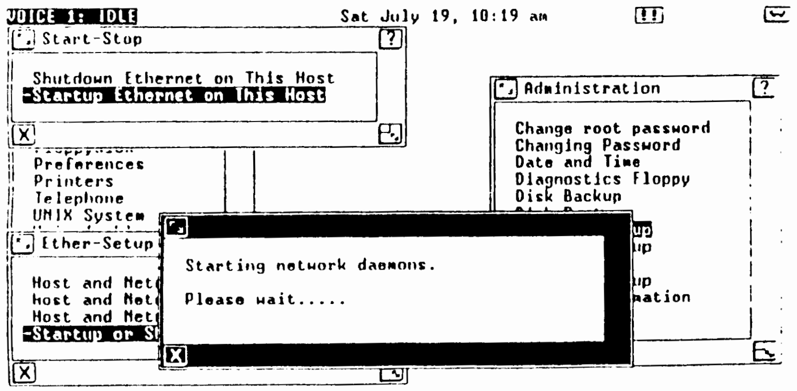
```
VOICE 1: IDLE                Mon June 2, 4:49 pm
Office of install     Start-Stop

  (unix)               -Shutdown Ethernet on This Host (netshut)
 -Administration        Startup Ethernet on This Host (netstart)
  Clipboard
  Ethernet
  Filecabinet
  Floppydisk      Confirm
  Preferences
  Printers         Continuing with this command will cause
  Telephone        all ethernet daemons on this host to be
  UNIX System      killed.
  Wastebasket
                      Touch ENTER to continue
                   or CANCL to stop.
```

Figure 6-49. Shutting Down Ethernet

Press ENTER to continue and shutdown the Ethernet daemons. The Confirm window will open again asking if you wish to unload the Ethernet driver. Press ENTER if you wish to unload the Ethernet driver, otherwise press CANCL.

6.4.2 Startup Ethernet

To start up Ethernet, select *Startup or Shutdown* to open the Start-Stop window. Select *Startup Ethernet on This Host (netstart)*. The Confirm window opens with the following message:



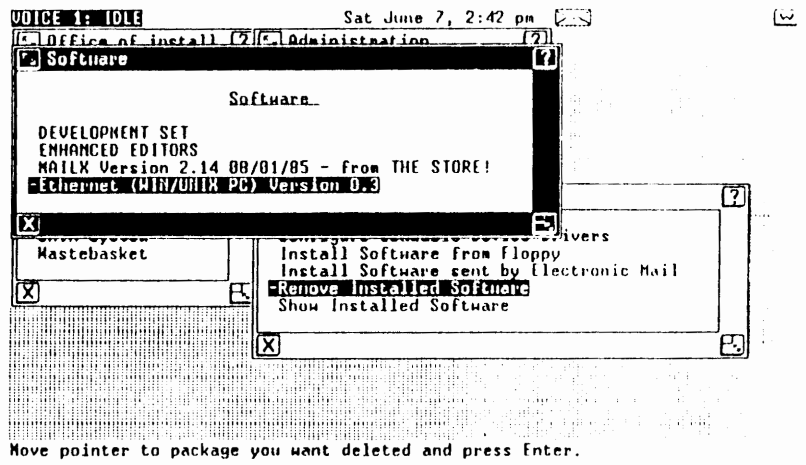
```
VOICE 1: IDLE                 Sat July 19, 10:19 am
Start-Stop
  Shutdown Ethernet on This Host
 -Startup Ethernet on This Host
  Preferences
  Printers
  Telephone
  UNIX System
Ether-Setup
  Host and Net
  host and Net
  Host and Net
 -Startup or S
Administration
  Change root password
  Changing Password
  Date and Time
  Diagnostics Floppy
  Disk Backup

  Starting network daemons.

  Please wait.....
```

Figure 6-50. Starting Up Ethernet

7. SOFTWARE DEINSTALLATION

In some cases, such as when new releases of WIN/3B LAN are issued, it will be desirable to entirely *deinstall* WIN/3B LAN. Deinstalling stops all network-related running processes and removes all network software from the disk. The *Remove Installed Software* option removes all files associated with WIN/3B LAN. All other files are left undisturbed.

To deinstall the WIN/3B LAN files, open the Administration window and select *Software Setup*. When the Software window opens, select *Remove Installed Software*. The Software window opens showing the currently installed software. Select *Ethernet* (WIN/3B LAN) Press ENTER to deinstall the Ethernet software.



```
VOICE 1: IDLE                 Sat June 7, 2:42 pm
Office of install   Administration
Software
                    Software

  DEVELOPMENT SET
  ENHANCED EDITORS
  MAILX Version 2.14 08/01/85 - from THE STORE!
 -Ethernet (WIN/UNIX PC) Version 0.3

  Wastebasket
                    ivers
  Install Software from Floppy
  Install Software sent by Electronic Mail
 -Remove Installed Software
  Show Installed Software
Move pointer to package you want deleted and press Enter.
```

Figure 7-1. Deinstalling Ethernet

The deinstall procedure first stops all daemons associated with WIN/3B LAN and then removes all files installed with WIN/3B LAN.

When the deinstallation is complete, a window appears with a message declaring that the Ethernet software has been removed. Press ENTER to return to the Software window.

8. NETWORK ERRORS

This chapter contains lists of most of the error messages associated with WIN/3B LAN. Next to them are possible causes and solutions. It is organized into five sections based on the probable message source. Within each section the errors are organized alphabetically according to the text of the error.

8.1 SYSTEM ERRORS

These occur when an improper operation occurs at the kernel level. They are produced by the lower level protocols. Many of these will be seen only by those in the process of developing new network applications with the SCI or TLI libraries.

Address already in use
: Addresses must be unique to each host and are permitted to be used only once.

Address family not supported by protocol family
: An address incompatible with the requested protocol was used.

Bad protocol option
: A bad option was specified in a *getsockopt* or *setsockopt* call.

Can't assign requested address
: This normally results from an attempt to *bind* a socket to an illegal address or to one already in use.

Can't send after socket shutdown
: A request to send data was disallowed because the socket had already been shut down with a previous *shutdown* call.

Connection reset by peer
A connection was forcibly closed by a peer. This normally results from the peer executing a *shutdown* call.

Connection timed out
A *connect* request failed because the connected party did not properly respond after a specified period of time. This could indicate that the remote host is down.

Connection refused
No connection could be made because the remote host actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host.

control block: symbol not in namelist
This error indicates a problem with the kernel symbol table. Contact AT&T Technical Support.

Destination address required
A required address was omitted from an operation on a socket.

Message too long
A message sent on a socket was larger than the internal message buffer.

Network dropped connection on reset
The host you were connected to crashed and rebooted.

Network is down
A socket operation encountered a dead network.

Network is unreachable
A socket operation was attempted on an unreachable network.

No buffer space available
An operation on a socket or pipe was not performed because the system lacked sufficient buffer space.

Operation already in progress
An operation was attempted on a non-blocking object that already had an operation in progress.

Operation now in progress
An operation was attempted that takes a long time to complete (such as a *connect*).

Operation would block
A blocking operation was attempted upon an object in the non-blocking mode.

Socket is already connected
A *connect* request was made on an already connected socket or a *sendto* or *sendmsg* request made on a connected socket specified a destination other than the connected party.

Socket is not connected
A request to send data was disallowed because the socket had already been shut down with a previous *shutdown* call.

Socket operation on non-socket
A socket operation was attempted on a non-socket device.

Socket type not supported
Support for the specified socket type has not been configured into the system or no implementation for it exists.

Software caused connection abort
A connection was aborted by the local host.

8.2 FTP AND TFTP ERRORS

Already connected to *hostname*, use disconnect first
You are currently connected to remote host *hostname*, even though you may not be logged on. Either log in or disconnect and try again.

bad port number
A zero or negative port number was specified by the user with the *port* option on the *open* command.

can't find list of remote files, oops
When invoking a multiple command, a temporary file is created in the */tmp* directory. The system cannot find this file, or it was accidentally removed. Try the procedure again.

***filename* not a plain file**
A directory or some special device was specified for a transfer operation when only a file was appropriate.

ftp/tcp: unknown service
The */etc/services* file does not contain an entry for the specified *ftp* service. An entry for this service should be created in this file. See *services(5)* in the **WIN/3B LAN Programmer's Reference Manual**.

Login failed
The user gave an invalid username or password. Try again.

Lost connection
The server on the remote host closed the connection. This could be caused by many different things, including the remote system crashing, automatic logout, etc.

No port available for data connection
This is an unlikely error. It could indicate software problems. Contact AT&T Technical Support.

No target machine specified
This error is associated with *tftp* when a file transfer is invoked without specifying a remote machine. Do an *open* before invoking the transfer command.

Not connected
You attempted to use an *ftp* command requiring a connection before you established the connection. Do a *connect* before invoking the transfer command.

Unknown host
The hostname given is not in the */etc/hosts* file. Verify the hostname and try again.

User already logged in
A user who is already logged into a remote host attempted to login again.

We only support non-print format, sorry
Currently *ftp* only supports the 'non-print' format.

We only support file structure, sorry
Currently *ftp* only supports the 'file' structure.

unknown mode
Attempted to set a mode that is not supported. Currently, the supported modes are binary, ASCII and tenex.

?Ambiguous command
Invoked an abbreviated *ftp* command that was not unique. Be sure to specify enough of the command to make it unique.

?Ambiguous help command ***command***

The *command* specified for the help command was not unique. Be sure to specify enough to make it unique.

?Invalid command

Invoked an unknown command. For a complete list of valid commands, type ***help***.

?Invalid help command ***command***

Invoked the help command with an invalid *command* name. For a complete list of valid commands, type ***help***.

8.3 TELNET ERRORS

bad port number

A zero or negative port number specified by the user with the port option on the ***open*** command.

Connection closed by foreign host

A connection was forcibly closed by a peer. This normally results from the peer executing a ***shutdown*** call, although it may also indicate that the remote host has crashed.

no open connection to negotiate options

You must be connected to a remote host before you can negotiate options.

telnet: tcp/telnet: unknown service

The ***/etc/services*** file does not contain an entry for the ***telnet*** service. An entry for this service should be created in this file. See ***services(5)*** in the **WIN/3B LAN Programmer's Reference Manual.**

unknown host

The hostname given is not in the ***/etc/hosts*** table or a hostname was specified incorrectly. See ***win3baddhosts*** in the WIN/3B LAN Ethernet Administration menu.

?Already connected to ***hostname***

You attempted to establish a connection with a remote host to which you are already connected.

?Ambiguous command

You invoked an abbreviated ***telnet*** command that was not unique. Be sure to specify enough of the command to make it unique.

?Ambiguous help command *command*

The *command* specified for the help command was not unique. Be sure to specify enough to make it unique.

?Invalid command

Invoked an unknown command. For a complete list of valid commands, type ***help***.

?Invalid help command ***command***

Invoked the help command with an invalid *command* name. For a complete list of valid commands, type ***help***.

8.4 REMOTE COMMAND ERRORS

Bad .rhosts ownership

~/.rhosts must be owned by the current user. Its ownership is determined to be the owner of the home directory in which the file resides.

Host name for your address unknown

The name of your host is not in */etc/hosts* on the remote host.

Login incorrect

An invalid username or password was specified. Try again.

Login timed out after *x* seconds

The user did not login within the required time, so the login procedure exited.

.netrc file not correct mode. Remove password or correct mode"

The *.netrc* file should only be readable by the owner. No other protection mode is allowed due to the sensitivity of the information this file contains.

No directory!

There is no home directory specified in the */etc/passwd* file for the user logging into a remote host.

No shell

There is no shell specified in */etc/passwd* file for a user logging into a remote host.

rcp: who are you?

rcp attempted to ascertain the username of the user but failed.

rcp: invalid user name

There is no such user on the remote host.

rcp: lost connection

During a remote copy, the remote host closed the connection. Possibly the remote host crashed.

too many hosts

The ***ruptime*** command can print information on a limited number of hosts, (about 100). If the number of hosts on a network exceeds this amount, ***ruptime*** produces this error.

no hosts!?!

This error occurs when the *ruptime* command is invoked and there is no information on any hosts in the */usr/spool/rwho* directory. Most likely *rwhod* is not running.

rwho too many users

The *rwho* command can print information on a maximum of about 1000 users. If the number of users on a network exceeds this amount, the *rwho* produces this error.

.rhosts is a soft link
: *~/.rhosts* file cannot be a link.

Unknown .netrc option
: *~/.netrc* contains illegal information. See *.netrc(5)* of the **WIN/3B LAN Programmer's Reference Manual.**

unknown service
: The */etc/services* file does not contain an entry for this command. An entry for this service should be created in this file. See *services(5)* in the **WIN/3B LAN Programmer's Reference Manual.**

You have too many processes running
: There are too many processes running on the remote host for one user.

8.5 ADMINISTRATIVE COMMAND ERRORS

arpbypass must have SYSPRV
: You must be a superuser in order to run the *arpbypass* command.

no interface for internet address
: No interface has been defined yet for the specified Internet address. See *ifconfig(1M)* in the **WIN/3B LAN Programmer's Reference Manual.**

No room in ARP table, try later
: Attempting an *arpbypass* command, the ARP table currently is full. Delete unwanted ARP entries with the *arpbypass(1M)* command.

Warning does not match utsnames
: This message is produced when you attempt to set the hostname to something other than the current UNIX system name.

APPENDIX A: HARDWARE INSTALLATION

[Installation of Ethernet software and expansion board is also described in the AT&T UNIX PC Ethernet Board Installation and Diagnostic Guide.]

APPENDIX B: SAMPLE SCENARIOS

This appendix gives three examples of problems that could arise with WIN/3B LAN. The purpose of this section is to demonstrate the network debugging process.

The examples in this section are performed from the shell You can perform the same examples using the menu interface

SCENARIO 1

When attempting to *ftp* a file from a remote host to the local UNIX PC, the system responds with the error message *connection refused*.

```
#ftp 3b2

ftp: connect: connection refused
ftp>
```

This indicates that the *ftpd* daemon on the remote host had trouble establishing the requested connection. If other WIN/3B LAN commands do not work, there may be a problem with the kernel level networking software on the remote host.

Log on as root to the remote host. The first thing to check is whether the remote *ftp* daemon was running by using *netstat -a:*.

```
#netstat -a

Active connections (including servers)

Proto  Recv-Q  Send-Q  Local Address  Foreign Address  (state)
tcp    0       0       *.shell        *.*              LISTEN
tcp    0       0       *.login        *.*              LISTEN
tcp    0       0       *.finger       *.*              LISTEN
tcp    0       0       *.telnet       *.*              LISTEN
```

Log off the remote host and log back onto the local host. Since there is no listing for *ftpd*, it is not running. To reactivate this daemon, simply type the command */usr/ethernet/daemons/ftpd*.

```
#/usr/ethernet/daemon/ftpd
```

Try the *ftp* command again:

```
#ftp 3b2

 Connection Opened
 Using 8-bit bytes.
< 3b2 comk FTP server
(Version 4.1 Wed Jun 26 01:38:17 PDT 1985)
 TWGVAX FTP User Process (Version 3.00)
*
```

SCENARIO 2

When attempting to make a loopback connection with *telnet*, the system responds with this error message:

```
#telnet 3b2
Trying...
telnet: connect:  Network is unreachable
telnet> quit
#
```

Again, find out whether the *telnet daemon* is down using *netstat -a*.

```
#netstat -a

Active connections (including servers)

Proto  Recv-Q  Send-Q  Local Address  Foreign Address  (state)
tcp    0       0       3b2.telnet     sperrypc.9235    ESTABLISHED
tcp    0       0       *.shell        *.*              LISTEN
tcp    0       0       *.login        *.*              LISTEN
tcp    0       0       *.finger       *.*              LISTEN
tcp    0       0       *.telnet       *.*              LISTEN
```

All the servers, including *telnet* are up and a *telnet* between the local host and a remote host is currently active. Possibly there is a problem with the loopback interface. Using the *netstat -i* command, check the interface.

```
        #netstat -i

Name   Mtu   Network   Address  Ipkts  Ierrs  Opkts  Oerrs  Collis
en0    1500  Peg       twgvax   3349   0      386    0      0
lo0*   1536  Loopback  3b2      0      0      0      0      0
```

The asterisk indicates that the loopback interface is down. To bring this interface back up use the *ifconfig* command.

```
#ifconfig lo0 up

#netstat -i

Name   Mtu    Network    Address   Ipkts   Ierrs   Opkts   Oerrs   Collis
en0    1500   Peg        twgvax    3349    0       386     0       0
lo0    1536   Loopback   3b2       0       0       0       0       0
```

The loopback interface is now up. The loopback mode should now function properly.

SCENARIO 3

None of the networking operations recognize the remote hosts.

```
#telnet 3b5
3b5: unknown host

#ftp 3b20
3b20: unknown host
```

First make certain that the interface for the network is up. Again use the *netstat -i* command.

```
#netstat -i

Name   Mtu    Network    Address   Ipkts   Ierrs   Opkts   Oerrs   Collis
en0    1500   Peg        twgvax    3349    0       386     0       0
lo0    1536   Loopback   3b2       0       0       0       0       0
```

All of the interfaces are up, possibly there is a problem with the */etc/hosts* file.

```
#more /etc/hosts

/etc/hosts: no such file or directory.
```

The */etc/hosts* file is missing! Without this file, none of the network commands will work.

For some reason, this file has been deleted or removed. Look for it first in the standard UNIX directory */lost&found*. If you do not find it, you will have to either recreate it or restore it from a backup copy. The format for this file is listed in Section 5 of the **WIN/3B LAN Programmer's Reference Manual.**

APPENDIX C: RECOMMENDED READING

For information about networking in general, the following selections are recommended:

E.B Brooner, ***The Local Area Network Book***, Howard Sams & Co., Inc, Indianapolis, IN, 1984. This book serves as an excellent introduction to local area networking. It is written for readers with minimal technical background.

J.H. Green, ***Local Area Networks***, Scott, Foresman and Company, Glenview, IL, 1985. Green's book discusses networking from the perspective of the business professional.

A.S. Tannenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1981. This is possibly the best text book for computer networks. Tannenbaum discusses all aspects of computer networking in a straight-forward, readable style.

A.S. Tannenbaum, "Network Protocols," *ACM Computing Surveys*, Volume 13, Number 4, December 1981. This excellent article addresses in a broad yet detailed fashion the use of protocols in networking and different network architectures.

INDEX

INDEX