UNISYS

CTIX™

Network Administrator's Guide

Second Edition (January 1990) 09-02320-01

Convergent Technologies makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Convergent Technologies reserves the right to revise this publication and to make changes from time to time in its content without being obligated to notify any person of such revision or changes.

Convergent Technologies and NGEN are registered trademarks of
Convergent Technologies, Inc.

Art Designer, Chart Designer, ClusterCard, ClusterNet, ClusterShare, Convergent, CT-DBMS, CT-MAIL, CT-Net, CTIX, CTOS, DISTRIX, Document Designer, The Operator, AWS, CWS, IWS, S/50, S/120, S/160, S/220, S/320, S/640, S/1280, Multibus, TeleCluster, Voice/Data Services, Voice Processor, and X-Bus are trademarks of Convergent Technologies, Inc.

UNIX and RFS are trademarks of AT&T.

This software and documentation are based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

This manual was prepared on a Convergent Technologies S/320 Computer System and printed on an Apple LaserWriter II printer.

Contents for Part 1

Section 1. Networking Terms and Concepts

Section 2. Super-user Status and CTIX Modes

Section 3. CTIX Boot Options

Contents for Part 1

Section 4. Using the Administration Tools

Section 5. Network Monitoring and Troubleshooting

Glossary

Index

Figures for Part 1

Tables for Part 1

About this Guide

This guide contains all the information needed by a network administrator for the CTIX operating system. If you are not already familiar with the principles of UNIX network administration, spend some time reading and studying this guide. Then get some ''hands-on'' experience with the network before you begin to support a group of users.

Note: *As there are always some differences between two separate releases of the CTIX Operating System, it is important to read the Release Note for the particular version you are running.*

Responsibilities of Network Administrator

Whether you spend all or only a small part of your time performing the role of network administrator, these are your specific responsibilities:

- Setting up network hardware
- Adding computers to your network
- Removing computers from your network
- Adding and deleting network services
- Ensuring network security
- Installing new releases of networking software
- Maintaining the networking software and hardware
- Troubleshooting problems when they arise
- Monitoring the network and optimizing network performance.

Organization of This Guide

The *CTIX Network Administrator's Guide* is really four separate manuals, each Part of the guide corresponding to a separate manual. Each Part contains its own Table of Contents, Glossary, and Index. The four Parts are organized as follows:

Part 1: Introduction

This Part provides basic information on CTIX networking.

- Section 1 explains basic networking terms and concepts.
- Section 2 explains what *super-user* status means, how to become the super-user, and how to take CTIX to single-user mode and back to multiuser mode.
- Section 3 describes the options available to the CTIX administrator for system boot.
- Section 4 explains the CTIX administration tools, a menu driven interface for system and network administration.
- Section 5 discusses network monitoring and troubleshooting.
- The final section is a networking glossary.

Part 2: Ethernet Administration

This Part provides detailed information on networking using Ethernet networking system.

- Section 1 explains Ethernet terms and concepts.
- Section 2 discusses Ethernet hardware setup.
- Section 3 discusses Ethernet software setup.

- Section 4 discusses Ethernet monitoring and troubleshooting.
- The final section is an Ethernet glossary.

Part 3: Serial Line Internet Protocol Administration

This Part provides information on networking using SLIP, or basic serial line, communications.

- Section 1 gives an overview of SLIP.
- Section 2 discusses static SLIP.
- Section 3 discusses switched SLIP.
- Section 4 discusses SLIP monitoring and troubleshooting.
- The final section is a SLIP glossary.

Part 4: Internet Administration

This Part provides information on Internet networking, networking between different networks.

- Section 1 discusses operating system configuration for networking, including some tuning tips.
- Section 2 discusses adding and deleting hosts and networks.
- Section 3 discusses adding and deleting network services.
- Section 4 discusses internet UUCP setup.
- Section 5 discusses routing setup.
- Section 6 discusses internet security.

- Section 7 discusses adding and deleting equivalent machines and users.
- Section 8 discusses internet monitoring and troubleshooting.
- Section 9 contains sample network setups.
- The final section is an internet glossary.

How To Use This Guide

The *CTIX Network Administrator's Guide* is intended primarily for experienced UNIX network administrators, and as such assumes that you already know the following:

- Basic information about the logical structure of the UNIX file system
- How to use the most common tools for manipulating files: **cat**, **grep**, **mv**, **cp**, and so forth
- How to use a UNIX editor
- How to use the *CTIX Operating System Manual* as a reference.

If You Are New to UNIX

If you are a beginning or new UNIX administrator, you will need some preparation before you are ready to set up and maintain your CTIX network. You will need to learn the basic UNIX concepts and skills listed in the previous section and enough CTIX system administration to understand how your systems and the network interact. If you follow the process outlined below, you will soon have a solid enough grasp of UNIX to get started and set up your network.

1. Have on hand one of the introductory books on the UNIX operating system listed at the end of this section. Read the introductory material in the *CTIX Operating System Manual*, and learn how the information is organized.

2. Study the *S/Series CTIX Administrator's Guide* and/or the *CTIX Administration Tools Manual.* Read and practice until you understand the responsibilities of the CTIX system administrator. Pay attention to the tools and strategies used by the system administrator.
3. Read the installation manual(s) for your system and its expansion cards and upgrades. If your systems are not installed, do that first or get the proper person to do it for you. Your systems should be running properly before you try to install the network. If the network is already installed, become familiar with the physical setup of the network. If it is your responsibility to install it, perform the installation.
4. Read Section 3 of this Introduction to learn how to start and stop your computers and how to become the super-user.
5. Read Section 4 of this Introduction to learn about CTIX boot options.
6. Read Section 5 of this Introduction to learn about the CTIX administration tools.
7. Load the networking software according to the instructions in the release notices. You will need the CTIX operating system tape and the TCP/IP tape.
8. If you are using Ethernet in your network, refer to the Ethernet Section, (Part 2) of this guide. Read Section 1, ''Ethernet Terms and Concepts,'' and Section 2, ''Hardware Setup.'' Set up the network hardware if that is your responsibility. Read Section 3, ''Software Setup,'' and then check your **/etc/rcopts** directory to make sure that the correct boot options are there for Ethernet.
9. If you are using SLIP in your network, refer to the SLIP Section (Part 3) of this guide. Read Section 1, ''Overview,'' and then continue with either Section 2, ''Direct SLIP,'' or Section 3, ''Switched SLIP.'' Direct SLIP can be used only on direct serial lines without modems. Set up any needed serial lines and modems if that is your responsibility. Check your **/etc/rcopts** directory to make sure that the correct boot options are there for SLIP.

10. Refer to the Internet section (Part 4) of this guide. Read Section 1, ''Network Setup,'' and then check your **/etc/rcopts** directory to make sure that the correct boot options are there for internetworking.

11. Read Section 2, ''Network and Host Management,'' in the Internet section (Part 4) of this guide. You need to update your host database manually by editing the **/etc/hosts** and **/etc/networks** files.

12. If you will have any gateways on your network, read Section 5, ''Routing,'' in the Internet section (Part 4) of this guide. Set up the routing needed for your network.

13. Read Section 3, ''Networking Services,'' in the Internet section (Part 4) of this guide. Check your boot option files to make sure that the services you want are started at boot time. Add services if necessary.

14. Read Section 4, ''Internet UUCP,'' in the Internet section (Part 4) of this guide. This section explains how to set up UUCP to work with Ethernet. UUCP supports CTIX mail over Ethernet.

15. Read Section 6, ''Internet Security,'' in the Internet section (Part 4) of this guide. Decide what level of security your systems need, and plan the steps needed to implement that security.

16. Read Section 7, ''Equivalent Machines and Users,'' in the Internet section (Part 4) of this guide. Based on your security needs and the needs of your users, decide who will be given privileges on each system.

17. To learn more about monitoring and troubleshooting your network, read Section 6, ''Network Monitoring and Troubleshooting,'' in this Introduction. It gives an overview of basic procedures and refers to specific information in the Ethernet, SLIP, and Internet Administration Parts of this guide.

If You Are an Experienced UNIX Network Administrator

If you have performed the role of network administrator on other UNIX systems, the transition to CTIX should be relatively easy. Become familiar with the few differences between the organization of the *CTIX Operating System Manual* and the standard AT&T or Berkeley documentation.

Read through this guide looking for differences between CTIX and other systems you have used. Then follow the steps below before you start to configure your network:

1. Note any differences between the distributed configuration files and those to which you are accustomed. CTIX 6.2 network software is based on Berkeley UNIX release 4.3 networking.
2. Note that **/dev/console** is not associated with a physical device. The crontab file **/usr/spool/cron/crontabs/root** indicates where the console messages are sent, the default being the file **/etc/log/confile**. If you wish to change this, you must modify the appropriate entry in the crontab file.
3. Read the hardware installation guides you received with your computer. Continue reading for more information about related documents.

The CTIX Release Note for your particular version of CTIX is also a valuable reference.

Conventions

The following conventions are used throughout this guide:

- ''S/Series'' is used generically to refer to the S/80, S/22*x*, S/280, S/320, S/480, and S/640 computers, except where differences are noted in the text. ''S/MT'' is sometimes used to refer to the S/22*x*, S/320, and S/640 computers.

- Square brackets [ ] indicate an optional command argument.
- Braces { } indicate that there is a choice of required options.
- Ellipses (pn . . . pn) indicate that the previous parameter can be repeated any number of times.
- Boldface (**telnet**) indicates that the item must be typed exactly as shown in the command line. Names of CTIX programs and CTIX filenames are also shown in boldface.

  A name in boldface followed by a number enclosed by parentheses, such as **ps**(1), indicates that the item is described in the *CTIX Operating System Manual*; the number in parentheses is the section containing the item.
- Italic type (*input*) indicates that the user must supply the appropriate information (such as filename or variable name). Note that italics are also used for new terms that appear in the glossary.
- In all examples, administrative input is in boldface; the system's responses are in normal type; *########* indicates unechoed input; and **^D** indicates Control-D (**Code-D** or **Finish** on a Programmable Terminal [PT] or a Graphics Terminal [GT], the default End-of-File sequence).

Related Documents

Hardware

The *S/80, S/120, S/220, S/280, S/320, S480,* and *S640 Installation Manuals* provide instructions for setting up, configuring, and installing internal options in the S/Series computers.

The *S/80, S/480,* and *S/640 Technical Reference Manuals* provide principles of operation, register descriptions, and connector information, as well as theory of operation and software interface information for the S/Series computers.

The *Diagnostics Manuals* explain how to use the diagnostics programs for the S/Series computers.

The *MightyFrame VME Ethernet Controller Card Manual* contains a hardware description of the VME Ethernet card and procedures for installing it.

The *S/80 SCSI/LAN Board Technical Reference* provides software interface information, and host software control information about the board.

The *S/MT Series Ethernet Combo Board Technical Reference* provides installation information, with a description of the software interface and theory of operation.

Additional technical reference material can be found in the *S/MT Series VME Communications Controller Card Technical Reference* and the *S/MT Series VME Expansion Technical Reference* guides.

Software

The CTIX Operating System

The *CTIX Operating System Manual, Version C*, 2nd ed., describes the CTIX Operating System, an operating system derived from the UNIX System V Release 3.2 operating system. The manual describes CTIX commands, application programs, system calls, library subroutines, special files, file formats, games, miscellaneous facilities, and system maintenance procedures.

System Administration

The *CTIX Administration Tools Manual* describes how to administer a CTIX system by using the CTIX administration tools software package and without using CTIX shell commands.

The *S/Series CTIX Administrator's Guide* is a complete guide to CTIX system administration without using the CTIX administration tools.

Program Development

The *AT&T UNIX System V Release 3.2 Programmer's Guide* provides a detailed explanation of standard System V Release 3.2 programming tools and operating system facilities.

The *Programmer's Guide: CTIX Supplement* discusses support tools, text editing and formatting, and special CTIX programming issues. Use the supplement with the *AT&T UNIX System V Release 3.2 Programmer's Guide.*

The *Networking on the Sun Workstation* manual contains the following protocol specifications: the Network File System (NFS), the Yellow Pages (YP), the Remote Procedure Call (RPC), and the External Data Transfer (XDR).

The *CTIX Network Programmer's Primer* provides instructions for programmers on choosing a networking method. It also includes details of the CTIX operating system's implementation of networking standards.

Communications and Networks

The *CTIX BSC 2780/3780 RJE Terminal Emulator Manual,* 2nd ed., describes the daemon, configuration files, operational, maintenance, and line monitoring utilities, as well as the CTIX BSC device drivers, which are components of the Terminal Emulator. This manual is designed to assist end-users and system administrators who must configure, operate, and maintain the CTIX BSC 2780/3780 RJE Terminal Emulator.

The *CTIX BSC 3270 Terminal Emulator Manual* provides a product description, a list of IBM Information Display System components emulated by the 3270, CTIX BSC device drivers and servers used, and operational, maintenance, and line monitoring utilities. This manual is designed to assist end-users and system administrators who must configure, operate, and maintain the CTIX BSC 3270 Terminal Emulator.

The *CTIX SNA 3270 Terminal Emulator Manual* outlines the features and functions of the SNA 3270 Terminal Emulator. The manual provides detailed instructions for using, installing, configuring, and troubleshooting the Emulator software.

The *CTIX SNA LU6.2 APPC Server Manual* describes the features, functions, advantages, and components of the server. The manual is designed to assist system administrators and transaction programmers who must install, configure, maintain, and troubleshoot the LU6.2 Server.

The *CTIX SNA Network Gateway Manual* provides technical information and operating procedures for the installation, configuration, operation, and maintenance of a CTIX SNA Network Gateway. The manual also defines IBM SNA general concepts and describes the gateway components.

The *CTIX SNA PU Type 2.1 Network Gateway Manual* describes gateway installation, configuration, operation, maintenance, and troubleshooting.

The *CTIX SNA RJE Manual* describes the SNA RJE subsystem. Built on the SNA Network Gateway, SNA RJE allows multiple, concurrent logical unit sessions with remote IBM-compatible hosts. The manual describes user interface features, installation, and a procedural interface for user-defined RJE application systems.

Introductory UNIX Books

The following books, available through technical bookstores, cover basic UNIX concepts.

AT&T UNIX System V Release 3.2 User's Guide.

Morgan, Rachel, and McGilton, Henry, *Introducing UNIX System V3.2*, New York, NY: McGraw-Hill, 1987.

Supplementary Reading

The following publications provide additional information on topics covered in this guide. This information is not required to administer networking on CTIX computers.

All publications except the Internet *Requests for Comments* (RFCs) can be obtained through technical bookstores. The RFCs can be obtained from the Department of Defense Network Information Center. See Section 2 of the Internet section (Part 4) of this guide for information about contacting the Center. Note also that the RFCs listed here are only a few of many relevant to CTIX networking.

IEEE CSMA/CD Standard 802.3 (Ethernet).

Internet Protocol Transition Workbook, SRI International.

Internet Protocol - RFC-791

Transmission Control Protocol - RFC-793

Name, Addresses, Ports, and Routes - RFC-814

Internet Requests for Comments (RFCs), SRI International.

Ethernet Address Resolution Protocol - RFC-826

Broadcasting Internet Datagrams - RFC-919

Domain Names - Concepts and Facilities - RFC-1034

Domain Names - Implementation and Specification - RFC-1035

Internet Standard Subnetting Procedure - RFC-950

Stallings, William, *Local Networks*, New York, NY: Macmillan, 1984.

Tanenbaum, Andrew, *Computer Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1981.

Section 1
Networking Terms and Concepts

Network

A *network* is a group of computers set up to communicate with one another. A computer on a network is also known as a *host.* CTIX networks use three types of hardware connections:

- RS-232 cables
- Telephone lines
- Ethernet.

Network Addresses

On a network, each computer has an address. The address works like a telephone number in the telephone system. Each computer on the network has a unique number that other computers can ''call.'' This guide discusses the Internet addressing scheme, developed by the Defense Advanced Research Projects Administration (DARPA). Internet addressing is discussed in detail in the Internet section (Part 4) of this guide.

Naming schemes have also been developed that impose a logical structure on top of the underlying numeric addresses.

Network Gateway

A *network gateway* is a computer that passes data from one network to another. A gateway computer has more than one network interface, and each network interface has an associated network address.

Network Commands

Network commands are programs that users run to access remote computers. A network command can be anything from a simple utility to a complex application (for example, a distributed database). A computer running a network command is called a *client.*

The Ethernet, Internet, and SLIP Administration sections of this guide explain the use of network commands for administering and monitoring your network. The reference for the CTIX network commands is the *CTIX Operating System Manual*, Section 1.

Network Services

A *network service* is a computer program that answers requests from a corresponding network command (usually run from a remote computer). A computer running a network service is called a *server*. Services are also called daemons; their names, therefore, often end with the letter d.

For example, **inetd** (the internet daemon) is a program that runs at all times, listening for remote users who want to access network services on your system. When a remote user runs the **rlogin** command to log into your system, your **inetd** starts the **rlogind** (remote login daemon) program. The **rlogind** program starts a process that allows the log in (after a proper security check).

Administration of CTIX network services is covered in the Internet and SLIP sections of this guide. The reference for the CTIX network services is the *CTIX Operating System Manual,* Section 1.

STREAMS

An important feature of UNIX System V, Release 3.0 and subsequent releases (and therefore the CTIX 6.2 release) is the STREAMS mechanism. STREAMS is an enhancement to the UNIX character input/output (I/O) system that supports the development of communication software. The mechanism is especially applicable to networking, and it extends the older device driver concept.

A STREAM has three parts: a STREAM head, optional modules, and a driver (or STREAM tail). In the Internet section (Part 4) of this guide, Section 1 discusses administration of STREAMS for CTIX networking software.

Operating System Support

The CTIX operating system contains networking protocols and subroutines in the form of *network modules.* A module is a compiled (or assembled) file that is not by itself an executable program. From the administrator's standpoint, there are two types of network modules:

- Modules that are linked into the operating system when it is compiled
- Device drivers and STREAMS modules loaded when the operating system is booted.

The Ethernet, Internet, and SLIP sections of this guide explain how to load the modules that your system needs for networking.

Network Security

The purposes of security are to protect the following:

- Network software from accidental damage
- User data from unauthorized access
- User data from tampering.

The following basic security principles should be followed:

- Provide adequate physical security for your computers.
- Password protect all user accounts.
- Back the system up periodically; store the backup tapes in a safe place.
- Study this guide and related CTIX documentation, paying special attention to the security sections.

Network Performance

In planning a network, be aware of performance issues. This guide covers two types of CTIX network media:

- Ethernet
- Serial line (SLIP).

SLIP connections are good for mail, transfer of small files, and remote login by one or two users. SLIP can use telephone lines, and therefore can be configured over long distances. For heavy use and distributed computing, serial line networks are not recommended because they are too slow.

Be especially careful with mixed networks containing SLIP connections and Ethernet networks. Do not let a slow network link be a critical path for high volumes of data.

Seven Layer and OSI Models

Usually, networks are designed using a model of seven interacting levels. The lowest layer represents the computer hardware, while the highest is software for the end-user. An example of this layered design is IBM's *Systems Network Architecture* (SNA).

The seven-layered model (see Figure 2-1) was developed by the International Standards Organization (ISO) and is called the Reference Model of Open Systems Interconnection (OSI).

Figure 1-1 illustrates the OSI model:

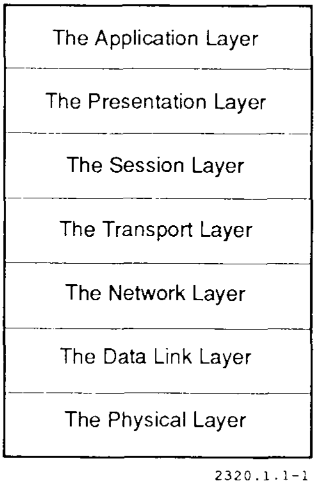


Figure 1-1. OSI Model

- The application layer represents compatible end-user programs that use a network to communicate.
- The presentation layer filters or transforms binary data from the lower layers into forms usable by the application layer.
- The session layer is the level where users negotiate connections (for example, remote logins) with other networked computers.
- The transport layer establishes source-to-destination reliability (if there is to be reliability) for the data sent through the lower layers.
- The network layer routes the data to their destination, through gateways if necessary.
- The data link layer provides an interface between a specific physical layer medium and a more general network layer protocol.
- The physical layer is the communications hardware itself, which implements the transmission of data bits from one computer to another.

Layering is used in the design of most network systems. Some networks have a different number of layers (often fewer).

Since the ISO is working on standard protocols for each layer of its OSI model, we refer to the model as the *seven layer model*; we refer to the *OSI model* only when discussing ISO standard systems.

Figure 1-2 relates the structure of this guide to the seven layer model.

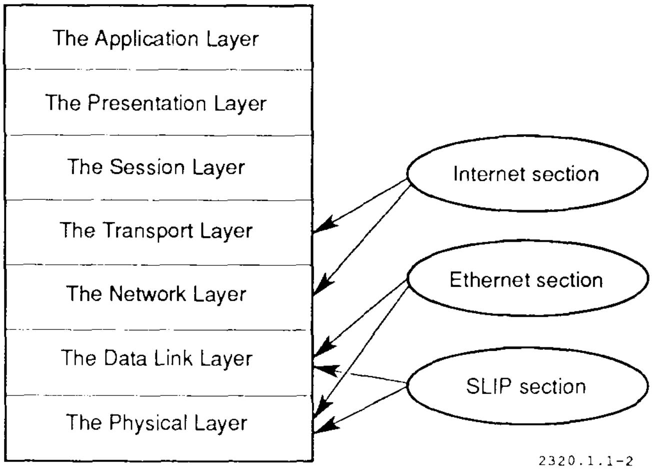


Figure 1-2. Relationship of the Seven Layer Model to this Guide

Section 2
Super-user Status and CTIX Modes

This section explains super-user status and single-user mode.

The basis for your ability to perform network administration is your use of super-user status. When using CTIX as the super-user, you can ignore CTIX restrictions on file access and allowable commands.

Normally a CTIX system runs in multiuser mode. Single-user mode is used for procedures that require an absence of normal disk activity: for example, rebooting the system. When CTIX is in this mode, only one terminal is usable. The single user in single-user mode has super-user status.

Super-user Status

Super-user status removes important CTIX restrictions. Most administrative CTIX commands in this guide require super-user status. CTIX gives the super-user three exemptions from normal restrictions:

- File read and write permissions do not apply to the super-user. The super-user can write to or read from any ordinary or special file. The super-user can create a file in or delete a file from any directory.
- Certain commands are executable only by the super-user.
- Certain commands have built-in safeguards or restrictions on the way they are used. Some of these safeguards and restrictions do not apply to the super-user.

When CTIX is running normally, there are two ways to obtain super-user status:

- Log in as user **root**.
- Use the substitute user program, **su**.

Both accesses to super-user status require knowledge of **root**'s user password. Consider **root**'s password sensitive information; change it regularly.

When CTIX is running in single-user mode, the sole user normally has super-user status.

The shell prompt changes to remind you that you are the super-user. Normally the default Bourne shell prompt is a dollar sign ($). When the super-user runs the shell, the default prompt is a pound sign (#).

Root User

In the password file, **/etc/passwd**, the user called **root** has numeric user ID 0; this value identifies **root** as the super-user.

Note that **root**'s password is a sensitive piece of information; anyone who knows it can become super-user. The **root** home directory is /.

su Program

To become a super-user while logged in as an ordinary user, use **su**, the substitute user program. Type

```
su
```

su prompts for a password; enter **root**'s password. If the password is verified, **su** runs the shell with its numeric user ID set to 0, giving the shell the same status as a shell run by **root**.

To return to normal user status, terminate the **su** shell with EOF (usually **Control-D**, or **Finish** on a Programmable Terminal or Graphics Terminal).

Rebooting the System

Rebooting the system does several things:

- The operating system is reinitialized. This is important if you have reconfigured your system or if some memory has become temporarily corrupted.
- All system and network daemons (services) are reinitialized. This is important if you have reconfigured your network services.
- All processes are killed. This can be valuable if unwanted processes are running on your system.
- The disk file systems are checked for inconsistencies and repaired if possible.
- Devices are reset. Power cycling is occasionally necessary to reset devices after unusual errors.

Note: *There are few situations (most of them listed above) in which you need to reboot the system. A system reboot should not be used as a quick fix to a network or system problem. If your network seems to have a problem, refer to one of the troubleshooting sections in this guide.*

There are two steps to rebooting your system:

1. Bringing the system to single-user mode
2. Rebooting the system to multiuser mode.

Section 4 of this Introduction discusses CTIX boot options. The rest of this section discusses single-user mode and rebooting your system; examples are included at the end of the section.

Single-user Mode

Single-user mode prevents ordinary users from communicating with the system. Users are barred from the system to prevent activity that might interfere with system maintenance.

CTIX can go to single-user mode in either of two ways:

- By commands from you, the administrator
- Automatically on start-up if CTIX decides it is not safe to go to multiuser mode.

When CTIX is in single-user mode, only one terminal is usable: the terminal that was used to take the system to single-user mode. The user on this terminal has super-user status.

Taking CTIX to Single-user Mode

To take CTIX to single-user mode:

1. Log in as **root**. (This is preferable to using the **su** program and changing your working directory to /, because a **root** login allows the proper unmounting of all file systems that should be unmounted.)
2. Run **shutdown** as follows:

   ```
   /etc/shutdown -g<grace>
   ```

   where *grace* is the number of seconds that users get to log out by themselves; if *grace* is omitted, users get 60 seconds. **shutdown**(1M) runs **wall**(1M) to warn the users, **killall**(1M) to terminate the users, and **init**(1M) to change the system mode; this process takes about a minute.

Be sensitive to users' needs when you bring the system to single-user mode. If the system shutdown can be anticipated, notify users in advance.

Taking CTIX to Multiuser Mode

To reboot the system to normal multiuser mode, press the reset switch on the processor cabinet or use the CTIX command:

```
reboot
```

Caution

Never reboot the system when it is in multiuser mode. To do so can cause damage to your disk file systems.

Examples

The following example shows how to become super-user and then return to normal user status.

```
$ su
Password: ########
# ^D
$
```

The following example shows a user logged in as **root** taking the system to single-user mode after giving users 2 minutes to log out. When the shutdown is completed, **root** reboots the system.

```
# cd /
# /etc/shutdown 120

SHUTDOWN PROGRAM

Fri Sep 11  9:58:30 PST 1987
```

```
Do you want to send your own message? (y or n): y
Type your message followed by ctrl d....


Taking system down for network administration.
You have 2 minutes to log out.
^D
```

```
Broadcast message from root (tty002) Fri Sep 11 10:00:27...
Taking system down for network administration.
You have 2 minutes to log out.
Broadcast message from root (tty002) Fri Sep 11 10:03:57...
SYSTEM BEING BROUGHT DOWN NOW ! ! !
All processes being killed.
```

```
Do you want to continue? (y or n):  y

Error logging stopped.


Slink exiting: SIGTERM.
cron aborted: SIGTERM

****    SYSCON CHANGED TO /dev/tty002    ****
INIT: New run level: S

INIT: SINGLE USER MODE
/ on /dev/dsk/c0d0s1 read/write on Sun Sep  6 10:52:19 1987
Entered Single User Mode on Fri Sep 11 10:06:39 PDT 1987
Ok To Stop Or Reset Processor
```

Type:

```
# reboot
```

Section 3
CTIX Boot Options

On a smoothly running CTIX system, networking is started when the computer boots up. Networking startup is controlled by the shell scripts /**etc/rc2**, /**etc/drvload**, and several scripts in the directory /**etc/rc2.d**.

A number of boot options can be enabled or disabled. These options are files in the directory /**etc/rcopts**; they are called **rcopts**.

Boot scripts check the directory /**etc/rcopts**. Some boot scripts look for the existence of **rcopts**, while other boot scripts look at the contents of the **rcopts**. Specific **rcopts** for network startup are discussed in the Ethernet (Part 2), SLIP (Part 3), and Internet (Part 4) sections of this guide.

Most **rcopts** are configured when software is installed.

Note: You must reboot your system before any changes made to **rcopts** *take effect.*

Modifying rcopts

Three administrative tasks must be performed for management of **rcopts**:

- Adding rcopts
- Deleting rcopts
- Changing rcopts.

All these tasks can be done with simple CTIX commands.

To add a **rcopt**, create the needed file in the directory **/etc/rcopts**. For example, to create the **rcopt** for Ethernet:

```
touch /etc/rcopts/KENP
```

To delete a **rcopt**, remove the file from the directory **/etc/rcopts**. For example, to delete the file from the previous example:

```
rm /etc/rcopts/KENP
```

To change a **rcopt**, edit the file with a text editor and make the needed changes. For example, to change the file in the previous examples (or add a file containing text):

```
vi /etc/rcopts/KENP
```

To cause the change to take effect, reboot your computer (as described in the previous section).

Section 4
Using the Administration Tools

Many network administration tasks can be performed with the CTIX administration tools. This section explains how to use the administration tools for network administration. The Internet, Ethernet, and SLIP sections of this guide give specific cases where you may use the tools.

Note that because the tools are automatic and incorporate error checking, using them usually speeds up your performance of a task and makes it more reliable.

Examples

The following example illustrates the use of the tools. To check the status of computers on your network, type (from the CTIX shell):

```
# adman
```

A menu such as the one in Figure 4-1 appears:

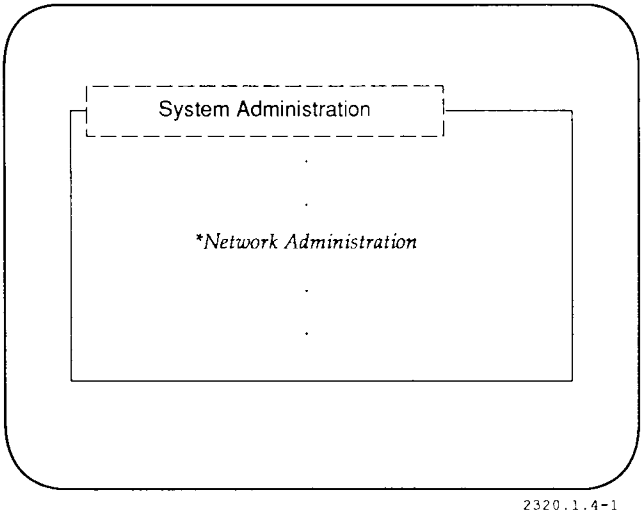


Figure 4-1. adman Menu

Select the *Network Administration* menu item.

The Network Administration menu (Figure 4-2) appears:

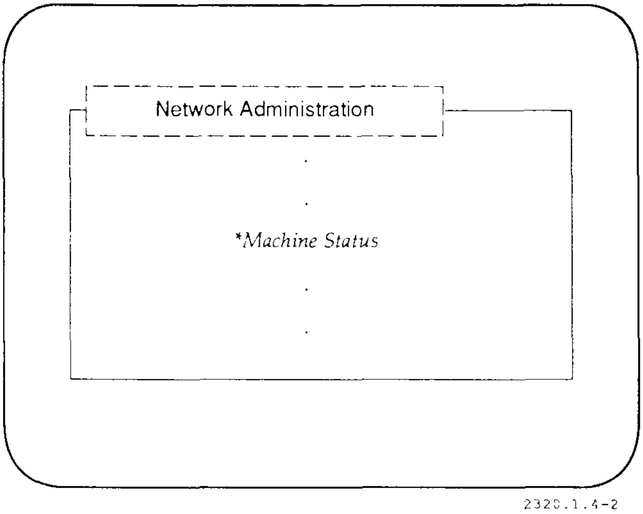


Figure 4-2. Network Administration Menu

Select the *Machine Status* menu item. The desired output appears in a new window.

In the following Parts (2, 3, and 4), flowcharts show how to follow the administration tools menus to the task you wish to perform. Figure 4-3 shows what the example above reduces to:

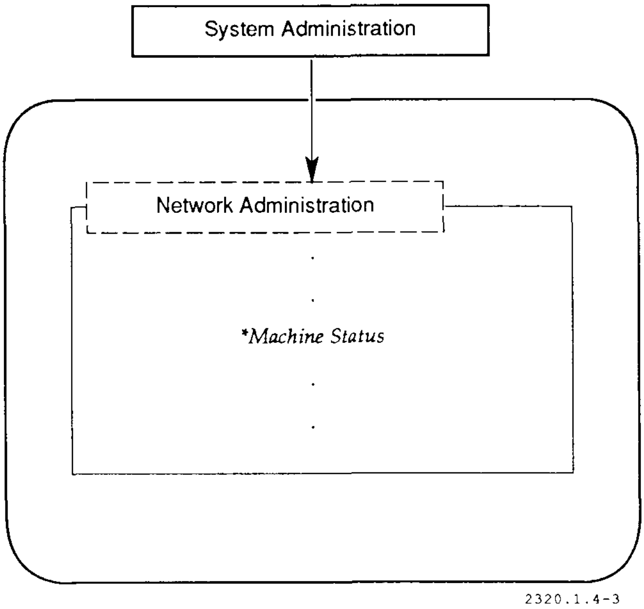


Figure 4-3. Typical Flowchart

Since the administration tools are enhanced frequently, your text and menu choices may be slightly different from the ones in this guide. See the software release notice for your software release for more details.

The administration tools have an on-line help facility to provide the administrator with basic information about using the tools. For reference information on the administration tools, see **adman**(1) and the *CTIX Administration Tools Manual.*

Section 5
Network Monitoring and Troubleshooting

To maintain your network, you need to know:

- How to monitor the network.
- Specific solutions for common network problems.

This section explains general monitoring and troubleshooting techniques. The parallel sections in the Ethernet (Part 2), SLIP (Part 3), and Internet Administration (Part 4) sections of this guide contain more information about monitoring and troubleshooting those subsystems.

Note: This section, and the corresponding sections in the Ethernet, SLIP, and Internet sections of this guide, are intended for technically proficient network administrators.

Monitoring the Network

To verify that your network is working properly, you must monitor the following:

- Which software modules are loaded in your operating system
- Network interfaces
- Each computer's ability to communicate with other hosts
- Network routing tables
- Network services
- Performance.

Checking the Operating System Modules

Use the **lddrv** command, as shown below, to check the software modules in the operating system:

```
/etc/lddrv/lddrv -sv
```

The output consists of entries like this:

```
DEVNAME   ID TYPE   BLK CHAR SIZE      ADDR    FLAGS
   stape      DRIVER -    65                     PRE-LOADED
     sxt   1  DRIVER -    27   0x2000 0x7fa31000 ALLOC BOUND
     plp   2  DRIVER -     6   0x1000 0x7fa33000 ALLOC BOUND
     pts   3  DRIVER -    23   0x1000 0x7fa34000 ALLOC BOUND
   clone   4  DRIVER -    80   0x1000 0x7fa36000 ALLOC BOUND
      sp   5  DRIVER -    81   0x1000 0x7fa37000 ALLOC BOUND
     log   6  DRIVER -    82   0x2000 0x7fa38000 ALLOC BOUND
  logbot   "  DRIVER -    83
   timod   7  STRMOD -    -    0x1000 0x7fa3a000 ALLOC BOUND
  tirdwr   8  STRMOD -    -    0x1000 0x7fa3b000 ALLOC BOUND
 llcloop   9  DRIVER -    96   0x1000 0x7fa3c000 ALLOC BOUND
      ip  10  DRIVER -    90   0x9000 0x7fa3d000 ALLOC BOUND
    icmp   "  DRIVER -    93
     udp  11  DRIVER -    92   0x2000 0x7fa47000 ALLOC BOUND
     tcp  12  DRIVER -    91   0x8000 0x7fa49000 ALLOC BOUND
     arp  14  DRIVER -    94   0x4000 0x7fa56000 ALLOC BOUND
 arpproc   "  STRMOD -    -
  socket  15  DRIVER -    97   0x8000 0x7fa5a000 ALLOC BOUND
```

To verify that a particular driver is in the operating system, you must know the following information:

- Module name
- Flags.

The module name is in the *DEVNAME* field above, and it should be the same as the first field of an entry in the **/etc/master** file. Of the flags mentioned above, **PRE-LOADED** means that this module was linked into the operating system (/**unix**) when the system was configured; *ALLOC BOUND* means that the module was dynamically loaded at boot time. [See **lddrv**(1M).]

It is also helpful to understand the module type. Of the module types above:

- **FSS** is a file system type.
- **STRMOD** is a STREAMS module.
- **SFTDRV** is a software driver.
- **DRIVER** is a device driver.

Checking the Interface Configuration

You can check the status of your network interfaces using the **ifconfig**(1M) command. There are three main interface types:

- Ethernet
- Loopback
- SLIP.

Ethernet Interface

To check your Ethernet interface, type the command

```
ifconfig en0
```

If your interface is configured, you should get output like this:

```
en0: flags=43<UP,BROADCAST,RUNNING>
inet 3.0.0.1 netmask ff000000 broadcast 3.255.255.255
```

- *flags* are the status of the interface (**UP, BROADCAST, RUNNING**).
- *inet* is the address family of the interface (Internet).
- *3.0.0.1* is the Internet address of the interface in this example.
- *netmask* # relates to subnetting. (See Section 2 of the Internet section (Part 4) of this guide.)

- *broadcast #* is the broadcast address for the network. (See Section 3 of the Ethernet section (Part 2) of this guide.)

The key flags for Ethernet are **UP**, **BROADCAST**, and **RUNNING**; other parameters vary.

If you have more than one Ethernet interface on your computer, the first one is **en0**, the second is **en1**, and so on. For other interfaces, substitute the appropriate name in the **ifconfig** command.

Loopback Interface

The loopback interface is a software interface that allows you to communicate with your own host using the network protocols. Unlike Ethernet and SLIP, it does not allow you to communicate with remote hosts.

An example of the use of the loopback interface is the command

```
rlogin loopback
```

This command allows you to log in on your own system.

The loopback interface is a useful debugging tool. By monitoring the loopback interface, you can test your computer's networking software without making assumptions about the state of network media and other systems. This makes it easier to isolate problems.

The loopback interface on CTIX systems is always associated with the Internet address 127.0.0.1 and the Internet name **loopback**.

Use the **ifconfig** command as follows to check the loopback interface:

```
ifconfig lo0
```

If the loopback is configured, you should get this output:

```
lo0: flags=49<UP,LOOPBACK,RUNNING>
inet 127.0.0.1 netmask ff000000
```

If your output differs, refer to "Troubleshooting the Network."

SLIP Interface

You can also use **ifconfig** to check SLIP interfaces (**sl0**, **sl1**, and so on). Note that the switched SLIP service attaches (configures) an interface when a call is made and detaches the interface when the call completes. This means that checking the interface does not tell you whether switched SLIP has failed.

Static SLIP interfaces normally are attached when the system boots. They do not need to be detached unless this is desired by the network administrator. See the SLIP section of this guide (Part 3) for more information about both types of SLIP.

Ping Command

The **ping**(1M) command is a simple utility for monitoring your network. It uses the ECHO_REQUEST feature of the Internet Control Message Protocol (ICMP) to elicit an ECHO_RESPONSE from a (usually remote) host. The **ping** command gives a simple test of your host's ability to send a packet to a host and receive a response.

A good way to use **ping** is shown below:

```
ping central 1 1
```

- *central* is the remote host name.
- *1* is the number of data bytes in the **ping** packet.
- *1* is the number of packets to send.

Note that you can use **ping** to check your local host, using either its Internet name or the **loopback** name.

See Section 9, ''Network Setup Samples,'' in the Internet Administration section (Part 4) of this guide for an example of the **ping** command; for more on the use of **ping** in troubleshooting, refer to ''Troubleshooting the Network'' below.

Performance Checks

It is important that you monitor system performance frequently; if you are familiar with the way your system perfoms, you can detect performance degradation quickly. Most often, you can tune the system to enhance performance; other times the degradation is a sign of trouble, which is best to notice right away.

The **netstat -m** command provides statistics about the network's buffer management. Use the command periodically to detect bottlenecks before they become a problem.

Troubleshooting the Network

Table 5-1 outlines possible problems and describes actions you can perform to fix the problems. In some cases you are referred to other sections of this guide. For example, if a suggested action is ''check Ethernet hardware,'' see the troubleshooting section in the Ethernet section of this guide (in Part 2); that section is arranged in a similar format to this section.

The table is arranged as follows:

1. *Symptom* is a problem you are experiencing.
2. *Additional Tests* are actions that help define the cause of the problem.
3. *Suggested Actions* are possible solutions to the problem. Some may not be applicable to your particular case, but correct the symptom in the majority of cases.

The additional tests and suggested actions help you understand the nature of the problem. If the problem cannot be immediately solved, contact your Technical Support representative.

Table 5-1. Problems and Solutions

Symptom	Additional Tests	Suggested Actions
Network command not found		Install TCP/IP software.
ping fails to remote host	Verify working hardware and isolate faulty hardware	Add missing **rcopt(s)**. For example, **KINET**. Add /**etc/hosts** file entry Add /**etc/networks** file entry Check routing to remote host by using **netstat -r** Try **ping** to the local host (or **loopback**) and check Ethernet or SLIP hardware
Network command (for example, **rlogin**) fails to remote host		Check Internet software
Performance degradation	Use **netstat -m** to check the buffer usage	Modify the appropriate tunable parameter

Glossary

A

administration tools

The CTIX administration tools software package is a set of programs and shell scripts that allow you to perform a number of system and network administration functions without using shell commands.

application layer

The application layer (layer 7) of the *seven layer model* represents compatible end-user programs that use a network to communicate.

C

client computer

A client computer is a computer that runs network commands. A client computer accesses *network services* on a *server computer.*

configuration files

The ASCII text files that define the various system-wide configuration parameters.

D

daemon

A daemon is a program that normally runs in the background: for example, **rlogind** (the remote login daemon), **rwhod** (the remote status daemon), and so forth.

data link layer

The data link layer (layer 2) of the *seven layer model* provides an interface between a specific physical layer medium and a more general network layer protocol.

E

equivalent machine

An equivalent machine is a remote computer to which you have given special permissions. Users on the equivalent machine with the same name as users on your machine are automatically *equivalent users.*

equivalent user

An equivalent user is a user on a remote system who is given the privileges of a user on the local system.

Ethernet

Ethernet is a 10-bps local area network system. Ethernet on CTIX computers follows IEEE standard 802.3.

H

host

A host is a computer on a network.

I

internet

Internet is a general term for the protocols, commands, and other software related to the Internet Protocol.

M

module

A module is a driver used to support an operating system function: for example, networking.

multiuser mode

Multiuser mode is the normal operating mode for CTIX.

N

network address

The network address is a numerical value identifying a computer. The network address is used in *routing.*

network gateway

A network gateway is a computer that passes data from one network to another. A gateway computer has more than one network interface, and each network interface has an associated network address.

network hardware

Network hardware includes the communications devices needed for networking: for example, network interfaces, cables, modems, and so forth.

network layer

The network layer (layer 3) of the *seven layer model* routes the data to their destinations, through gateways if necessary.

network service

A network service is a program running on a networked computer that provides a service to users of that computer or other users on the network. CTIX network services are *daemons.*

network security

The purposes of network security are: to protect the network software from accidental damage, to protect user data from unauthorized access, and to protect user data from tampering.

P

physical layer

The physical layer (layer 1) of the *seven layer model* is the communications hardware itself, which implements the transmission of data bits from one computer to another.

presentation layer

The presentation layer (layer 6) of the *seven layer model* filters or transforms binary data from the lower layers into forms usable by the application layer.

protocol

A protocol is a set of rules for the format and timing of data exchanged between communicating systems.

R

routing

Routing is the process by which data are directed from one *host* to another. If the hosts are on different networks, data are transmitted step by step through network gateways until their destinations.

S

server computer

A server computer offers *network services* to other computers on the network (*client computers*).

session layer

The session layer (layer 5) of the *seven layer model* is the level where users negotiate connections (for example, remote logins) with other networked computers.

seven layer model

The seven layer model is a general design model for network systems. See also *physical layer*, *data link layer*, *network layer*, *transport layer*, *session layer*, *presentation layer*, and *application layer*.

single-user mode

Single-user mode is the CTIX mode that allows only one terminal to be in use. The single user in single-user mode has super-user status. Single-user mode is used for procedures that require an absence of normal disk activity.

SLIP

SLIP (Serial Line Internet Protocol) is a set of communication programs that allows users to use Internet commands between computers connected with a serial line.

STREAMS

STREAMS is an enhancement to the UNIX character input/output (I/O) system that supports the development of communication software. STREAMS is new to the AT&T UNIX 5.3 and the CTIX 6.0 releases.

super-user

Super-user status is conferred on a user whose user id is 0. Super-user status removes important CTIX restrictions: the super-user can write to or read from any ordinary or special file; the super-user can also execute certain commands that no other user can execute.

T

transport layer

The transport layer (layer 4) of the *seven layer model* establishes source-to-destination reliability (if there is to be reliability) for the data sent through the lower layers.

tuning

Adjusting the values of system variables to enhance system performance.

U

UUCP

UUCP (UNIX-to-UNIX copy program) is a batch-oriented file transfer protocol that can run on direct serial lines, telephone lines, and networks.

Index

Contents for Part 2

Contents for Part 2

Figures for Part 1

Tables for Part 1

Section 1
Ethernet Terms and Concepts

Ethernet is a fast (10-Mbps) network system. The layout of an Ethernet network is a simple bus; each computer or peripheral is attached to a common coaxial cable that serves as the transmission medium. The ends of the transmission cable are not joined and the cable does not double back on itself, allowing a single connection between transmission points.

Ethernet on CTIX computers follows IEEE standard 802.3, which was developed from the original Xerox Ethernet. Note that CTIX Ethernet is not compatible with Xerox Ethernet.

Placement in Seven Layer Model

Ethernet spans two layers of the seven layer network model, the physical layer and the data link layer.

The Ethernet physical layer (layer 1 of the seven layer model, which was shown as Figure 1-1 in Part 1) provides for data transfer on the network. This layer, which is implemented in the Ethernet transceiver and on the Ethernet interface card, includes data collision detection. Collision means that two hosts are transmitting data onto the coaxial cable at the same time. When a collision occurs, the data must be retransmitted.

The Ethernet data link layer (layer 2 of the seven layer model) provides an interface between the physical layer of Ethernet and the network layer protocol.

Compatibility

CTIX computers are designed to be compatible with each other over an Ethernet network.

Ethernet specifies only the lowest two layers of a network architecture. If a computer is to communicate with other computers, all the relevant layers must match. Thus, it is possible that two computers on an Ethernet network cannot communicate because their higher level protocols do not match.

Hardware

Figure 1-1 illustrates the hardware setup for an Ethernet network.

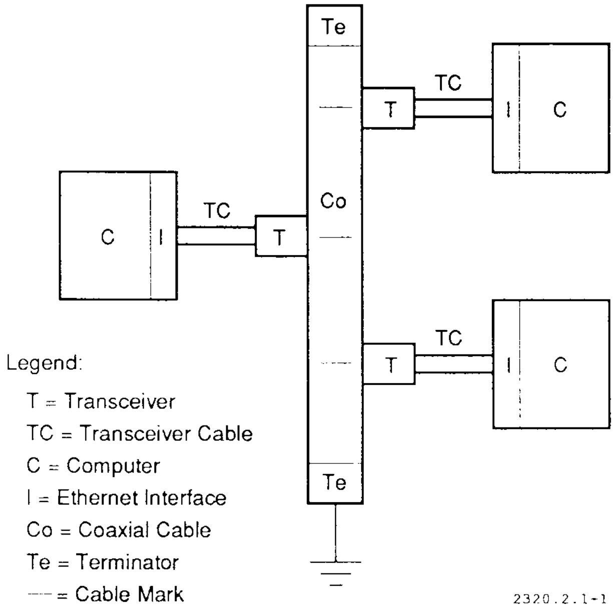


Figure 1-1. Ethernet Network

Coaxial Cable

The coaxial cable is the backbone of an Ethernet network. The coaxial cable is a shielded cable that serves as the transmission medium for Ethernet networks. The coaxial cable must have a terminator at each end, and one end must be grounded. Up to 100 connections (taps) can be made to a single coaxial cable segment.

The maximum length of an Ethernet coaxial cable segment is 1500 feet. This length can be extended by using a repeater to connect more than one cable segment. Another extended configuration using two cables involves a gateway computer. (See Figure 1-2.)

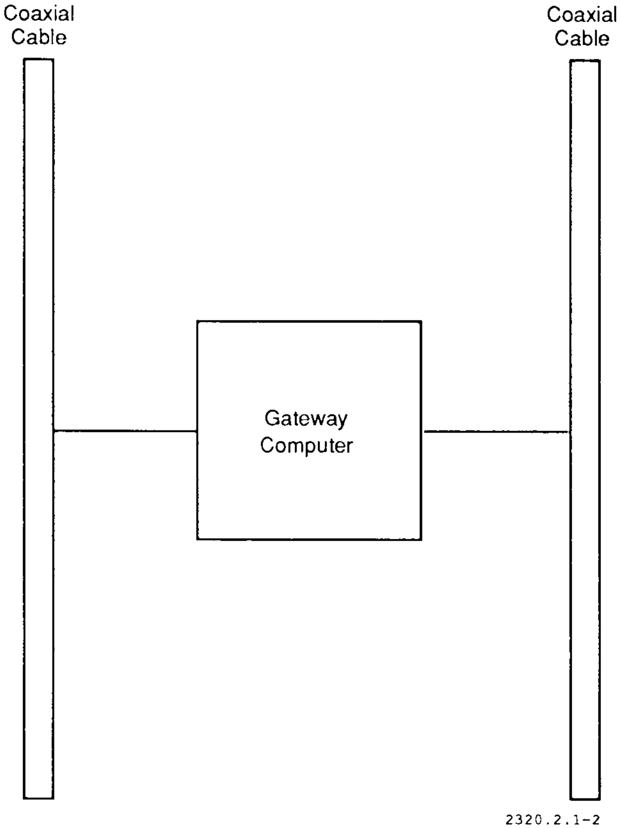


Figure 1-2. Ethernet Gateway Configuration

Ethernet can use either of the following types of coaxial cable: thick (0.4 inch) or thin (0.2 inch). With thick cabling, the cable is usually pierced with a cable coring tool and tapped to make a connection. Thin cable connections are made using plug-in connectors; occasionally this method is also used with thick cabling. Note that throughout the Ethernet section (Part 2), thick cabling (with taps) is assumed unless otherwise specified.

Transceivers

The transceiver is an electronic device that connects the coaxial cable to each computer. It detects signals sent from other computers and signal collisions that occur when two or more computers try to broadcast at the same time. Transceivers must be based on IEEE standard 802.3 to be compatible with CTIX Ethernet interfaces.

Ethernet transceivers must be attached to the coaxial cable at multiples of 2.5 meters. Most Ethernet cables are marked to show where transceivers can be attached. For thin cabling, transceivers are attached to the coaxial cable using plug-in T connectors; standard length coaxial cables with plug-in connectors on both ends are used to extend the coaxial cable.

Ethernet Controller

The Ethernet controller is located on an expansion card that interfaces to the computer and supports the Ethernet locally. Each card contains its own dedicated microprocessor. Three Ethernet interface options exist for S/Series machines:

- Ethernet RS-232 Board (S/22*x*, S/320, S/480, and S/640 only)
- SCSI/LAN Board (S/80 and S/280 only)
- VME (CMC) Ethernet controller (S/22*x*, S/320, S/480, and S/640 only).

Note that the CMC Ethemet controller is used only on machines with VME capability.

Transceiver Cable

The transceiver cable connects the Ethemet interface card with its transceiver.

Ethernet Repeater

The Ethemet repeater is an optional device used to extend the length of an Ethemet network. It consists of a special amplifier used to connect two Ethemet coaxial cables.

Section 2

Hardware Setup

This section describes how to set up an Ethernet network. Two options are described in this section: a standard setup with one transceiver for each computer on the network, and the multiport transceiver.

Setting Up the Network

Figure 2-1 shows a typical Ethernet network. To set up the hardware for an Ethernet network, install the following:

1. Ethernet interface for each computer on the network.
2. Coaxial cable.
3. Transceivers.

Materials Needed

The following paragraphs list the equipment needed to set up the Ethernet hardware.

Ethernet Interface Installation

- Ethernet interface
- Instructions

Coaxial Cable Installation

- Ethernet cable (no more than 1500 feet in length)
- Barrel connectors (as needed) to join sections of cable
- Two 78-ohm terminators
- One cable ground clamp
- One ground wire (insulated)
- Wire strippers
- Electrical tape.

Transceiver Installation

- Open-end wrench (9/16-inch)
- Cable coring tool
- Shield removing tool
- Inner shield piercing tool
- Tweezers or small needle-nose pliers
- Tap blocks (one per network node)
- Transceivers (one per network node)

- Transceiver cables (one per network node)
- One can of compressed air (optional).

Ethernet Network

Figure 2-1 illustrates an Ethernet network of S/Series computers.

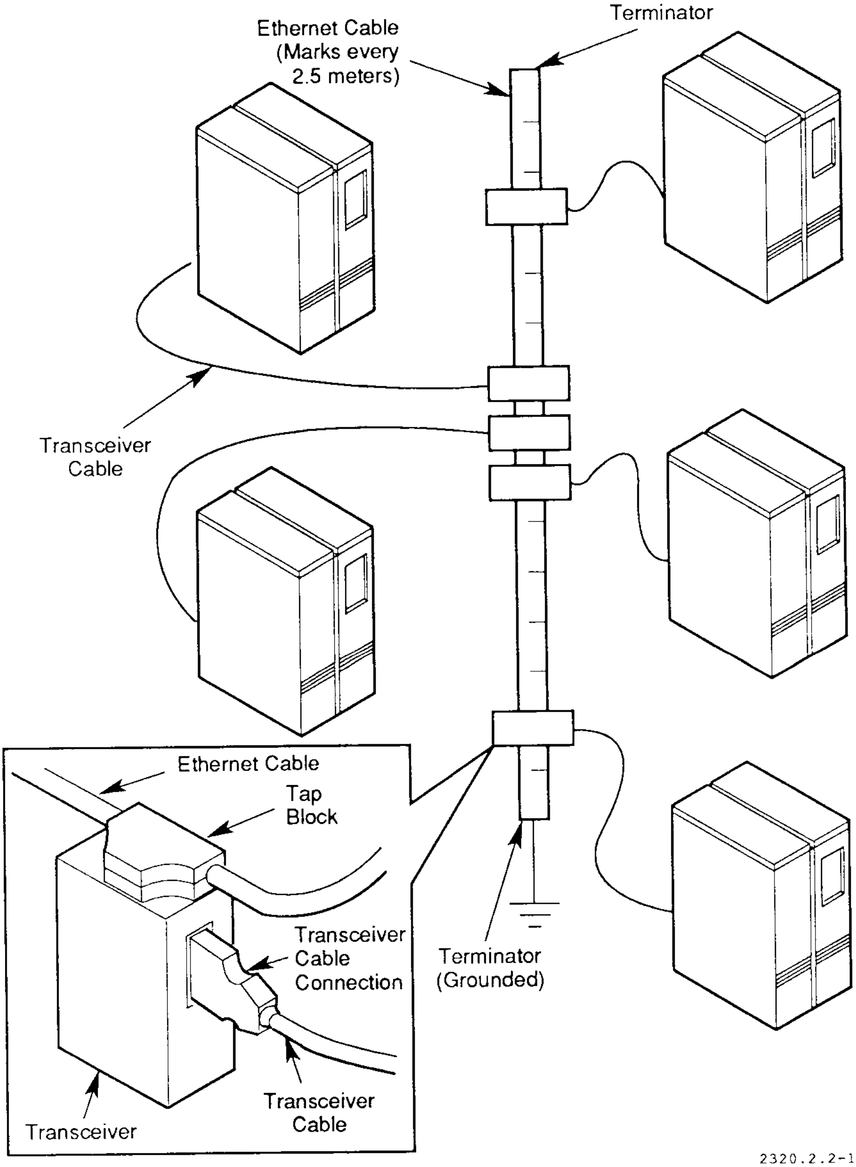


Figure 2-1. Example of an Ethernet Network

Installing the Ethernet Interface

Install the Ethernet interface card for each computer on your network. Installation instructions are included with the card.

The three Ethernet interface options for the S/Series are the VME Ethernet card (S/22*x*, S/320, S/480, and S/640 only), the Ethernet RS-232 Board (S/22*x*, S/320, S/480, and S/640 only), and the SCSI/LAN Board (S/80 and S/280 only).

The following restrictions apply to the Ethernet RS-232-C Board:

- The S/480 and S/640 computers contain three DMA channels each.
- The S/320 computer contains two DMA channels for intelligent processors. One can be assigned for VME bus controllers (the other can be assigned for the IOP or RS422 controller); one or more (up to four) Ethernet RS-232-C controllers can share *one unassigned* DMA channel.

  The S/320 can contain the following controller combinations:

  — One or more Ethernet RS-232-C controller(s) and one or more VME controller(s)

  — One or more Ethernet RS-232-C controller(s) and an IOP or RS422 controller.

  The system *cannot* contain an Ethernet RS-232-C controller with both an IOP/RS422 controller and a VME controller.
- The S/80 and S/280 computers contain onboard SCSI buses, to which you can connect a SCSI/LAN board.

Network Layout

If you have not done so already, plan your network layout and location of taps. Note that the distance between taps must be in multiples of 2.5 meters (8.2 feet). The Ethernet cable is marked with a heavy black bar every 2.5 meters to help you determine tap locations.

Setting Up the Coaxial Cable

Build your Ethernet cable using barrel connectors to connect cable segments. Remember that the length of the Ethernet cable must not exceed 1500 feet. Terminate each end of the cable with the 78-ohm terminators. Route your cable according to your network layout.

Next, select the terminator that is closest to an existing ground termination point or a separately driven ground rod, and ground the cable *at one end only*, as described in the following steps:

1. Clamp the cable ground clamp around the terminator you have chosen as the ground origin. Do not overtighten the clamp.
2. Cut the ground wire to fit between the terminator and the ground termination point.
3. Strip the insulation from 1/2 inch of each end of the ground wire.
4. Attach one end of the ground wire into the receptacle in the ground clamp. Tighten the screw holding the ground wire in place.
5. Attach the other end of the ground wire to the ground termination point.
6. Insulate all connector junctions and terminators by wrapping them with electrical tape.

Installing Transceivers

A transceiver (with tap block) must be installed at each network node. To install the transceiver at a node, follow these steps:

1. Unscrew the transceiver from the tap block (only if the transceiver and block were shipped together).

Caution

Do not install a tap block where it can touch grounded objects (conduit, piping, and so on). Interference with grounded objects can impair data integrity.

2. Clamp the tap block onto the cable at the planned node location. The threaded hole in the tap should point toward the node being installed. Tighten the clamp until the tap block halves are securely clamped together.
3. Insert the cable coring tool into the threaded hole in the tap block. Screw the tool in until the threads gently bottom out. Then screw it partly out and back in a few times. Remove the coring tool.
4. Carefully inspect the hole in the cable for stray wires, and remove all loose particles of insulation and shielding.
5. Repeat steps 3 and 4 to verify that the cable is properly cored.
6. Insert the shield removal tool into the hole. Apply heavy hand pressure and rotate the tool *clockwise* for several revolutions. This removes shielding braid and foil. Remove all loose particles using tweezers or needle-nose pliers.

Caution

Be sure that all loose fragments are removed from the hole. Loose fragments can short-circuit the coaxial cable when the transceiver is installed and disable the network.

7. Screw the inner insulation piercing tool into the tap block until the threads bottom out. Remove the tool.
8. If one is not already in place, install the o-ring onto the connector threads of the transceiver. Screw the transceiver onto the tap block. Finger-tighten the transceiver; *do not overtighten.*
9. Connect the appropriate end of the 15-pin transceiver cable to the transceiver. Secure the cable in place with the slide lock.
10. Connect the other end of the transceiver cable to the Ethernet interface card.

Removing Transceivers

A transceiver can be removed (permanently or for relocation) at any time.

Caution

You do not have to power down the network when you remove transceivers. However, if you accidentally touch the shield to the connector, you will short-circuit 12 V to ground and cause the network to malfunction. Exercise extreme caution when you remove transceivers from a "live" network.

1. Unscrew the transceiver from the tap block. Keep the o-ring with the transceiver.
2. Install a tap block plug into the empty tap block. You can reuse this location at any time by removing the plug and reinserting a transceiver.

Multiport Transceiver

Another option for setting up an Ethernet network is to use a multiport transceiver. A multiport transceiver, also known as a multiplexer, is a simple device that does the work of several (typically eight) transceivers.

It is useful in two types of situations:

1. A small network where all computers are within 50 feet of each other. (See Figure 2-2.)
2. A work group with several computers connected to a network, where the network as a whole extends more than 50 feet, but the computers in the work group are all within 50 feet of each other. (See Figure 2-3.)

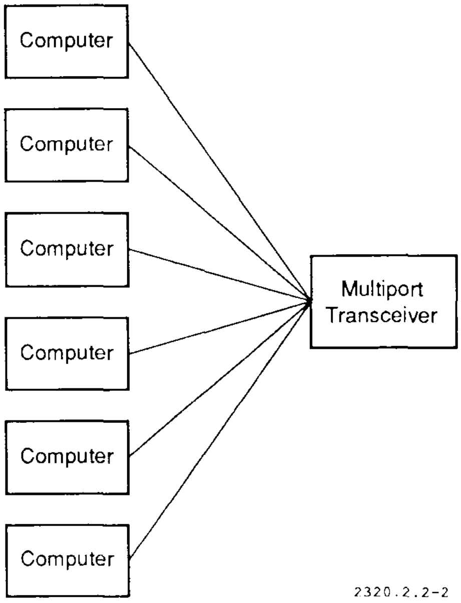


Figure 2-2. Local Network with Multiport Transceiver

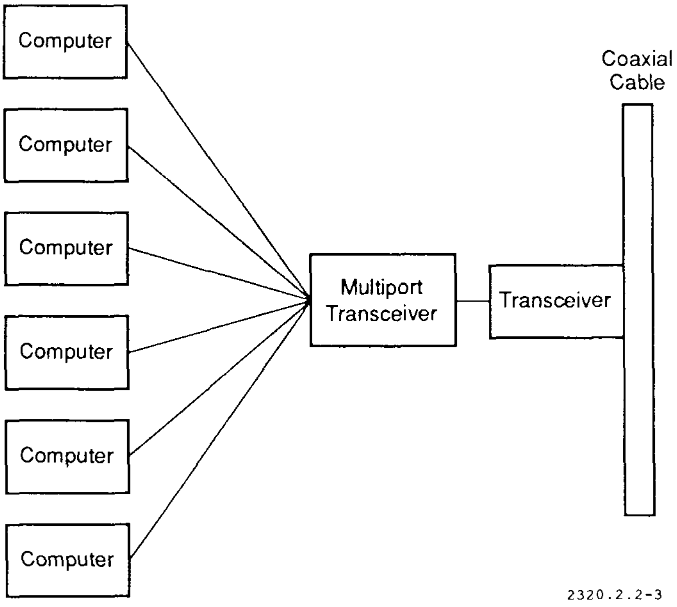


Figure 2-3. Hosts Connected to Ethernet Network by Multiport Transceiver

In a small network situation, the multiport transceiver acts in local mode. Follow these steps:

1. Switch the multiport transceiver to local mode.
2. Connect the computers to the multiport transceiver with transceiver cables.

Note that ordinary transceivers and coaxial cables are not needed in this configuration.

In the work group situation, the multiport transceiver acts in remote mode. Follow these steps:

1. Switch the multiport transceiver to remote mode.
2. Connect an ordinary transceiver to the coaxial cable.
3. Connect a transceiver cable from the ordinary transceiver to the multiport transceiver.
4. Connect the computers to the multiport transceiver using transceiver cables.

What To Do Next

Go on to Section 3 for information on how to set up the software for your network.

For a complete description of the steps needed to set up your network, see the section called "About this Guide," in the front of this manual.

Section 3
Software Setup

Setting Up the Software for Ethernet

To set up your Ethernet network software you must do the following:

- Configure your VME Ethernet address (only if you are using the VME Ethernet interface card).
- Verify that your Ethernet interface is configured in the **rcopts** boot options.
- If you are setting up a gateway computer, edit the appropriate configuration files.

If you are administering a network with old (5.*x* CTIX or 4.2BSD) and new (6.*x* CTIX and/or 4.3BSD) systems on it, you may optionally decide to change the *internet broadcast address* for your new systems. The internet broadcast address of an interface is the generic address used for sending data to all hosts on the network. The last topic in this section explains how to do this.

Configuring the VME Ethernet Addresses

If you have one VME Ethernet board on your computer, configure the VME Ethernet address by performing the following steps:

1. Edit the file **/etc/system** using your editor of choice. In the **!VMESLOTS** section of the file, find the lines that look like this:

```
* slot  type    address length  [ Init-routine-name ]

* (1 CMC ethernet controller - each needs 128K of A24 space)
* 0     1       CODE0000        131072
```

(Note that the header line might not appear in the same position in the file as shown in this example; it is included here for clarity.)

2. Remove the leading asterisk (*) in column one of the Ethernet controller entry. (The line above the entry is a descriptive comment only.) The lines should now look like this:

```
* slottypeaddresslength[ Init-routine-name ]

* (1 CMC ethernet controller - each needs 128K of A24 space)
0       1       C0DE0000        131072
```

3. If you have more than one Ethernet board, perform the steps shown above to add more lines to the **/etc/system** file. The Ethernet board configuration entry consists of four fields:

   - Slot number
   - Board type (always 1 for Ethernet)
   - Address
   - Address length (always 131072).

The base address is assigned according to the following table:

Board Number	Address
0	C0DE0000
1	C0E00000
2	C0E20000
3	C0E40000
4	C0E60000
5	C0E80000
6	C0EA0000
7	C0EC0000

Note that these addresses correspond to the addresses that you set with the jumpers as described in the *Controller Installation Manual*; here, however, they are preceded by **C0**.

Therefore, if the board configured above is in slot 0, it is board type 1, at address C0DE0000, with address length 131072.

To configure another board for this example, add another entry to the **/etc/system** file. With two boards configured, the Ethernet entries in the **/etc/system** file look like this:

```
* slot  type          addresslength[ Init-routine-name ]

* (1 CMC ethernet controller - each needs 128K of A24 space)
0       1       C0DE0000        131072
1       1       C0E00000        131072
```

Entries for more boards can be added in the same manner. Note that when determining slot numbers, you must consider any other VME boards on your system. If you have an SMD controller in slot 0, the Ethernet board should be configured for slot 1, and so on.

4. Log in as root and load the new configuration into non-volatile memory by using the **ldeeprom** command as follows:

   ```
   /etc/lddrv/ldeeprom
   ```

   You are now finished with the VME Ethernet address configuration. Continue with the steps in this section, and be aware that you must reboot the system for the changes to take effect.

Ethernet Interface Configuration

The file **/etc/rcopts/KENP** controls the configuration of the Ethernet interface(s).

Single Interface Computer

If your computer has one Ethernet interface, you should have an empty **/etc/rcopts/KENP** file. Use the **ls** command as shown below to check for its existence:

```
ls -l /etc/rcopts/KENP
```

The **ls** command displays the following listing if the file exists:

```
-rw-r--r--  1 root  root     0 Jun 8  1987 /etc/rcopts/KENP
```

If the file does not exist, use the following command to create it:

```
touch /etc/rcopts/KENP
```

Gateway Computer

Gateways have two or more Ethernet interfaces. To configure a gateway, you must know the following about each Ethernet interface:

- The network number (from the **/etc/networks** file) associated with the interface.[1]
- The device name associated with the interface. This is determined by the hardware configuration of the machine. Interfaces are configured in the following order:

  1. VME interfaces in slot order
  2. Ethernet RS-232 Boards.

  The first interface configured is **en0**; the second interface is **en1**; and so on.
- The internet address associated with the interface. This is configured in the **/etc/hosts** file.[2]
- The internet name associated with the interface. This is also configured in the **/etc/hosts** file.[3]

To configure a gateway, you must edit two files: **/etc/system** and **etc/rcopts/KENP**.

Add the following line, under the **!TUNEABLES** section, to **/etc/system**:

```
net_ipforwarding=1
```

Use the **/etc/uconf**-w command program to update the operating system image with the new tunable parameter value; when the system is rebooted, the new value takes effect.

[1] See Section 2 of the Internet section of this guide.
[2] See Section 2 of the Internet section of this guide.
[3] See Section 2 of the Internet section of this guide.

The **/etc/rcopts/KENP** file must contain the internet names of the interfaces, on one line and separated by blanks, in the order of the interface device names (**en0**, **en1**, and so on). For example, a system with the following interfaces:

Device Name	Network Number	Internet Address	Internet Name
en0	22	22.2	personl
en1	2	2.7	personl-gw

would have the following **KENP** file:

```
personl personl-gw
```

The device names, network numbers, internet addresses, and internet names must be consistent. To check this:

- Verify that the Ethernet interface is connected to the proper transceiver cable.
- Check that the network number associated with the interface matches the network part of the internet address.
- Verify that the internet name is in the correct position in the **KENP** file.

The **KENP** file is interpreted by the script **/etc/rc2.d/S79devices** and two steps are performed automatically when the system is booted:

1. Each Ethernet interface is loaded with its download image.
2. Each interface is configured by using **ifconfig**(1).

Tuning for the Ethernet Driver

The CTIX system provides *tunable parameters* to help you modify the system load and enhance performance.

To change the value of a tunable parameter:

1. Edit the parameter's value in the **/etc/system** file.
2. Use the **uconf**(1M) command with the **-w** option to update the system memory image.
3. Reboot the system.

Once you add an Ethernet driver to your system, it is important that you tune your system for it; otherwise, the number of STREAMS buffers can become a bottleneck for system performance. Add (or edit) the following three lines in the **/etc/system** file:

```
v_nblk4096=16
v_nblk2048=64
v_nblk1024=24
```

Broadcast Address Issues

Note: This section discusses an incompatibility between old and new internetworking software. One suggestion for resolving this problem is to upgrade your old software to the new release. If that strategy is your choice, ignore this topic. If you need to keep computers at the old release, you can still consider ignoring the problem, as it only affects the output of **rwho***(1) and* **ruptime***(1) on computers using the old release.*[4]

The internet broadcast address is the internet address used for sending data to all hosts on the network. In the old release, the broadcast address was made up of a network number and a host number of zero. In the new release, the default broadcast address is a network number with a host number consisting of all ones in binary representation. For example, for network 3 the old broadcast address was 3.0, and the new address is 3.255.255.255. The new release recognizes the old broadcast address, but the old release does not recognize the broadcast address of the new release.

The effect is that on a network with both new and old systems, the old systems do not receive **rwho** daemon status from the new systems. Thus, the new systems are invisible to the old systems in the output of a **rwho** command or a **ruptime** command.

The remedy is to set the broadcast addresses of the new machines to the old broadcast address. You can do this by adding a line or lines like this to the **/etc/rcopts/ROUTING** file:

ifconfig *interface* broadcast *net*.0.0.0

[4] In this discussion, old release means 4.2BSD or CTIX 5*.x* (TCP/IP 2*.x*); new release means 4.3BSD or CTIX 6*.x* (TCP/IP 3*.x*) or later.

For example, the line for Ethernet interface **en0** on network 3 is:

```
ifconfig en0 broadcast 3.0.0.0
```

Note that your address may look slightly different based on the address class of your network. See Section 2, ''Network and Host Management,'' in the Internet Part of this guide for more information.

What To Do Next

Continue to Section 1 of the Internet Part of this guide for information about how to set up the underlying software for your network.

For a complete description of the steps needed to set up your network, see Part 1, ''Introduction,'' of this manual.

Section 4
Ethernet Monitoring and Troubleshooting

This section explains how to monitor and troubleshoot your Ethernet network.

Address Resolution Protocol Status

The Address Resolution Protocol (ARP) translates between Ethernet and internet addresses. [See **arp**(7).] You can use the **arp**(1M) command as shown below to monitor the ARP tables:

```
arp -a
```

The output looks like this:

```
sales (3.0.0.1) at 2:cf:1f:10:14:72
mktg (3.0.0.2) at 0:11:0:3d:0:0
```

The first text field is the system name of the remote system; the dot-delimited number in parentheses is your internet address. The colon-delimited number is your Ethernet address. Note that an ARP table entry should be created automatically when your computer contacts a remote host.

Interface Status

To view the status of your computer's Ethernet interface, follow the administration tools procedure shown in Figure 4-1:

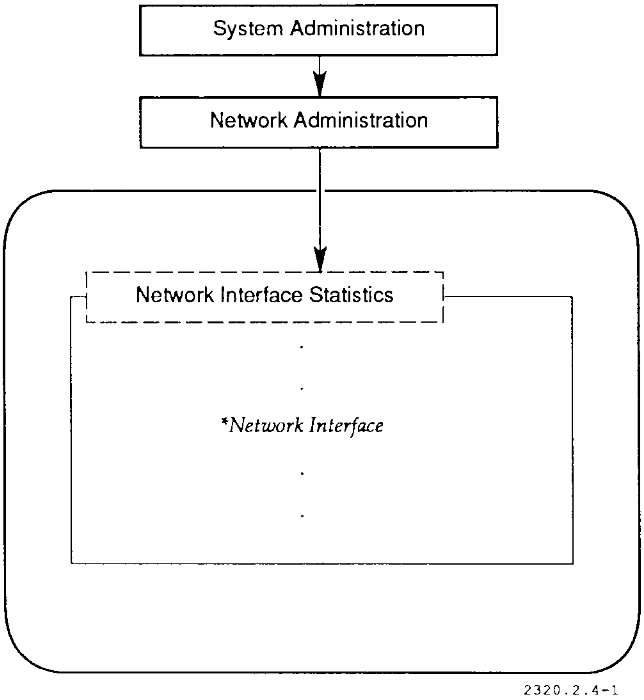


Figure 4-1. Administration Tools

This is the same as using **netstat**(1) with the **-i** option. Normal output looks like this:

```
Name  Mtu   Network   Address    Ipkts  Ierrs Opkts  Oerrs Collis
en0   1500  ThreeNet  MyMachine  26029  0     13960  0     0
lo0   2048  Loopback  loopback   216    0     216    0     0
```

The interface name for the first Ethernet interface is *en0*. To verify that the interface is working, check the following:

- The number in the *Ipkts* column should be greater than zero. This is the number of input packets transmitted to the interface.
- The number in the *Opkts* column should also be greater than zero. This is the number of output packets transmitted from the interface.
- *Ierrs* and *Oerrs* are error counters. Their values should be much lower than *Ipkts* and *Opkts*. High error counts can sometimes indicate Ethernet hardware problems. See below for more information about troubleshooting Ethernet hardware problems.

Troubleshooting Ethernet Hardware

In troubleshooting Ethernet hardware, it is important to isolate the faulty component or to reduce the number of possible faulty components by verifying that some parts of the network hardware work properly.

The first thing to do is to isolate the problem. Perform the following procedure:

1. Perform a test to isolate the problem.
2. Observe the feedback from the test and based on that feedback, create a solution to the possible problem.
3. Test the solution. If it doesn't work, you may not have isolated the true problem.
4. Return to step 1, performing a *new* test to isolate the problem.

The **ping**(1) command can help to isolate a problem, as shown in Section 6 of Part 1 of this manual. If **ping** to a remote host succeeds, test **ping** to your local host (or **loopback**).

Figure 4-2 shows a simple Ethernet network (or part of one) that is not working properly; three computers (one, two, and three) communicate on the network:

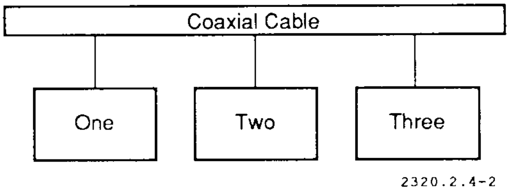


Figure 4-2. Simple Hardware Setup

In this example, users on computer two complain that their network commands are not working properly. If you have already checked that the software is set up correctly, try to communicate with computer three from computer one. Use the command:

```
ping three
```

If this command works, then the Ethernet interfaces on computers one and three are working. Next, check the other hardware components; three likely problems might be:

- The Ethernet interface on computer two is not working.
- The transceiver connected to computer two (or its connection to the coaxial cable) is bad.
- The transceiver cable on computer two (or one of its connections) is bad.

Use the **ifconfig** command as shown below to check the Ethernet interface on computer two:

```
ifconfig en0
```

If the interface is marked ''UP'', it is probably working. The transceiver and its cable are probably the cause of the problem. To fix the problem, perform the following:

1. Check all cable connections.
2. Verify that the appropriate terminators are in place.
3. Re-tap the coaxial cable (move the transceiver to a different mark).
4. Replace the transceiver.
5. Replace the transceiver cables.

Troubleshooting Table

Table 4-1 outlines possible problems and solutions to help you troubleshoot your Ethernet hardware problems. The table is arranged as follows:

- *Symptom* is a problem you are experiencing.
- *Additional Tests* are actions you take to help define the cause of the problem.
- *Suggested Actions* are possible solutions to the problem. Some of them may not be applicable to your particular case, but correct the symptom in the majority of cases.

Table 4-1. Problems and Solutions

Symptom	Additional Tests	Suggested Actions
ping fails to remote host	Follow procedure to verify working hardware and isolate faulty hardware	Check for missing **KENP rcopt**
		Check **/etc/system** file for proper configuration of VME Ethernet interfaces
		Check interfaces by using **ifconfig**. Verify that the interfaces are connected to their proper networks with correct addresses
		Replace hardware component(s)
You cannot contact remote host	Use the **netstat -i** command, checking for output or input packet errors	Check cables
Poor performance	Use **netstat -m** to check for relatively high failure rates in the STREAMS buffer blocks	Increase the value of the appropriate tunable parameter

Glossary

A

Address Resolution Protocol

The Address Resolution Protocol (ARP) translates between Ethernet and internet addresses.

B

broadcast address

The broadcast address is the internet address used for sending data to all hosts on the network.

bus

A network configuration using a conductor (for example, a *coaxial cable*) for transmitting signals.

C

coaxial cable

The coaxial cable is a shielded cable that serves as the transmission medium for Ethernet networks.

collision

Collision means that two hosts are transmitting data onto the *coaxial cable* at the same time. When a collision occurs, the data must be retransmitted.

D

data link layer

The data link layer (layer 2) of the *seven layer model* provides an interface between a specific physical layer medium and a more general network layer protocol.

E

Ethernet

Ethernet is a 10-Mbps local area network system. Ethernet on CTIX computers follows IEEE standard 802.3.

G

gateway

A gateway is a computer that passes data from one network to another. A gateway computer has more than one network interface, and each network interface has an associated network address.

M

multiport transceiver

A multiport transceiver, also known as a multiplexer, is a simple device that does the work of several (typically eight) transceivers.

P

physical layer

The physical layer (layer 1) of the *seven layer model* is the communications hardware itself, which implements the transmission of data bits from one computer to another.

R

repeater

A hardware device that increases the effective broadcast range along a coaxial cable.

S

seven layer model

The seven layer model is a general design model for network systems.

T

transceiver

The transceiver is an electronic device that connects the coaxial cable to each computer. The transceiver detects signals sent from other computers and signal *collisions* that occur when two or more computers try to broadcast at the same time.

transceiver cable
The transceiver cable connects the Ethernet interface card with its *transceiver.*

tunable parameters
The system-wide variables that can be adjusted to improve system performance.

Index

Contents for Part 3

Tables for Part 3

Section 1
Overview

What is SLIP?

The Serial Line Internet Protocol (SLIP) is a set of STREAMS communication modules that allows users to use Internet commands between computers connected with a serial line. You can also use SLIP to access an Ethernet network through a modem or serial line, therefore, allowing you to communicate through the network with systems to which you are not directly connected.

SLIP primarily establishes a connection between two systems on a serial line. It depends on the Internet protocols for what happens once that connection is made. Refer to the Internet section (Part 4) of this guide for more information.

Configuring SLIP

To set up SLIP, do the following:

1. Choose the type of SLIP you need to use. Direct SLIP works only on hard-wired direct lines; switched SLIP works with modems.
2. Create the **/etc/rcopts/KSLIP** rcopt file to configure the SLIP operating system module. (See below.)
3. Continue to Section 2 for information on direct SLIP or Section 3 for information on switched SLIP.
4. Continue to Part 4, the Internet section of this guide to learn how to configure the Internet protocols for your SLIP system.

SLIP Module Configuration

Verify that the file **/etc/rcopts/KSLIP** exists by using the **ls** command. If the file does not exist, use the **touch** command to create it:

```
touch /etc/rcopts/KSLIP
```

The script /etc/**drvload** loads the network modules (drivers) into the operating system. If a **KSLIP** file exists in the **/etc/rcopts** directory, **drvload** loads the **slip** module. The **KSLIP** file should be empty.

Section 2

Direct SLIP

Direct SLIP is easier to administer and has less overhead than switched SLIP when used on a direct line. Direct SLIP supports baud rates up to 38400 baud. Direct SLIP does not support modems of any kind; use switched SLIP for modem lines. All **gettys** and **uugettys** running on ports intended for direct SLIP should be disabled before direct SLIP is initialized.

To connect computers using direct SLIP, you must explicitly initialize each network connection using the program **slattach**(1M). **slattach** sets up the appropriate interface and route descriptions in the operating system. For each serial port that you want to use for SLIP, you should add an **slattach** line to the file /**etc/rcopts/DEVICES** for that port.

You can also use **slattach** and **sldetach**(1M) at any time to create and remove SLIP links. You should determine whether to run these commands from /**etc/rcopts/DEVICES** or from the command line based on how often the connections change. To set up direct SLIP, do the following:

1. Connect the two machines with the appropriate cable attached to their communications ports.
2. Make an **slattach** entry in the /**etc/rcopts/DEVICES** file of both machines. (This automatically reinstates the connection upon reboot.) These lines should come at the beginning of the file. Here is a sample **slattach** entry:

```
/etc/slattach /dev/tty001 sales mktg 9600
```

Analyzing the arguments:

- */dev/tty001* is the communication device for the connection.
- *sales* is the internet name of the local host.[1]
- *mktg* is the internet name of the remote host.
- *9600* is the baud rate of the serial connection.

3. Reboot the systems.
4. If either or both of the machines is a gateway, configure the routing entry by use of the **route** command. See Part 4, the Internet section of this guide for more information on setting up routes.

[1] See Part 4, Section 2 of the Internet section of this guide.

Section 3
Switched SLIP

Switched SLIP can run through a telephone switching system, or in other words, can use modems and ordinary (or special) telephone lines.

The SLIP daemon **slipd** runs in two modes: *master* mode and *slave* mode.

In master mode, **slipd** manages SLIP communication modem connections. It can be started up at boot time, and while it is running, it makes all necessary calls to other hosts. The following steps occur when **slipd** in master mode calls a host:

1. A login connection, using **slipin** as the login shell, is made to the remote system. The master **slipd** and the slave **slipin** begin to communicate.
2. The network protocol is initialized as in direct SLIP. This process is equivalent to running **slattach**.
3. After the network activity is finished, **slipd** detaches the connection. This process is equivalent to running **sldetach** in direct SLIP.

Normally, **uugetty** is run on ports intended for switched SLIP.

A link to the **slipd, /etc/slipin**, services SLIP login requests. When called by the name **slipin, slipd** runs in slave mode.

See Section 4, ''SLIP Monitoring and Troubleshooting,'' for information about monitoring **slipd**. Also see **slipd**(1M) for detailed reference information.

Switched SLIP Setup

To set up switched SLIP:

1. Create an empty file **/etc/rcopts/SLIPD**. This causes **slipd** to be started at boot time.
2. Set up UUCP on the two hosts. This allows you to resolve all serial line and dialing issues before bringing in the SLIP and Internet protocols.
3. Convert from UUCP to SLIP.

See Part 4, Section 9, ''Network Setup Samples,'' in the Internet section of this guide for more information.

UUCP Setup

Study Section 10, ''UUCP,'' in the *CTIX Network Administrator's Guide*; follow the instructions carefully. Set up the serial line with a **uugetty**(1M) on both ends. Find a **Systems** file entry that works reliably. Then proceed to the next steps.

Convert from UUCP to SLIP

To convert from UUCP to SLIP:

1. Add the following line to your **/usr/lib/uucp/Sysfiles** file, if it is not already present:

   ```
   service=slip  systems=Systems.slip
   ```

2. Copy your **/usr/lib/uucp/Systems** file entry to the file **/usr/lib/uucp/Systems.slip**. Create this file if it does not exist.

3. Note the following line in your **/etc/passwd** file:

   ```
   slip:NONE:72:2:SLIP:/usr/lib/slip:/etc/slipin
   ```

   This is the line for the **slip** login.

4. Give the **slip** login a password. The super-user can do this with the following command:

   ```
   passwd slip
   ```

5. Change the login name in your new **/usr/lib/uucp/Systems.slip** entry (probably **nuucp**) to **slip**. The **slip** login is dedicated to SLIP. (See your **/etc/passwd** file.) For example:

   ```
   sales Any ACU 2400 3331112 "" "" \r\d ogin nuucp
   ```

   would become

   ```
   sales Any ACU 2400 3331112 "" "" \r\d ogin slip
   ```

6. Make sure the line

   ```
   LOGNAME=slip
   ```

   appears in the file **/usr/lib/uucp/Permissions**. This line must be in the **/usr/lib/uucp/Permissions** files on both sides of the SLIP connection.

Note: For added security, you can create your own SLIP login using the login **slip** *as a model. A new SLIP login must be configured in* **/etc/passwd**, **/usr/lib/uucp/Permissions**, *and* **/usr/lib/uucp/Systems.slip**.

Section 4
Troubleshooting

This section explains how to monitor and troubleshoot static and switched SLIP. The following tools (described in the Introduction and Internet sections [Part 4] of this guide) can be used with SLIP:

- **netstat**
- **ifconfig**
- **lddrv**
- Any network command, such as **rlogin**.

In addition to these commands, **cu**(1C) and the switched SLIP log file can be used for debugging.

Using cu

The **cu** (call unix) program can be used during the initial setup of switched SLIP or as a debugging tool. Because **cu** uses the serial line hardware and the same **/usr/lib/uucp/Devices** and **/usr/lib/uucp/Systems** file entries that SLIP does, it can be used to validate those parts of the SLIP setup.[1]

[1] The **Systems** file entry for **cu** is normally configured in **/usr/lib/uucp/Systems**, not **/usr/lib/uucp/Systems.slip**.

The command syntax for testing the connection with a computer **central** is shown below:

```
cu central
```

The UUCP system can also test serial hardware, **Devices**, and **Systems** file entries.

The Switched SLIP Log File

slipd writes log information to the file **/etc/log/sliplog**. Each line in the log has the following format:

```
date; pid; role [(device)]; [name (internet addr)]; message
```

1. *date* is the date and time of the action.
2. *pid* is the process number of the **slipd**.
3. *role* is the role **slipd** takes in the action. For example:
   - master (**slipd**, caller)
   - slave (receiver of call, **slipin** program)
   - child (process spawned to make a connection)
4. *device* is the serial line associated with the action.
5. *name* is the Internet name of the remote host.
6. *internet addr* is the Internet address of the remote host.
7. *message* is the log message. For example:
   - *requested: pid 11394* - master requests connection
   - *attach sl0* - connection succeeded
   - *done* - action completed

- *lost line* - carrier dropped
- *conn failed (CALLER SCRIPT FAILED)* - error dialing modem
- *conn failed (LOGIN FAILED)* - error logging in to remote host.

Here is a full example line of the completion of a slave session:

```
Oct 30 13:36:12 EST; 13999; slave (0, 1); central (3.0.0.1); done
```

slipd also provides a debug mode for writing more

```
/etc/slipd -d
```

This is a toggle command: to disable debugging, you type the same command. The extra information from debug mode is detailed status information about connections.

Troubleshooting SLIP

Table 4-1 outlines some typical SLIP problems, along with some possible solutions. The table is arranged as follows:

- *Symptom* is a problem you are experiencing.
- *Additional Tests* are actions you take to help define the cause of the problem.
- *Suggested Actions* are possible solutions to the problem. Some may not be applicable to your particular case, but will correct the symptom in the majority of cases.

The additional tests and suggested actions help you understand the nature of the problem. If the problem cannot be solved, contact your Technical Support representative.

Table 4-1. Problems and Solutions

Symptom	Additional Tests	Suggested Actions
slattach fails (static SLIP)		Check that cable is null modem type. Check **/etc/inittab** on both computers, no **gettys** or **uugettys** should be on the SLIP ports.
slipd fails to connect to remote host		Check cables, modems, and telephone lines. Check **Devices** and **Systems** entries using **cu**. Check SLIP log for error messages. Check Internet software.

Glossary

B

baud rate

Baud rate is the speed at which data are transmitted over certain types of serial channels; for example, RS-232-C. Baud rate is usually measured in bps (bits per second).

D

direct SLIP

Direct SLIP is the type of SLIP that works only on hard-wired direct lines.

G

getty

A getty is a UNIX program that monitors the active serial lines of a system, waiting for a connection from another device or system.

I

internet

Internet is a general term for the protocols, commands, and other software related to the Internet Protocol.

M

modem

A modem is a serial communications device that allows transmission of computer data across regular phone lines.

S

SLIP

Serial Line Internet Protocol.

STREAMS

STREAMS is an enhancement to the UNIX character input/output (I/O) system that supports the development of communication software. STREAMS is new to the AT&T UNIX 5.3 and the CTIX 6.0 releases.

Serial Line Internet Protocol

The Serial Line Internet Protocol is a set of communication programs that allows users to use Internet commands between computers connected with a serial line.

switched SLIP

Switched SLIP is the type of SLIP that can run through a telephone switching system, or in other words, can use modems and telephone lines.

Index

Contents for Part 4

Section 1. Network Setup

Section 2. Network and Host Management

Section 3. Networking Services

Contents for Part 4

Section 4. Internet UUCP

Section 5. Routing

Section 6. Internet Security

Section 7. Equivalent Machines and Users

Section 8. Internet Monitoring and Troubleshooting

Section 9. Network Setup Samples

Glossary

Index

Figures for Part 4

Tables for Part 4

Section 1
Network Setup

The internet software consists of the following parts:

- The Transmission Control Protocol (TCP)
- The Internet Protocol (IP)
- Other needed operating system modules
- Configuration files
- Network services
- Network commands.

This section explains how to set up the underlying software for the network. All the information is based on the **rcopts** scheme, therefore, once your software is configured, all the systems are initialized when your computer boots up. The following topics are discussed:

- Loadable operating system modules
- STREAMS
- Sockets
- Network services.

Loading the System Modules

The script **/etc/drvload** loads the modules (drivers) needed for internetworking into the running operating system. The following **rcopts** are required:

- **/etc/rcopts/KSTRM** causes the STREAMS and Transport Layer Interface (TLI) drivers (**streams, clone, sp, log, timod,** and **tirdwr**) to be loaded.
- **/etc/rcopts/KINET** causes the pseudo-terminal, loopback, and TCP/IP drivers (**pts, llcloop, ip, icmp, udp,** and **tcp**) to be loaded.
- **/etc/rcopts/KSOCK** causes the socket driver to be loaded.

The above **rcopts** should be empty files. They can be created by **root** with the command

```
touch file-name
```

If the file **/etc/rcopts/LOCDRVR** exists, it is executed as a shell script to load local drivers.

STREAMS Startup

The **STREAMS** package is needed for all CTIX networking. The file **/etc/rcopts/KSTRM** must exist for STREAMS to be activated in the operating system at boot time.

/etc/rcopts/KSTRM causes the program **slink**(1) to run. This program links the STREAMS protocols (TCP, IP, etc.,) into the system. It gets configuration information from the file **/etc/netcf** and continues to sleep in the background as a daemon to maintain the linkage.[1]

[1] It is not necessary to modify the **/etc/netcf** file to set up your network.

Sockets Startup

The required file /etc/rcopts/**KSOCK** causes sockets to be initialized. Initialization is done by the **ldsocket** program, which tells the operating system how to associate the socket interface with the STREAMS protocols configured with **slink**. **ldsocket** also reads the /etc/**netcf** file. (See **slink**(1) for more information.)

Services Startup

Network services are started up as follows:

- /etc/**rcopts**/**KINET** starts the inet daemon (**inetd**). **inetd** is a general purpose service that manages the startup of other services. This scheme allows the subsidiary services to run only when they are needed. In earlier releases, numerous services ran at all times "listening" for calls from remote hosts. Most **inetd** services are configured in the file /etc/**inetd.conf**.
- If the file /etc/**rcopts**/**NETD** exists, daemons named in it are started. These daemons are services that need to be outside of the **inetd** scheme.
- /etc/**rcopts**/**UNLS** controls the startup of the Network Listener Service, which is needed for Remote File Sharing (RFS).
- /etc/**rcopts**/**URPC** controls the startup of the portmap daemon, which is needed for the Network File System (NFS).

See Part 4, Section 3, the Internet section in this guide, for more information about network services and their administration.

Tuning the System

The CTIX system provides *tunable parameters* that adjust the amount of resources devoted to certain networking services to optimize system performance.

All parameters have set default values that should work for an average system. As you customize your system, you might find that modifying some parameter values results in increased performance.

To change the value of a tunable parameter:

1. Edit the parameter's value in the **/etc/system** file.
2. Use the **uconf -w** command to update the system memory image; see *uconf*(1M) for details.
3. Reboot the system.

The following paragraphs describe some tunable parameters that can improve the performance of systems by using the Network File System (NFS) or Remote File Sharing (RFS).

Note: *Keep in mind that modifying the value of a tunable parameter does not always enhance system performance. If the value is too small, the system might not provide enough resources to properly handle the load; if the value is too large, your system's resources could be excessively drained. Use the monitoring tools available to check the system load, and tune the system as needed; refer to the Troubleshooting sections throughout this guide for information about monitoring system load.*

Tuning for NFS

Tuning the following three parameters can help to enhance the performance of a network using NFS. (Note that the CTIX system provides several other NFS tunable parameters, available for performance enhancements in specific situations; refer to the *S/Series CTIX Administrator's Guide* for details.)

v_nblk4096 Specifies the number of 4K STREAM buffers for NFS. A good value for most NFS systems is 32. Be careful of increasing the value beyond 32, and do not increase the value beyond 64.

v_nblk2048 Specifies the number of 2K STREAM buffers for NFS. A good value for most NFS systems is 64.

v_nbuf Specifies the number of system buffers for NFS client caching. A suggested value is 150.

Tuning for RFS

Tuning the following three parameters can help to enhance the performance of a network using RFS. Most RFS systems use 2K buffers. (Note that the CTIX system provides several other RFS tunable parameters, available for performance enhancements in specific situations; refer to the *S/Series CTIX Administrator's Guide* for details.)

v_nblk4096 Specifies the number of 4K STREAM buffers for RFS. A good value for most RFS systems is 16.

v_nblk2048 Specifies the number of 2K STREAM buffers for RFS. A good value for most NFS systems is 64.

v_nstream Specifies the number of STREAMS that can be open at once. A suggested value is 69.

What To Do Next

Continue to Section 2 for information about how to configure your hosts and networks.

Section 2
Network and Host Management

This section discusses how to add computers and networks to your configuration files and how to remove them if necessary. The first part of the section explains hosts, networks, names, and addresses. The second part of the section explains how to administer the host and network configuration on each of your computers.

Hosts and Networks

A single computer on a network is called a *host*. Host names (one for each network interface on the computer) are set when the system is booted up. One of these names is considered the primary name.

The primary host name is made up of two dot-separated parts:

- The UUCP node name, set from the file /etc/**rcopts/NODE**
- An Internet domain name, set from the file /etc/**rcopts/INET-DOMAIN**

The Internet domain name is optional. Its purpose is to provide unique names for a large number of hosts on a large number of networks. The Internet domain name is analogous to the path name of a file in a CTIX file system. The domain name specifies a network path originating at the *root* of an established network, for example, the ARPANET/MILNET. (See below.) Unlike a file path name, the root of an Internet domain name is on the right, for example,

MyCompany.COM

refers to all networked hosts in the corporation *MyCompany*. MyCompany is part of the larger group *COM*, the set of commercial networks. COM is connected to the root of the internetwork.

Adding a node name to the above example creates a full host name:

database.MyCompany.COM

If a computer communicates on more than one network, it is a *gateway*. A gateway has more than one host name. The additional name(s) identify the gateway on networks other than its primary network. Each host name is associated with an internet address. Thus, a gateway with three network interfaces communicates on three networks and has three host names and three internet addresses.

An internet *network* is made up of a number of hosts. These computers communicate with each other using TCP/IP. Each host knows about the other hosts on the network through a table or database.

When two or more networks are connected, they form an *internetwork*. Transfer of data between networks is controlled by the Internet Protocol (IP).

If you need to connect your network with the Department of Defense Advanced Research Projects Network (ARPANET) or the Military Network (MILNET), contact the Defense Data Network (DDN) Network Information Center (NIC). The center provides information regarding Internet address and domain name assignments. The address is

Host-Master, Room EJ291
Network Information Center
333 Ravenswood Avenue
Menlo Park, CA 94025
(800) 235-3155

If you do not plan to be on the ARPANET/MILNET, you may assign your own arbitrary names and addresses. (See below.)

You must modify two configuration files to add and remove hosts:

- **/etc/hosts** (the system hosts table)
- **/etc/networks** (the system networks table).

Internet Addresses

The internet address uniquely identifies the location of a host. The internet address is a single 32-bit integer. In the simplest case, it can be divided into a network number and a host number. The internet address also contains an address class identifier and an optional subnet number.

The **/etc/hosts** file accepts four decimal representations of the internet address. These address forms are made up of one, two, three, or four digits separated by dots:

- *A* implies one 32-bit address value.
- *A.B* implies division into an 8-bit part and a 24-bit part.
- *A.B.C* implies division into two 8-bit parts and a 16-bit part.
- *A.B.C.D* implies division into four 8-bit parts.

To construct simple addresses for your **/etc/hosts** file, use an A.B format. Let A be the network number, and B contain the host number.

1. Note that CTIX software is distributed with one network preconfigured in the **/etc/networks** file, network 3.
2. Assign host numbers to your computers. Start with host number 1, and continue with 2, 3 and so on.
3. Put the addresses in A.B (*net.host*) format. Your first address is 3.1. The first host entry is

```
3.1   mysystem   # First host
```

Note that all text after the # is ignored by the subroutines reading the file.

4. Note that your second host has the address 3.2, and so on. If more than one network is connected by a gateway, you can add another network to your **/etc/networks** file. For example:
   - Edit the **/etc/networks** file.
   - Add the line

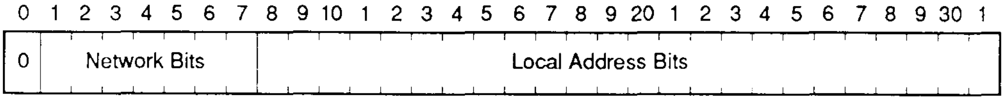
     ```
     Second-Net   4   # Second Ethernet Network
     ```
5. Note that Second-Net is the name of the network, 4 is the network number, and everything after the # is ignored. The first host on network 4 has address 4.1.

Address Classes

If your networks are not likely to be connected to the ARPANET/MILNET, you do not need to be concerned about address classes. Remember that ARPANET/MILNET addresses are assigned by the Network Information Center (NIC).

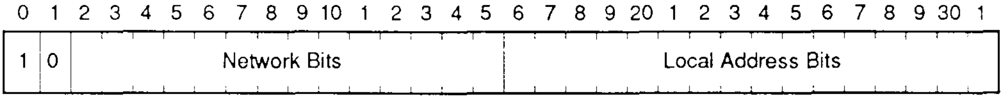
An internet address is made up of an address class identifier, a network number, and a local host address number. The address class identifier is 0, 10, or 110 (binary) for class A, B, or C, respectively. The network number is assigned to a physical network in the Internet, for example, a computer network at a corporation. The local address carries information to address a host in the network identified by the network number.

The internet address is a 32-bit quantity formatted differently in three types, or classes, to accommodate different network size configurations. Class A allocates a 7-bit network number and a 24-bit local address. Class B allocates a 14-bit network number and a 16-bit local address. Class C allocates a 21-bit network number and an 8-bit local address. Figures 2-1, 2-2, and 2-3 give the formats of the address types.



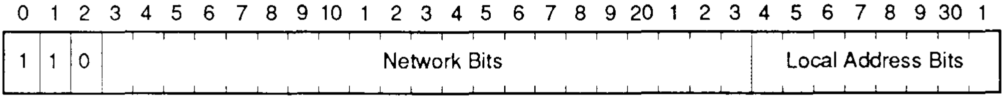
2320.4.2-1

Figure 2-1. Class A Address



2320.4.2-2

Figure 2-2. Class B Address



2320.4.2-3

Figure 2-3. Class C Address

This system provides unique addresses for the statistical distribution that might be expected in the population of networks using this address system. There is a smaller number of large networks, having many hosts (class A), a larger number of small networks consisting of a lesser number of hosts (class C), and a medium number of networks made up of a medium number of hosts (class B).

Because each network has a particular address format and length (class A, B, or C), the IP maps between the internet local addresses and the actual address format used in the particular network. Ethernet uses the Address Resolution Protocol (see RFC 826).

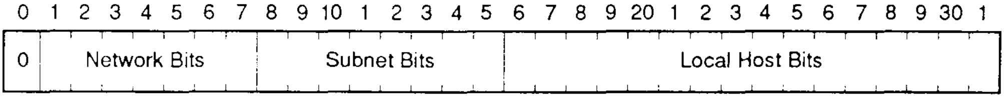
Subnets

Note: *The following section applies only to large networks with numbers assigned by the NIC, and is intended for experienced network administrators with significant knowledge of internet addressing.*

If an organization has only one network number (assigned by the NIC) for a large number of hosts, the administrator can set up *subnetting*. Subnetting means that each host's address has three significant parts: the network number, a subnet number, and the host number. (Note that the subnet number is known only to hosts on the local network. To other networks, the subnet number is part of the host number.)

To communicate with a subnetted network, the configuration of the interface communicates with that network by using the **ifconfig**(1M) command. The interface parameter that is changed is called the *netmask*. The netmask is used to derive the subnet number from the 32-bit internet address. The netmask is a 32-bit number which, when bitwise-ANDed with the internet address, gives both the network and the subnet numbers.

For example, in Figure 2-1, the Class A address has no subnetting. Its netmask is FF000000 hex. This value masks off only the network number. In Figure 2-4, the netmask is FFFF0000 hex. This allows 8 bits for the subnet number, and 8 bits for the network number.



2320.4.2-4

Figure 2-4. Class Address with Subnetting

To change the netmask for your system, use the **ifconfig**(1M) command. For example,

```
ifconfig en0 netmask 0xffff0000
```

configures subnetting for your **en0** Ethernet interface. This **ifconfig** command can be added to the file /**etc/rcopts/ROUTING**. It is then run automatically when the system boots.

The /**etc/hosts** file must also be configured for subnetting. Two logical formats for a subnetted address are the A.B.C and A.B.C.D forms discussed earlier.

In the A.B.C format, A can be the network number, B can be the subnet number, and C can be the local host number. For example, the following would be a **hosts** entry with subnetting (class A with a netmask of FFFF0000 hex):

```
3.2.1   mysystem   mysystem.MyCompany.COM   # First host
```

In this case, the network number is three, the subnet number is two, and the local host number is one.

Host Administration

Note: *To add, change, or delete hosts, you must log in as root or the superuser.*

To add a host to your **/etc/hosts** file, follow the administration tools procedure shown in Figure 2-5.

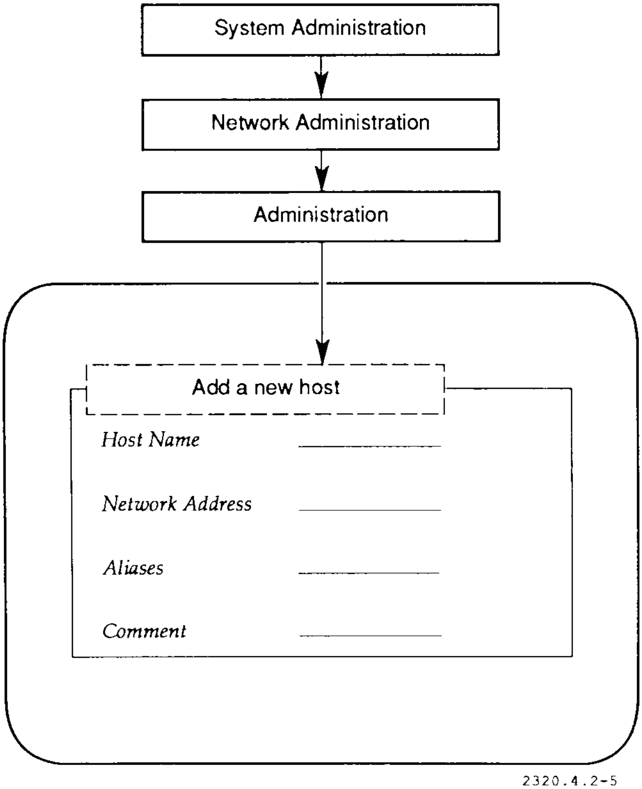


Figure 2-5. Adding Host

- Fill in the *Host Name* and *Network Address* fields; the others are optional.
- This procedure is equivalent to editing the /etc/**hosts** file and adding a line of the form

  Address Name Aliases # Comment

  To change or delete a host, follow the administration tools procedure shown in Figure 2-6.

  - Select the host you wish to change (or delete) from the list given.
  - Enter the needed information when you are prompted.

  This procedure is equivalent to editing the /etc/**hosts** file and changing or deleting a line.

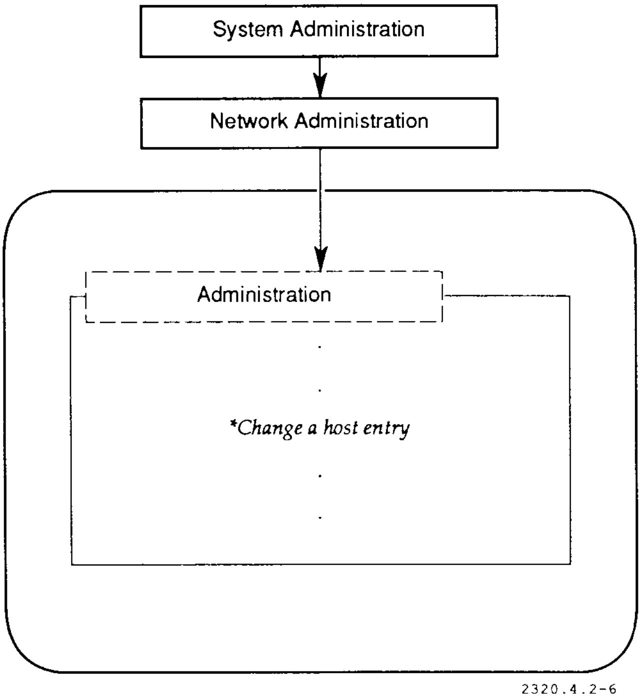


Figure 2-6. Changing or Deleting Host

Adding a Network

To add a new network to your **/etc/networks** file, you need to:

1. Find out the number and name of the new network.
2. Edit the **/etc/networks** file.

When you add a host that resides on a network with which you have never communicated, you need to add its network to your **/etc/networks** file. The best way to do so is to either get the entry from an **/etc/networks** file on a central machine on your network, or from the host you are adding to your system.

The network number can be derived from the internet address of the new host. It is the first component of that address. For example, if you add a host with internet address 30.2, the network number for that host is 30.

Once you have the information you need; edit the **/etc/networks** file. Add a line with the new information, using surrounding lines as a guide, for example:

```
Mark-Net  10  #  Marketing
```

If you add a network to your **networks** file, you will also need to add a route if you are using static routing. See Part 4, Section 5, the Internet section of this guide, for more information.

Managing Configuration Files

To keep the network running properly, the **/etc/hosts** files on all computers must be consistent. To implement this consistency:

1. Designate a computer as the central repository for configuration files.
2. When you need to change the **hosts** file, change it on the central computer.

3. Periodically copy the **hosts** file from the central computer to the other computers using **ftp** or **rcp**.

If your network is connected to other networks, the **/etc/networks** file needs to be managed in the same way.

Section 3
Networking Services

Networking services are programs that a host provides for use by other hosts. This section explains

- The types of services available with CTIX
- **inetd**, a network services manager
- Network service administration.

Service Types and Descriptions

CTIX employs four types of services:

- AT&T System V.3 services
- 4.3BSD (Berkeley) services
- Internet services implementing ARPANET/MILNET protocols
- Sun Network Services. These services are based on the Remote Procedure Call (RPC) interface. See **rpcinfo**(1M) and **rpc**(4) for more information.

Table 3-1 describes the CTIX networking services.[1] Refer to the *CTIX Operating System Manual* for more information about the services. The first column of Table 3-1 contains references to relevant sections of that manual.

[1] **named** and **gated** are not supported in the CTIX TCP/IP 3.2 release.

Table 3-1. CTIX Networking Services

Service	Command	Description	Source
fingerd(1M)	**finger**	User information service	Berkeley
ftpd(1M)	**ftp**	File Transfer Protocol	Internet
gated		Exterior Gateway Protocol	Internet
inetd(1M)		"super-server"	Berkeley
listen		connection service	AT&T
named(1M)		Domain name service	Internet
nserve		Name server	AT&T
portmap(1M)		Port Mapper	Sun
rexecd(1M)		Remote execution service	Berkeley
rlogind(1M)	**rlogin**	Remote login service	Berkeley
routed(1M)		Routing service	Berkeley
rshd(1M)	**rcmd**	Remote shell service	Berkeley
rwhod(1M)	**rwho**	Host status service	Berkeley
sendmail(1M)		Mail Protocol (SMTP)	Internet
talkd(1M)	**talk**	User communication service	Berkeley
telnetd(1M)	**telnet**	**telnet** login service	Internet
tftpd(1M)	**tftp**	Trivial file transfer	Internet
uucpd(1M)	**uucp**	Network **uucp** services	Berkeley

Internet services are assigned logical numbers, called *ports*, to use for communication with other hosts. These ports simulate real connections, making a single network look as if it were made up of many individual communication lines. Internet services are assigned fixed port numbers. (See **services**[4].)

inetd

inetd is a general purpose service that manages the startup of other services. **inetd** services are configured in the file **/etc/inetd.conf**. Here is a sample **inetd.conf** entry:

```
telnet  stream  tcp  nowait  root  /etc/telnetd  telnetd
```

Analyzing the fields:

- *telnet* is the service name from **/etc/services**.
- *stream* is the socket type of the service.
- *tcp* is the protocol name from **/etc/protocols**.
- *nowait* is a socket parameter.
- *root* is the user ID of the process (telnetd is the process in this case).
- */etc/telnetd* is the path name of the service program.
- *telnetd* is the argument list for the service program.

See **inetd.conf**(4) for more information.

Note that **rwhod** and **ouucpd** should not be started with **inetd**. They should be listed in the file **/etc/rcopts/NETD**.

Network Service Administration

Note: *To change your networking services, you must be logged in as root or the super-user.*

To add a network service, follow the administration tools procedure shown in Figure 3-1.

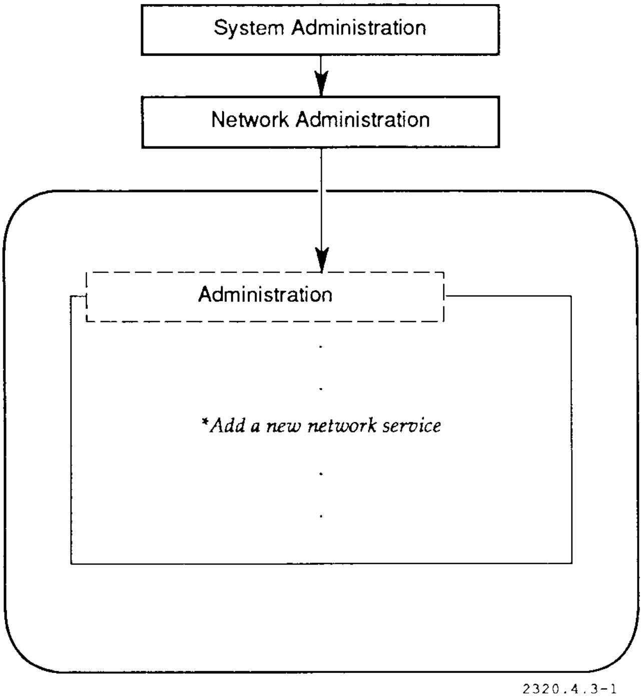


Figure 3-1. Adding a Network Service

Choose a service from the list of available services.

Services are configured in the files **/etc/inetd.conf** (for **inetd** services) and **/etc/rcopts/NETD**. The above procedure is equivalent to adding a line to one of those files.

To delete a network service, follow the administration tools procedure shown in Figure 3-2.

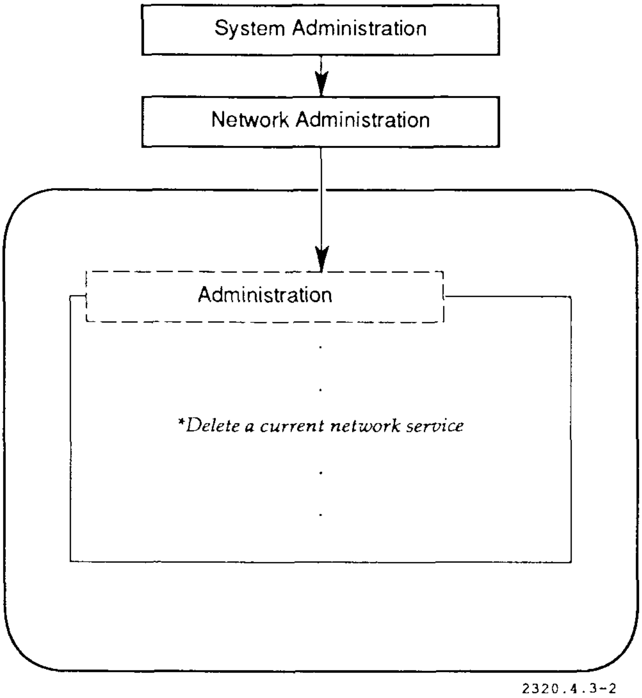


Figure 3-2. Deleting a Network Service

Select the service you wish to delete from the list given. As mentioned earlier, services are configured in the files **/etc/inetd.conf** (for **inetd** services) and **/etc/rcopts/NETD**. The above procedure is equivalent to deleting a line from one of those files.

Section 4
Internet UUCP

UUCP is a batch-oriented protocol used to transfer files across telephone lines, direct lines, and networks. CTIX UUCP can run on top of TCP/IP and therefore can run on CTIX Ethernet or SLIP. S/Series UUCP is compatible with 4.3BSD and AT&T 3B2 releases.

Note: *See the release notices for the machine-appropriate operating system and TCP/IP distribution for operating system compatibility and other special requirements for UUCP being used with TCP/IP.*

UUCP is used by electronic mail and other application systems. It can also be used directly from the shell. Once the setup described below is performed, UUCP uses the internetwork communications transparently to the user processes.

UUCP Routing

UUCP routes consist of multiple hops made up of physical and/or logical telephone lines. You must specify these routes in their entirety, because UUCP itself provides no routing protocol. When UUCP runs on top of TCP/IP, TCP/IP routing is transparent to UUCP.

For example, host *one* is connected to host *two* via a telephone line. Host *two* is connected to host *three* via Ethernet. Host *three* is connected to host *four* via SLIP. All four hosts have UUCP configured. To send a file from host *one* to host *four*, use the command

```
uucp Myfile two!three!four!~
```

Setting Up UUCP For Internetworking

UUCP requires an entry in the file **/usr/lib/uucp/Systems** for each site with which you wish to communicate. **uucico**(1M) uses this information to make connections. UUCP also requires an entry in the file **/usr/lib/uucp/Devices** for the TCP/IP "device." Both file formats are explained below.

To set up UUCP to run on top of TCP/IP:

1. Set up UUCP as directed in the appropriate release notice.
2. Use the administration tools to start up the UUCP service (**uucpd**[1M]).
3. According to the specifications outlined below, add needed entries to **/usr/lib/uucp/Systems** and **/usr/lib/uucp/Devices**.

To keep UUCP running properly, the **Systems** files on all computers must be up to date. To manage this, use the central repository computer described at the end of Section 2 in Part 4, the Internet section of this guide. Change the **Systems** file on that computer when necessary and copy it to the other hosts on your network(s) using **ftp** or **rcp**.

Systems File Formats

A 4.3BSD file format and an AT&T UNIX file format are provided below. Each type of line entry allows connection to both types of systems. The 4.3BSD format may be necessary in some situations if the host name of the remote system differs from its UUCP node name. The AT&T format is compatible with AT&T user interface software.

The 4.3BSD **Systems** file format is

```
Sname  TimetoCall  DeviceCode  Port  HostName  LoginProto
```

The AT&T **Systems** file format is

```
Sname TimetoCall DeviceCode unused unused LoginProto
```

where

- *SName* is the name of the site (node) you wish to call, as identified by UUCP. The node name must be unique within the file **/usr/lib/uucp/Systems**. Note that the internet address used by TCP/IP to contact the site is derived from the HostName field, if it is present.
- *TimetoCall* is the time you want to place calls. ''Any'' allows your computer to call other hosts at any time. For a complete discussion of this parameter, see Section 10, ''UUCP,'' in the *S/Series CTIX Administrator's Guide.*
- *DeviceCode* for BSD format is UCBTCP, and for AT&T format is TCP.
- *unused* means an unused field. There are two unused fields in the TCP style line. You can insert any string, but the word ''unused'' is recommended.
- *Port* should almost always be **uucp**. UUCP looks up the word in **/etc/services**. (AT&T entries require a port entry in the **Devices** file.)
- *HostName* is the host name from **/etc/hosts**. This entry determines the actual address of the remote host. AT&T entries do not require a network name. The AT&T scheme looks up *Sname* in **/etc/hosts**.
- *LoginProto* is a simplified expect-send sequence for network logins. There is no need to wait for any return strings, although login and password prompts are sent.

Here is a sample entry using the 4.3BSD scheme:

```
rochester Any UCBTCP uucp ur-seneca login nuucp sswd FiXeS
```

In this example, rochester is a site known to UUCP but known to the Internet as ur-seneca. Rochester also happens to be the name of a different Internet host. The computer can make calls at any time. The login protocol logs in as *nuucp*, with password *FiXeS*.

Devices File Formats

You must add a line in the file **/usr/lib/uucp/Devices** for each TCP/IP device. If you have TCP lines in your **Systems** file, use

```
TCP unused unused Any TCP uucp
```

in your **Devices** file.

- *TCP* is the device name.
- *unused* means an unused field.
- *Any* is the call times.
- *uucp* is the port name.

If you have UCBTCP lines in your **Systems** file, use

```
UCBTCP unused unused Any UCBTCP
```

in your **Devices** file.

- *UCBTCP* is the device name.
- *unused* means an unused field.
- *Any* is the call times.

For more information on UUCP, see **uucp**(1) and related entries in the *CTIX Operating System Manual* and Section 10 in the *S/Series CTIX Administrator's Guide.*

Sites with Earlier Release Levels

Some sites with which you wish to communicate may not have updated their release version of UUCP. This situation may be indicated because the above **Systems** file entry does not work in their cases. You can still communicate with them if you make **Systems** file entries in the following earlier format:

SName TimetoCall DeviceCode BaudRate LoginAcct

where

1. *SName* is the same as the current format above.
2. *TimetoCall* is the same as the current format above.
3. *DeviceCode* is INET.
4. *BaudRate* is always 9600.
5. *LoginAcct* is **nuucp**.

Note that in this scheme:

- A **Devices** file entry is not needed.
- A **.rhosts** file should be created on the remote host in the home directory of *nuucp* (usually **/usr/spool/uucppublic**) to allow the *nuucp* user on the local system login privileges. See Section 7 of Part 4 of the Internet section of this guide for more information.

If you need to communicate with an ''old site'' for which you have made an entry in the above format, you must start **ouucpd**. (See **uucpd**[1M]). Use the administration tools as described in Section 3, ''Networking Services,'' in Part 4 of the Internet section of this guide.

Section 5
Routing

Routing is the process by which data are directed from one host to another. If the hosts are on different networks, data are transmitted step by step through gateways to the final destination.

A *route* is a CTIX data structure that contains information about how to direct data from one host to another. All routes on a given host (taken collectively) make up its *routing tables.*

There are two types of routes, *host routes* and *network routes.* A host route specifies the gateway to use from the local host to another host (on the local network or on another network). A network route specifies an intermediate host through which traffic from the local network can be routed on the way to destinations in another network. Each network route specifies the route to one network.

The network protocol prioritizes host routes over network routes. When translating an address, the protocol checks first for a host route entry and will use a network route only if a host route cannot be found.

Routes are administered in two ways, either by dynamic routing or static routing.

Dynamic Routes

Dynamic routing means that routes are automatically initialized and updated by the **routed**(1M) service. **routed** interprets data received from other hosts and converts these data into usable routing table entries. It also sends routing information to other hosts. **routed** is run by default on S/Series CTIX systems unless the file **/etc/rcopts/ROUTING** exists.

Two special **routed** features should be used when routed is run on a gateway:

- The **-g** option, which causes **routed** to advertise the gateway as a default route. (See below.)
- The /etc/**gateways** file, that contains routes for **routed** to add to the system routing tables. You should use this facility to initialize routes for networks that are several gateway hops from the local network. It is not necessary to put all nearby gateways in the **gateways** file, because **routed** will be informed of these automatically.

Default Route

The default route is the routing table entry that is used when there is no other route to a destination. Internally, the default route is represented as the gateway to network 0.

Using a local gateway as the default route simplifies routing for non-gateway hosts. These hosts send messages destined for distant networks to the local gateway, which forwards them to an appropriate remote gateway. When this scheme is used, only gateways need to keep track of detailed routing information.

When a gateway starts **routed** with the **-g** option, **routed** broadcasts a request interpreted by each non-gateway host. The request asks that the hosts initialize their default route entries to contain the requesting gateway.

To configure a gateway as a default route advertiser, override the usual **routed** startup by adding this line to the file **/etc/rcopts/ROUTING**:

```
/etc/routed -g
```

/etc/gateways File

Routes configured in the **/etc/gateways** file are added to the system routing tables when **routed** is started up. **/etc/gateways** routes can be *active*, *passive*, or *external*.

Active routes are associated with gateways that have **routed** running. Status information is expected from these gateways. If your system does not receive status information from an active gateway for a certain period of time, **routed** removes the entry from the routing tables.

Passive routes (gateways) are computers where **routed** is not running. They are not removed from the routing tables as active routes can be. Passive routes are not transmitted to other hosts in **routed** status messages.

An external route is a route that should not be added by **routed** because another type of routing service will add it.[1] Like passive routes, external routes are not transmitted to other hosts in **routed** status messages. Note that an external route is only necessary when both **routed** and the other routing service can learn about the same destination.

Here is the form for an **/etc/gateways** line:

<net|host> *name1* **gateway** *name2* **metric** *value* **<passive|active|external>**

Fields are as follows:

- *<net/host>* is a keyword. *Net* means that the destination is a network; *host* means that the destination is a host.
- *name1* is the destination name (or Internet address) from the **/etc/hosts** file or the **/etc/networks** file.

[1] CTIX TCP/IP 3.0 does not support alternative routing services.

- *name2* is the gateway name.
- *value* is the *metric* for the route, or the number of gateway hops needed to reach the destination.
- *<passive|active|external>* is the route type.

For example:

net Four_Net gateway Three_Net metric 2 passive

means that to send data to Four_Net, data are routed through the gateway Three_Net. There are two hops to get to Four_Net, and the route is a passive one (**routed** may not be running on Three_Net).

Configuring Static Routes

The routing tables can also be manipulated directly or at boot time using the **route**(1M) program.

The **route** program takes two commands: **add** a route or **delete** a route. There is also an option, **-f**, that flushes the routing tables of all gateway entries.

To flush the routing tables, use

```
route -f
```

To add a route, use

```
route add DEST Gateway N
```

- *DEST* is the name or network number of the network to which you are connecting.
- *Gateway* is the name or internet address of the gateway machine.
- *N* is the metric, or number of gateways between your network and the network to which you are connecting. N must be at least one. You must use the metric option of the **route** command when creating routes through gateways.

To delete a route, use

```
route delete DEST Gateway
```

To set up routes at boot time, add the **route** command lines to the **/etc/rcopts/ROUTING** file after the **slattach** commands (if any).

Routing Information

To view the routing tables currently being used by your system, follow the administration tools procedure shown in Figure 5-1.

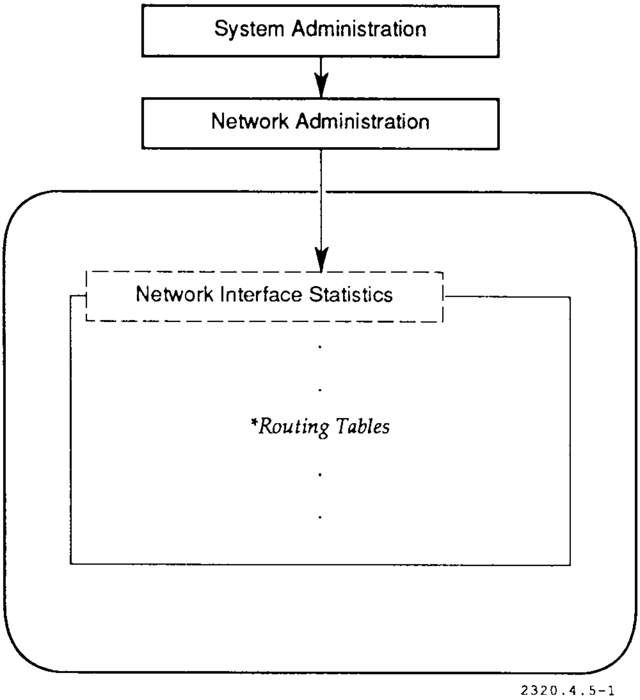


Figure 5-1. Viewing the Routing Table

This procedure is equivalent to running the **netstat** command with the **-r** option. Here is a sample output from this menu item:

```
Routing tables
Destination     Gateway        Flags    Refcnt Use      Interface
loopback        loopback       UH       0      0        lo0
Six-Net         Net-Gateway    UG       0      0        en0
NineNet         Net-Gateway    UG       1      1815     en0
Three-Net       central        U        11     11191    en0
Four_Net        Net-Gateway    UG       0      130      en0
Five_Net        Net-Gateway    UG       0      40579    en0
```

Analyzing this output:

- *Destination* is the network or host to which data are routed.
- *Gateway* is the gateway through which the data are routed to reach the destination.
- *Flags* gives the state of the route:
  - — U means the route is up.
  - — H means the route is a host.
  - — G means the route is a gateway.
  - — S implies a switched SLIP route.
  - — T means a route pointing across a switched SLIP link.
  - — V means a route installed by SLIP in *slave mode*.[2]

[2] See Section 4 in Part 3, the SLIP section of this guide, for more information.

- *Refcnt* gives the current number of active uses of the route.
- *Use* is the number of packets sent using the route.
- *Interface* is the network interface used for the route.

Section 6
Internet Security

System Security

It is important to keep your network secure, because a system with poor security can jeopardize the entire network. As mentioned earlier, physical security is important, as well as password protection on all user accounts.

Do not change the permissions on your system files to make them more accessible. This can seriously compromise your system security.

Security Configuration Files

ftpusers

The file /etc/**ftpusers** contains a list of users who are not allowed to log in using **ftp**. The list usually contains user names that have no passwords, but that would ordinarily not present a security problem (for example, nuucp).

.rhosts

The **.rhosts** files are configuration files for *user equivalence*. User equivalence means that a user on a remote system is given the privileges of a user on the local system. The **.rhosts** file contains a list of users who can access your system without password validation using the Berkeley ''r'' commands (**rlogin**, **rcmd**, and so on).

Each user with an entry in the /etc/**passwd** file can create a **.rhosts** file. The file must reside in the user's home directory. Note that user names not associated with a person can have **.rhosts** files. For example, the user **nuucp** can have a **.rhosts** in the directory **/usr/spool/uucppublic**. To monitor the **.rhosts** files on your system, locate them with this command:

```
find / -name .rhosts -print
```

To configure **.rhosts** files, read Section 7, ''Equivalent Machines and Users.'' For a sample **.rhosts** configuration, see Section 9, ''Network Setup Samples.''

hosts.equiv

The file **/etc/hosts.equiv** contains a list of computers that are *machine equivalent* to your system. This means that a user on a foreign system with the same name as a user on your system can log in to your system without password validation. This allows users with logins on more than one machine to have easy access to his/her logins. An exception is the **root** user, who is not given access for security reasons.

Because the **hosts.equiv** file allows users on remote systems to use your system without password validation, you should make sure that only the super-user can write to the file. Occasionally check your system to verify it has the right permissions and entries.

To configure your **hosts.equiv** file, read Section 7, ''Equivalent Machines and Users.'' For a sample **hosts.equiv** configuration, see Section 9, ''Network Setup Samples.''

Services Security

To configure a secure system, you must be aware of how different services allow users to communicate with your system. Services fall into several different categories:

- UUCP daemons (**uucpd, ouucpd**) enforce security using UUCP conventions. See the UUCP section of the *S/Series CTIX Administrator's Guide* for details.

- The Berkeley r-daemons (**rexecd, rlogind, rshd**) enforce password validation, except when the **.rhosts** files or the **hosts.equiv** file overrides this validation.
- The **telnet** daemon enforces password validation.
- The **ftp** daemon enforces password validation, but without actually starting up a login shell. This means that a UUCP login with no password might serve as a security hole for **ftp**. To prevent this problem, add logins with nonstandard shells to the **/etc/ftpusers** file.
- The **tftp** daemon does not enforce password validation. Thus, a user on one system can read publicly readable files on another system using **tftp** without password checking. There are strict limits as to what tftp can do: it cannot destroy data, but some administrators may not like this read-only security hole.

If you do not like the security a service provides, you can choose to omit it on your system. See Section 3, ''Networking Services,'' to learn how to delete network services.

Section 7
Equivalent Machines and Users

This section explains how to administer permissions for the Berkeley ''r'' commands (**rlogin, rcmd**, and so on). There are two ways of allowing permission:

- Machine equivalence, configured in the file **/etc/hosts.equiv**.
- User equivalence, configured in each user's $HOME/**.rhosts** file.

Section 6, ''Internet Security,'' explains these schemes. Section 9, ''Network Setup Samples,'' provides samples of user and machine equivalence.

Administering Equivalent Machines

Equivalent machines are configured in the file **/etc/hosts.equiv**. You can modify the configuration using a text editor or the administration tools.

> **Caution**
>
> It is important to system security to check your list of equivalent machines (**/etc/hosts.equiv** file) periodically. Delete all entries for machines that should not have equivalence privileges.

To add an equivalent machine, follow the administration tools procedure shown in Figure 7-1.

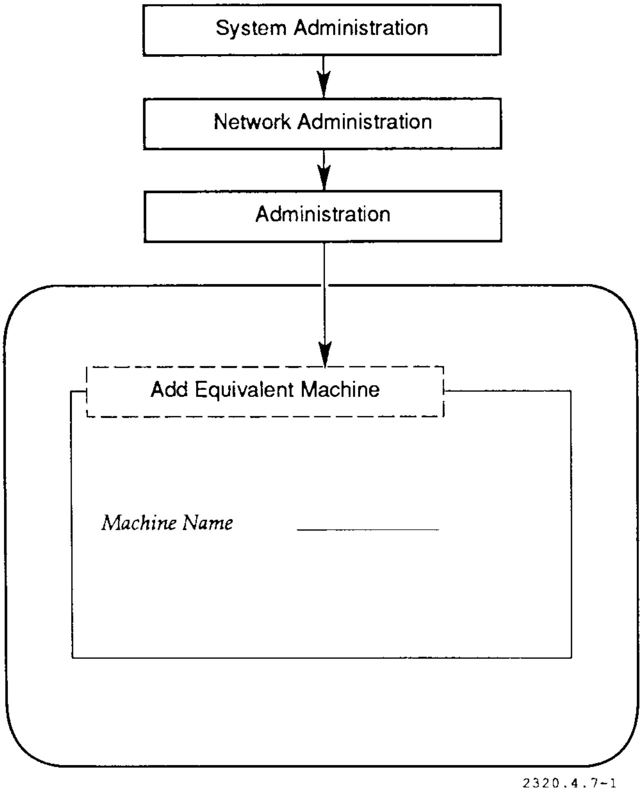


Figure 7-1. Adding an Equivalent Machine

To delete an equivalent machine, follow the administration tools procedure shown in Figure 7-2.

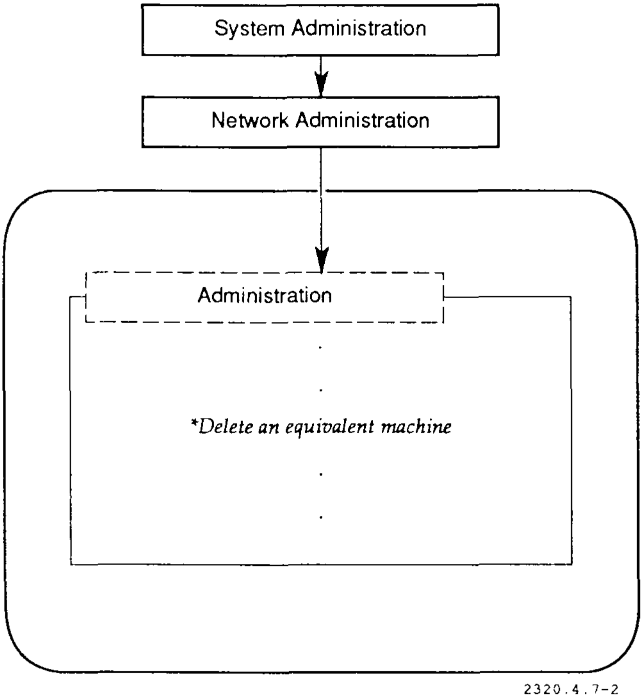


Figure 7-2. Deleting an Equivalent Machine

Administering Equivalent Users

To add an equivalent user, follow the administration tools procedure shown in Figure 7-3.[1]

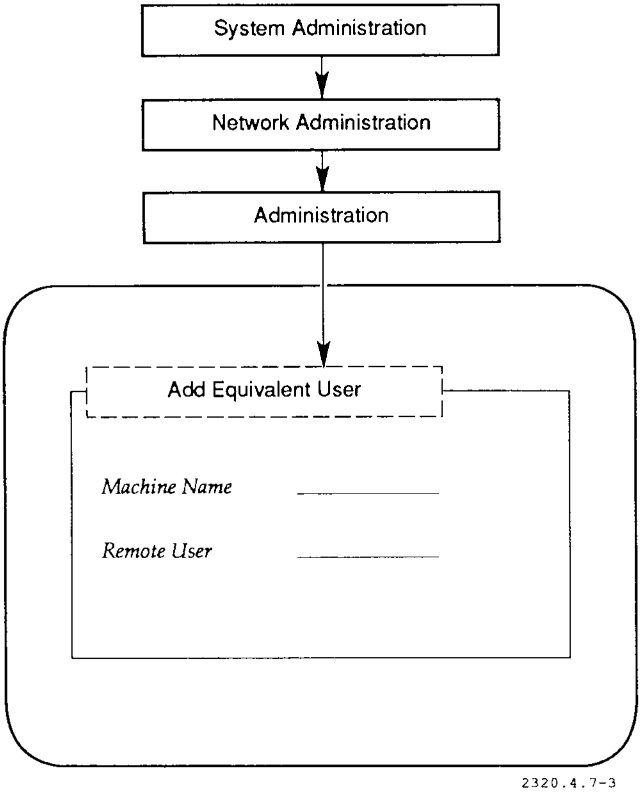


Figure 7-3. Adding an Equivalent User

[1] This procedure can be used by any login, not just the super-user.

Caution

It is important to system security to remind your network users to check their **.rhosts** files periodically and delete any users who should not have equivalence privileges.

To delete an equivalent user, follow the administration tools procedure shown in Figure 7-4.

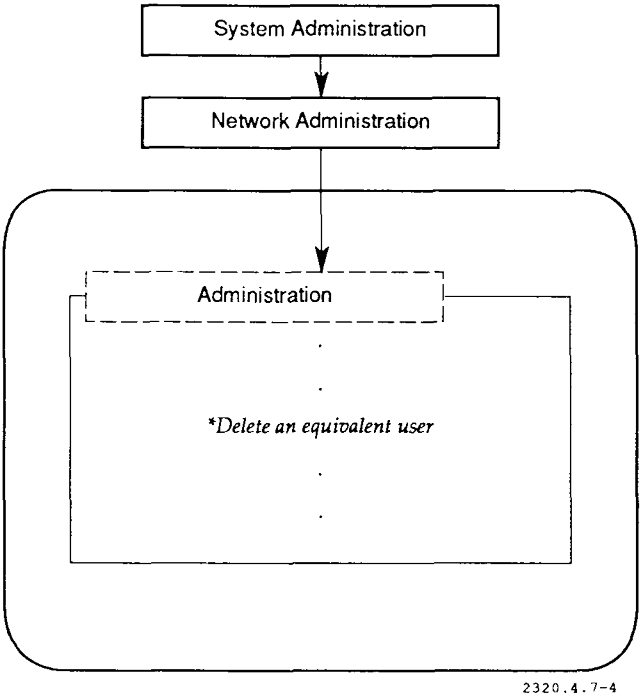


Figure 7-4. Deleting an Equivalent User

When the DELETE option is selected, a list of users will be displayed on the screen.

Section 8
Internet Monitoring and Troubleshooting

This section explains how to monitor and troubleshoot the Internet layers of your network. For more information, see Section 6, ''Network Monitoring and Troubleshooting,'' in the ''About This Guide'' section in the front of this manual.

Internet Functions

The Internet layers of your network perform the following functions:

- Routing. (See Section 5).
- Host status monitoring.
- User status monitoring.
- Support for network services.
- Protocol status monitoring.

You can monitor each function that uses CTIX networking commands. These commands can be invoked through the administration tools or the shell.

Checking Host Status

To check host status on your network, use the **ruptime**(1) command. This can be done either through the shell or with the administration tools.

To view the status of hosts on your network using the administration tools, follow the procedure shown in Figure 8-1.

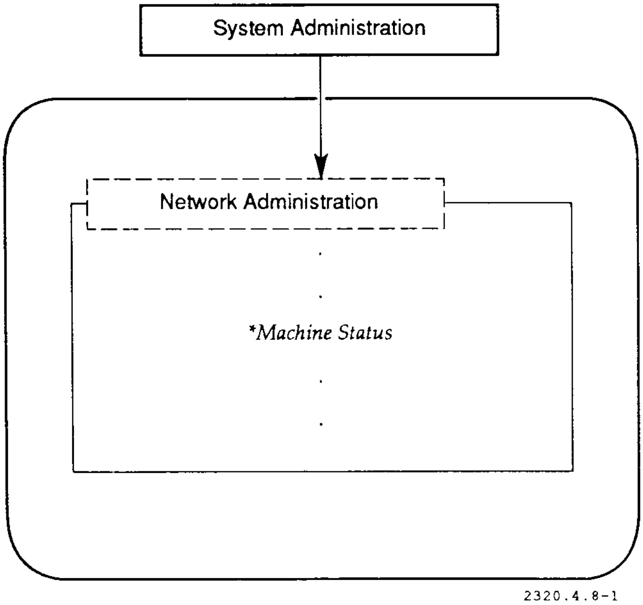


Figure 8-1. Viewing the Status of Hosts

This procedure is equivalent to invoking the **ruptime** command. Here is a sample output from this menu item:

```
host1         up 44+06:04,     9 users,   load 0.06,  0.06,  0.07
host2         up 23+00:06,     0 users,   load 2.00,  2.00,  2.00
host3         up  4+21:33,     1 user,    load 0.00,  0.00,  0.00
```

Analyzing this output for the first line:

1. *host1* is the computer name.
2. *up 44+06:04* means that the computer has been up on the network for 44 days, 6 hours, and 4 minutes. A remote host is considered up when the local status service (**rwhod**[1M]) has received status information from the remote host within the last 5 minutes. The other value is *down*.
3. *9 users* means nine host1 users have been active in the past hour.
4. *load 0.06, 0.06, and 0.07* describes the load on the system (based on the average number of jobs in the run queue) for the last minute, 5 minutes, and 15 minutes.

This output applies only to hosts on your local network. Status is not broadcast to and from other networks through gateways.

Checking User Status

To view users currently on the network, follow the administration tools procedure shown in Figure 8-2.

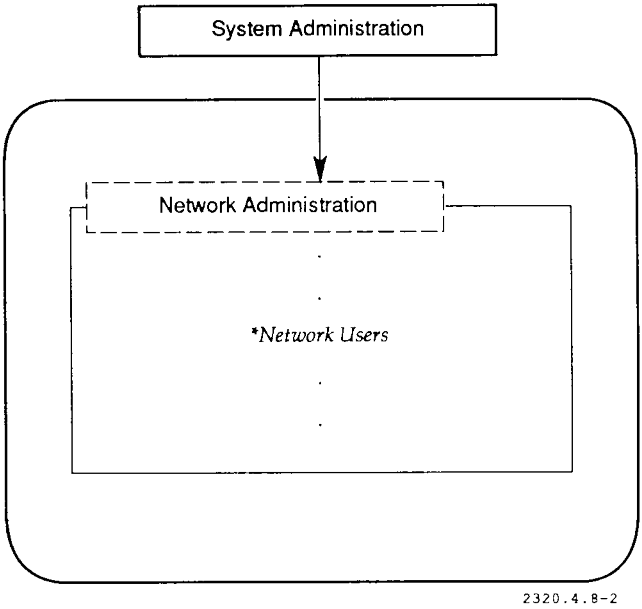


Figure 8-2. Viewing Users Currently on Network

This procedure is equivalent to invoking the **rwho**(1) command. The output also applies only to users on your local network. Here is a sample output from this menu item:

```
root      central:tty280   Oct 30 14:47 :10
john      admin:ttyp00     Oct 30 09:47 :03
mary      finance:tty020   Oct 30 14:37 :10
```

The first field is the user name. The second field is the host name and the device to which the user is logged in. The rest of the fields are the date the user logged on and how long the user has been idle.

Checking Services Status

To view the computer's current network connections (based on the service being used), follow the administration tools procedure shown in Figure 8-3.

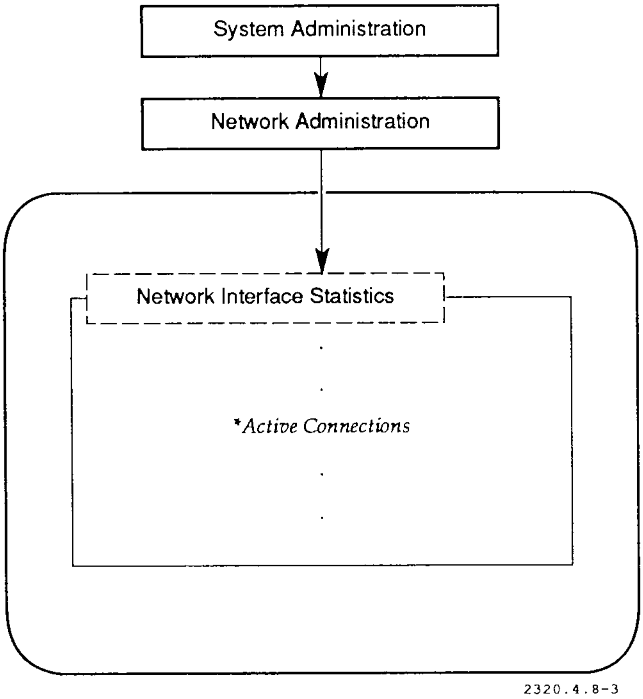


Figure 8-3. Viewing Computer's Current Network Connections

This procedure is equivalent to using the **netstat**(1) command with the **-a** option. Here is a sample output from this menu item:

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address   Foreign Address   (state)
tcp        0      0  central.login   admin.1023        ESTABLISHED
tcp        0      0  *.uucp          *.*               LISTEN
tcp        0      0  *.telnet        *.*               LISTEN
tcp        0      0  *.shell         *.*               LISTEN
tcp        0      0  *.login         *.*               LISTEN
tcp        0      0  *.exec          *.*               LISTEN
tcp        0      0  *.olduucp       *.*               LISTEN
tcp        0      0  *.ftp           *.*               LISTEN
udp        0      0  *.who           *.*
udp        0      0  *.tftp          *.*
udp        0      0  *.*             *.*
```

Analyzing this output:

1. *Proto* is the protocol name.
2. *Recv-Q* is the size of the receive queue (in bytes).
3. *Send-Q* is the size of the send queue (in bytes).
4. *Local Address* is the address of the local system including the service name. (* is a wild card that represents any internet address or any service.)
5. *Foreign Address* is the address of the remote system.
6. *(state)* is the internal state of the protocol.

The first line means that a user on admin has logged into central remotely. The other lines represent services on central that are listening for clients on other computers asking for a connection.

Protocol Statistics

To view the statistics for your computer's Internet protocols, follow the administration tools procedure shown in Figure 8-4.

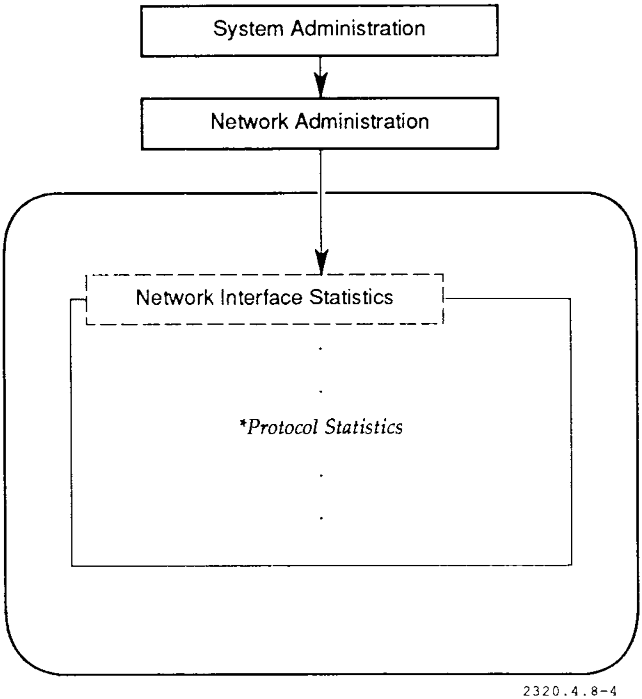


Figure 8-4. Viewing Statistics for Internet Protocols

This procedure is equivalent to invoking **netstat**(1) with the -s option.

Here is a sample output from this menu item:

```
ip:
        87445 total packets received
        0 bad header checksums
        0 with size smaller than minimum
        0 with data size < data length
        0 with header length < data size
        0 with data length < header length
        0 fragments received
        0 fragments dropped (dup or out of space)
        0 fragments dropped after timeout
        0 packets forwarded
        0 packets not forwardable
        0 redirects sent
icmp:
        172 calls to icmp_error
        0 errors not generated 'cuz old message was icmp
        Output histogram:
                destination unreachable: 172
        0 messages with bad code fields
        0 messages < minimum length
        0 bad checksums
        0 messages with bad length
        Input histogram:
                routing redirect: 209
        0 message responses generated
tcp:
        0 incomplete headers
        0 bad checksums
        0 bad header offset fields
        44 retransmitted packets
udp:
        0 incomplete headers
        0 bad data length fields
        0 bad checksums
```

The output of this selection is broken down by protocol. Table 8-1 lists the meaning of each protocol abbreviation.

Table 8-1. Protocol Abbreviations

Abbr.	Protocol	Function
ip	Internet Protocol	Packet delivery
icmp	Internet Control Message Protocol	Error checking
tcp	Transmission Control Protocol	Reliable connections
udp	User Datagram Protocol	Datagram delivery

Internet Troubleshooting

Table 8-2 outlines possible problems and solutions to help troubleshoot Internet problems. The table is arranged as follows:

- *Symptom* is a problem.
- *Additional Tests* are actions that help define the cause of the problem.
- *Suggested Actions* are possible solutions to the problem. Some may not be applicable to your particular case, but correct the symptom in the majority of cases.

The additional tests and suggested actions help you understand the nature of the problem. If the problem cannot be solved, contact your Technical Support representative.

Table 8-2. Problems and Solutions

Symptom	Additional Tests	Suggested Actions
rlogin fails to remote host	**rlogin** to loopback interface	Add missing **rcopts** (see Part 4, Section 1 of the Internet section of this guide). Check that **inetd** is running on the remote host Check **inetd.conf** file for **rlogind** entry on remote host
rlogin fails to loopback interface	Check **rcopts** (see Section 1 of Part 4, the Internet section of this guide)	

Section 9
Network Setup Samples

This section contains setup examples for Internet networks. The three examples fit together, building from a simple network to a complex one. These are the examples:

1. A simple Ethernet network.
2. Adding a gateway to the Ethernet network to allow communication with another Ethernet network.
3. Adding a SLIP link to an Ethernet network.

Simple Ethernet Network

The initial sample network consists of six S/MT computers in the same building. No gateways are used and the computers are far enough apart to make the use of a multiport transceiver impractical. There are no plans to connect these computers to the ARPANET/MILNET.

There is one Ethernet interface in each computer (Ethernet RS-232-C Board, no VME). The interface name on each computer is **en0** when monitored using **ifconfig**(1M) or **netstat**(1).

One computer has been chosen to be the central repository for network configuration files. This computer is named central. The other computers are called admin, finance, stats, print, and quality. When a configuration file is updated, it is first changed on central. It is then copied from central to the other computers. The only configuration file administered on the central computer in this sample is the **/etc/hosts** file. When mail is configured on the network, mail configuration files should also be administered from central.

To set up the Ethernet hardware for the Ethernet network, perform the following steps:

1. Set up the Ethernet hardware (see the Ethernet section of this guide in Part 2 for more details):
   a. String the coaxial cable.
   b. Attach the transceivers to the coaxial cable.
   c. Install the Ethernet interface boards.
   d. Connect the transceiver cables to the interface boards and the transceivers.

To set up the first computer on the Ethernet network, perform the following steps:

1. Start with the central computer.
2. Install the latest software, operating system, and TCP/IP, following carefully the instructions in the release notice.
3. Check that the following empty files exist in the directory **/etc/rcopts**: **KENP, KINET, KSOCK, KSTRM**.
4. Check that the following files are in the directory **/etc/rcopts**: **NETD** and **NODE. NODE** should contain the name of the system (central). **NETD** should contain the names of network services to be run on the system. Many network services are run by **inetd**(1M) and are configured in the file **/etc/inetd.conf**. In this example, the default is acceptable.
5. Note that the default network number (3) from the **/etc/networks** file is used as the network number for the network. It is not necessary to change the **networks** file. Since the network is not connected to other networks, no special routing is needed.

6. Configure the **/etc/hosts** file using the administration tools or an editor.
   1. Assign central the first address, 3.1. The first line of the **hosts** file is

      ```
      3.1    central    # central configuration repository
      ```

   2. Continue with the next computer:

      ```
      3.2    admin      # administration
      ```

   3. Add lines for the rest of the computers. The **hosts** file is now configured.
7. Shut down and reboot the central computer.
8. When the system has booted up, check that the networking hardware and software are working:
   1. Run **ifconfig en0**. The interface should be *UP*.
   2. Run **rlogin loopback**. A login prompt should come back. Enter ^D to close the connection. If this command works, STREAMS, **sockets**, TCP/IP, **rlogind**, and the loopback interface are working.

If both tests work, continue to set up the other systems. If there are problems with these tests, see Section 6, ''Network Monitoring and Troubleshooting,'' in the ''About This Guide'' section in the front of this manual.

To set up the other computers in the Ethernet network, perform the following steps:

1. Install the latest software on the other computers (operating system and TCP/IP) carefully following the instructions in the release notice.
2. Check the **rcopts** files on the other computers.
3. Copy the **hosts** file from central to the other computers (using quarter-inch tape).
4. Reboot the other systems.
5. From the central computer, begin to test connections with the other computers on the network:

a. Type the command:

```
# ping finance 1 1
```

This tests whether packets are being sent from central to finance. The output should look something like this:

```
PING finance: 1 data bytes
9 bytes from 3.0.0.3: icmp_seq=0.

----finance PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
```

b. Next try the **telnet**(1) command:

```
# telnet finance
```

The output should look something like:

```
Trying...
Connected to finance.
Escape character is '^]'.

CTIX (tm: Convergent Technologies) User Mode (s)

login:
```

You can exit by typing:

```
^D
```

or you can log in to finance and verify that the connection is working properly.

c. Next try the **rlogin**(1) command:

```
# rlogin finance
```

The output should look something like:

```
login:
```

Again **^D** exits, or you can log in and test the connection.

If these commands work properly, you have verified that your Ethernet and software protocols are working properly. If any of these commands do not work properly, go to Section 6 of the "About This Guide" section for troubleshooting information.

To set up equivalences on the Ethernet network, perform the following steps:

1. The machine equivalences for this network are as follows:

   - The computer central is equivalent to the computer admin.
   - The computer admin is equivalent to the computer central.

   This means that users on central with accounts on admin are able to log in to admin without giving their passwords (and vice versa).

2. To configure the equivalence on central, edit the file **/etc/hosts.equiv**. The following line is added:

   ```
   admin
   ```

3. Next, edit the file **/etc/hosts.equiv** on admin. Add the following line:

   ```
   central
   ```

4. These equivalences are tested by a user (for example **billr**) on one computer trying to log in to the other computer with the command:

   # rlogin *computer-name*

   The login should be allowed with no password checking. (**rlogin** assumes that the user name is the same on both computers.)

5. Only one user equivalence is configured at the start. The user **johnl** on central wants to give the user **helenw** on admin his permissions. To do this, create the file **/u/johnl/.rhosts** on central. Add the following line to the file using a text editor:

   ```
   admin helenw
   ```

6. User **helenw** tests this equivalence with the command:

   ```
   # rlogin central -l johnl
   ```

If **helenw** can log in without entering a password, the equivalence is working properly.

If there is a problem with these tests, see Section 6, ''Network Monitoring and Troubleshooting,'' in the ''About This Guide'' section in the front of this manual.

Adding an Ethernet Gateway

The next step in the growth of the sample network begins with the decision to connect the network with another in an adjacent building. The second network is an Ethernet network with six computers.

A gateway S/Series is used to link the networks. The gateway will be connected to both networks, but it will be physically set up in the building with the original sample network. The gateway computer has VME capability and two VME Ethernet interface boards. There are no other VME boards on the system. Each interface has a name and an Internet address associated with it.

Recall that the original network is network number three, with the network name Local-Net (the default from the **/etc/networks** file). The second network is network number four, with the network name Four-Net.

Based on the above factors, it is decided that the gateway will have the name Three-GW (address 3.7) on the network three interface and Four-GW (address 4.7) on the network four interface.

When a gateway is added, the **/etc/networks** file is modified. From then on, the **networks** file should be managed on the central computer and copied to all computers on both networks along with the **/etc/hosts** file.

To set up Ethernet hardware on the Ethernet gateway, perform the following steps:

1. Set up the Ethernet hardware (see the Ethernet section [Part 2] of this guide for more details):

a. Extend the coaxial cable from the ''four'' network to the ''three'' building, using a repeater if necessary.

b. Attach a transceiver to the new coaxial cable in the ''three'' building within range of the gateway computer.

c. Attach a transceiver to the original network ''three'' coaxial cable within range of the gateway computer.

d. Set up the gateway computer.

e. Install the two VME Ethernet interface boards in the gateway computer.

f. Connect the transceiver cables to the interface boards and the transceivers.

To set up software on the gateway computer, perform the following steps:

1. Install the latest software (operating system and TCP/IP), carefully following the instructions in the release notice.

2. Modify the **/etc/system** file to configure the two Ethernet interfaces. The following lines should be present:

   ```
   * (one CMC ethernet controller - each requires 128K of space)
   0     1     C0DE0000     131072
   1     1     C0E00000     131072
   ```

   The controller in slot 0 is **en0**, and the controller in slot 1 is **en1**.

3. Check that the following empty files exist in the directory **/etc/rcopts**: **KINET**, **KSOCK**, and **KSTRM**.

4. The following files also should be in the directory **/etc/rcopts**: **KENP**, **NETD**, and **NODE**. **NODE** should contain the name of the system (Three-GW). **NETD** should contain the names of network services to be run on the system. Note that many network services are run by **inetd**(1M) and are configured in the file **/etc/inetd.conf**. In this example, the default is acceptable.

KENP should contain the Internet names of the two interfaces:

```
Three-GW
Four-GW
```

The order of names in the **KENP** file is important. Three-GW must correspond to the **en0** interface, while Four-GW corresponds to the **en1** interface.

ROUTING should contain the line

```
/etc/routed -g
```

This specifies the gateway as the default router. Because there are no distant routes, it is not necessary to configure an /etc/**gateways** file.

To set up the configuration files on the Ethernet gateway, perform the following steps:

1. Add the new network to the /etc/**networks** file on central. This line needs to be added:

   ```
   Four-Net    4    # Second network
   ```

   Copy the file from central to all hosts on both networks, including the gateway.

2. Add the entries for the gateway to the **hosts** file on central:

   ```
   3.7   Three-GW    # Network 3 side of gateway
   4.7   Four-GW     # Network 4 side of gateway
   ```

   Add the entries for the computers on network 4 to the **hosts** file on central. Copy the /etc/**hosts** file from central to all hosts on both networks, including the gateway.

3. Shut down and reboot all the computers. This configures the interfaces on all computers and allows **routed** to construct routing tables on all computers.

To test the gateway, perform the following step:

1. Test the operation of the gateway using the **ping** command. From the central computer, type:

   ```
   # ping 4.3 1 1
   ```

   Since central is on network 3 and host 4.3 is on network 4, data must go through the gateway to reach host 4.3. The output should look like this:

   ```
   PING 4.3: 1 data bytes
   9 bytes from 4.0.0.3: icmp_seq=0.

   ----4.3 PING Statistics----
   1 packets transmitted, 1 packets received, 0% packet loss
   ```

Adding a SLIP Link

In addition to the two buildings mentioned so far (which are now connected using Ethernet) a sales office in another city wants to be able to connect to the main network(s). A SLIP connection is to be set up between the central computer on network three and the sales office computer.

The sales office computer is called sales-br, and it has the internet address 5.0.0.1. Note that it is on a different network from the computers we have discussed so far, network 5.

The connection between central and sales-br is established over telephone lines using modems. Since the link is used mostly for mail and some file transfer in the late evening, a 2400 baud line is used. The modems are Hayes compatible.

To set up the hardware for the SLIP link, perform the following steps:

1. Have the telephone lines installed on both ends of the connection.
2. Install the modems on central and sales-br using the installation guides supplied with the modems.

To set up the software for the SLIP link, perform the following steps:

1. Configure the **/etc/inittab** files on both computers. Both computers will use **/dev/tty001** as the modem port. Modify the **tty001** line in the **inittab** file to look like this:

   ```
   001:2:respawn:/usr/lib/uucp/uugetty -r -t 60 tty001 2400
   ```

2. Add the following line to the **/usr/lib/uucp/Devices** file on both systems:

   ```
   ACU tty001 - 2400 hayes
   ```

   You can do this with the administration tools or a text editor.

3. Give the **slip** user a password on both systems. This can be done as **root** with the following command:

   ```
   # passwd slip
   ```

   For this sample, the password for central is Inter87 and the password for sales-br is Inter78.

4. Add the following line to the /usr/**lib/uucp/Systems.slip** file on central:

   ```
   sales-br Any ACU 2400 9=6143217654 in:--in: slip word: Inter78
   ```

5. Add the following line to the /usr/**lib/uucp/Systems.slip** file on sales-br:

   ```
   central Any ACU 2400 9=4082348765 in:--in: slip word: Inter87
   ```

6. Add the following line to the /etc/**hosts** file on central:

   ```
   5.0.0.1    sales-br    # SLIP link to sales office
   ```

   Copy the changed **hosts** file to all hosts on the internetwork.

7. Add the following line to the /etc/**networks** file on central:

   ```
   Sales-Net    5    # Sales network
   ```

   Copy the changed **networks** file to all hosts on the internetwork.

8. Reboot both systems.

To test the SLIP link, perform the following step:

1. To test SLIP from central, use the command:

   ```
   # rlogin sales-br
   ```

   A login prompt should be returned. Expect a several second delay.

Glossary

A

ARPANET

Department of Defense Advanced Research Projects Administration Network.

active route

Active routes are associated with gateways that have *routed* running. Status information is expected from these *gateways*.

address class

The A, B, and C address classes are internet address formats suitable for large, medium, and small networks. See also *address class identifier*.

address class identifier

The address class identifier is a set of bits in the internet address that identify the *address class* of the address. The address class identifier is 0, 10, or 110 (binary) for class A, B, or C, respectively.

C

coaxial cable

The coaxial cable is a shielded cable that serves as the transmission medium for Ethernet networks.

collision

Collision means that two hosts are transmitting data onto the *coaxial cable* at the same time. When a collision occurs, the data must be retransmitted.

D

DDN
Defense Data Network.

daemon
A daemon is a program that normally runs in the background: for example, **rlogind** (the remote login daemon), **rwhod** (the remote status daemon), and so forth.

default route
The default route is the entry in the system routing tables that is used when there is no other route to a destination. Internally, the default route is represented as the gateway to network 0.

dynamic routing
Dynamic routing means that routes are automatically initialized and updated by the **routed** service.

E

Ethernet
Ethernet is a 10-Mbps local area network system. Ethernet on CTIX computers follows IEEE standard 802.3.

external route
An external route is a route that should not be added by **routed** because another type of routing service will add it.

G

gateway

A gateway is a computer that passes data from one network to another. A gateway computer has more than one network interface, and each network interface has an associated network address.

H

host

The host is the main system in a given network.

host number

The host number is an integer that identifies a particular host. The host number is part of the internet address of a host. See also *network number.*

host route

A host route specifies the gateway to use from the local host to another host (on the local network or on another network).

I

IP

Internet Protocol.

internet address

The internet address is a 32-bit integer that uniquely identifies the location of a host.

internetwork

An internetwork consists of two or more connected networks.

Internet Protocol

The Internet Protocol provides for delivery of network data packets. It is not designed to provide reliability. See also *Transmission Control Protocol.*

L

LAN

Local Area Network.

M

MILNET

Military Network.

machine equivalence

An equivalent machine is a remote computer to which you have given special permissions. Users on the equivalent machine with the same name as users on your machine are automatically *equivalent users.*

metric

The metric for a route is the number of gateway hops needed to reach the route's destination.

modem

A modem is a serial communications device that allows transmission of computer data across regular phone lines.

module

A module is a driver used to support an operating system function; networking is an example.

N

NFS

Network File System.

NIC

DDN Network Information Center.

netmask

The netmask is the interface parameter used to derive the subnet number from the internet address.

network number

The network number is an integer that identifies a particular network. The network number is part of the internet address of a host. See also *host number*.

network route

A network route specifies an intermediate host through which traffic from the local network can be routed on the way to destinations in another network. Each network route specifies the route to one network.

P

passive route

Passive routes (gateways) are computers where **routed** is not running.

ports

Ports are logical numbers assigned to internet services.

protocol

A protocol is a set of rules for the format and timing of data exchanged between communicating systems.

R

RFS

Remote File Sharing.

RPC

Remote Procedure Call.

root

The root is the main system in an internetwork.

routing

Routing is the process by which data are directed from one host to another. If the hosts are on different networks, data are transmitted through gateways to the final destination.

S

SLIP

SLIP (Serial Line Internet Protocol) is a set of communication programs that allows users to use Internet commands between computers connected with a serial line.

STREAMS

STREAMS is an enhancement to the UNIX character input/output (I/O) system that supports the development of communication software. STREAMS is new to the AT&T UNIX 5.3 and the CTIX 6.0 releases.

sockets

Sockets are a programming interface used to facilitate the development of network systems.

static routing

In static routing, the network administrator configures the system routing tables using the **route** command. This can be done automatically at boot time.

subnet

A subnet is a portion of a network.

super-user

Super-user status is conferred on a user whose user id is 0. Super-user status removes important CTIX restrictions: the super-user can write to or read from any ordinary or special file; the super-user can also execute certain commands that no other user can execute.

subnet number

The subnet number is the part of the *host number* used to identify the hosts in a *subnet.*

T

TCP

Transmission Control Protocol.

TLI

Transport Layer Interface.

transceiver

The transceiver is an electronic device that connects the coaxial cable to each computer. The transceiver detects signals sent from other computers and signal *collisions* that occur when two or more computers try to broadcast at the same time.

Transmission Control Protocol

The Transmission Control Protocol provides for reliable network connections between hosts.

tunable parameters

The system-wide variables that can be adjusted to improve system performance.

U

user equivalence

An equivalent user is a user on a remote system who is given the privileges of a user on the local system.

UUCP

UUCP (UNIX-to-UNIX copy program) is a batch-oriented file transfer protocol that can run on direct serial lines, telephone lines, and networks.

W

WAN

WAN is an X.25 Wide Area Network.

Index