HP AdvanceNet

HP 9000 Series 300/400 and 600/700/800 Computers
Installing and Administering LAN/9000

HEWLETT PACKARD

Edition 2
E0691

98194-60526
Printed in U.S.A. 06/91

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Hewlett-Packard Co.
19420 Homestead Rd.
Cupertino, CA 95014 U.S.A.

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

Note that many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

Edition 1 . . . . . February 1991
Edition 2 . . . . . June 1991

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

List of Effective Pages

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

Pages	Effective Date
All	June 1991

Preface

This manual provides information for installing and administering the LAN/9000 product. The LAN/9000 product allows HP 9000 computers to connect to an IEEE 802.3 or Ethernet Local Area Network.

The manual describes how to load, configure and initialize LAN software. It also describes how to maintain the network interface and how to use troubleshooting utilities. Finally, the manual provides a listing of diagnostic and event logging messages.

The information in this manual is intended for network managers or operators who install and administer LAN/9000. It is assumed the reader is experienced with HP-UX and is familiar with the basics of local and wide area networking.

The manual is organized as follows:

Chapter 1	"Product Overview" describes product structure, relates software components to the Open Systems Interconnect (OSI) Reference Model and lists useful manuals.
Chapter 2	"Networking Concepts" defines networking terms and explains network interface name and unit, network addresses, names and subnets, and LAN device concepts.
Chapter 3	"Installing LAN" describes how to load and configure LAN/9000 software. It includes steps for configuring software automatically using the System Administration Manager (SAM).
Chapter 4	"Maintaining LAN" explains how to administer LAN once the network is up. It also lists useful networking daemons and library routines.

Chapter 5	"Troubleshooting LAN" provides flowcharts for diagnosing common LAN/9000 problems.
Chapter 6	"Using Network Diagnostics" explains useful diagnostic utilities.
Chapter 7	"Using the Logging and Tracing Facility" describes the common logging and tracing tool, *nettl*.
Appendix A	"Installation Error Messages" lists error messages related to loading and configuring LAN/9000 software.
Appendix B	"Diagnostics Error Messages" lists error messages returned by diagnostic utilities.
Appendix C	"Network Event Logging Messages" lists network event logging messages returned by LAN/9000.
Appendix D	"LANIC Statistics" describes LAN card status values and statistics returned by *landiag(1M)*.
Appendix E	"LANIC Self-test Codes" lists error codes returned by a "failed" *landiag(1M)* interface test.
Appendix F	"LAN Filesets" describes the S800 include statements and S300/S700 keywords required for generating a new kernel.

Contents

Chapter 2 Networking Concepts

Chapter 3 Installing LAN

Chapter 4 Maintaining LAN

Chapter 5 Troubleshooting LAN

Chapter 6 Using Network Diagnostics

Figures

Tables

Syntax Conventions

nonitalics

Words in syntax statements which are not in italics must be entered exactly as shown. Punctuation characters other than brackets, braces, and ellipses must also be entered exactly as shown. For example:

```
EXIT;
```

italics

Words in syntax statements that are in italics denote a parameter that must be replaced by a user-supplied variable. For example:

```
CLOSE filename
```

[ ]

An element inside brackets in a syntax statement is optional. Several elements stacked inside brackets indicates the user may select any one or none of these elements. For example:

```
[A]
[B]  User may select A or B or C or none.
[C]
```

{ }

When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example:

```
{A}
{B}  User must select A or B or C.
{C}
```

...

A horizontal ellipsis in a syntax statement indicates that a previous element may be repeated. For example:

```
[,itemname]...;
```

In addition, vertical and horizontal ellipses may be used in examples to indicate that portions of the example have been omitted.

A shaded delimiter preceding a parameter in a syntax statement indicates that the delimiter must be supplied whenever (a) that parameter is included or (b) that parameter is omitted and any other parameter that follows is included. For example:

```
itema[,itemb][,itemc]
```

means that the following are allowed:

```
itema
itema,itemb
itema,itemb,itemc
itema,,itemc
```

Δ

When necessary for clarity, the symbol Δ may be used in a syntax statement to indicate a required blank or an exact number of blanks. For example:

```
SET[modifier] Δ (variable)
```

underlining

Brackets, braces, or ellipses appearing in syntax or format statements which must be entered as shown will be underlined. For example:

```
LET var[[subscript]] = value
```

Output and input/output parameters are underlined. A notation in the description of each parameter distinguishes input/output from output parameters. For example:

```
CREATE (parm1,parm2,flags,error)
```

[Key Cap]

A string in bold font enclosed by brackets may be used to indicate a key on the terminal's keyboard. For example, **[Enter]** indicates the carriage return key.

[CTRL]-char

Control characters are indicated by **[CTRL]** followed by the character. For example, **[CTRL]-Y** means the user presses the control key and the Y key simultaneously.

1

Product Overview

This chapter is an overview of the LAN/9000 product. It includes:

- Product Structure.
- Product Description.
- Reference Manual Guide.

Product Structure

The LAN/9000 product consists of hardware and software components that allow you to connect an HP 9000 to an IEEE 802.3 or Ethernet local area network. Hardware components vary somewhat for different HP 9000 models. With a few minor exceptions, software is identical for all models. The exceptions are clearly noted in this manual.

Hardware Components

The main hardware component is the LAN Interface Controller. Card (LANIC). This may be referred to as the LAN card (Series 600/800), the System Card (Series 300), or the Core IO card/EISA card (Series 700). This manual uses the terms LAN card, System card, and CORE IO card interchangeably to indicate that the LAN card is the communication link between HP 9000 systems and the Local Area Network. Depending on your HP 9000 model, other hardware components may include the Medium Attachment Unit (MAU), Attachment Unit Interface (AUI) cable and the stub cable.

LAN Card

The LAN card is the communication link between HP 9000 and the LAN. It transmits and receives data and control packets. It also monitors collisions on the LAN to ensure collided frames are retransmitted. Depending on your HP 9000 model, you may have one of the types of LAN card as shown in the following table.

Table 1-1. Types of LAN Card

HP 9000 Model	LAN Card
All Series 300/400 models	98171A (DIO) LAN Card
All Series 600 models	36967A-20C (CIO) LAN Card
Models 808, 815, 822, 832	36967A-20N (NIO) LAN Card
All other Series 800 models	36967A-20C (CIO) LAN Card
All Series 700 models	A1094-66530 CORE IO Card 25567A Add-on EISA Card

Each of these is functionally equivalent but physically different. Each also is available in different configurations.

Series 300/400 models and Series 600/800 workstation models come with a LAN card installed. In the case of Series 300/400 models, the factory-installed LAN card is actually part of the mother board. For Series 600/800 models, it is a separate card. Series 700 workstations only have one Core IO (LAN) card.

All HP 9000 models can accommodate additional "add-on" LAN cards for gateway use. For Series 600/800 workstations, the add-on cards are identical to factory-installed LAN cards. For Series 300/400 models, add-on cards are physically different than the factory-installed units. For Series 700 models, the add-on cards are all EISA cards. All series can accommodate up to four add-on cards for a total of five LAN cards.

MAU

The Medium Attachment Unit (MAU) connects the LAN card to the LAN medium. There are two types of MAU: ThinMAU, for use with thin coaxial cable; ThickMAU for thick (10 mm) coaxial cable. HP supplies these cables as ThinLAN and ThickLAN, respectively.

The MAU passes packets between the LAN card and network cable. In addition, it prevents LAN card malfunctions from jamming the network.

For some Series 300/400 models, the factory-installed LAN card can be ordered with integrated ThinMAU. In this case, the motherboard connects directly to ThinLAN. Series 300/400 models not equipped with integrated ThinMAU have a 15-pin AUI connector on the motherboard. The connector allows attachment to an offboard MAU for ThinLAN, ThickLAN or Ethertwist connection.

AUI

The Attachment Unit Interface (AUI) cable connects the LAN card to the MAU. As noted above, an AUI may or may not be required, depending on your HP 9000 model number and configuration. The AUI cable is available in several sizes. This allows flexibility in the distance between the HP 9000 and LAN cable.

Stub Cable

For Series 600/800 CIO models, a stub cable links the LAN card to the AUI cable. One end of the stub cable plugs into a 15-pin connector on the LAN card. The other end plugs into the D-connector on the AUI cable.

Software Components

LAN/9000 software may be provided on tape or disc, depending on your HP 9000 model. LAN/9000 software includes programmatic interfaces, network protocol modules and tools for LAN administration.

Programmatic Interfaces

Programmatic interfaces include NetIPC, Berkeley Sockets (BSD IPC) and Link Level Access (LLA).

NetIPC and Berkeley Sockets allow peer process communication between an HP 9000 and other network nodes. They provide programmatic access to the Transport Layer (OSI Layer 4).

Link Level Access provides an interface to the Link Layer (OSI Layer 2). It allows direct access of network drivers using standard HP-UX system calls.

Protocol Modules

LAN/9000 provides various protocols to implement network communication at the Physical, Link, Network and Transport Layers (OSI Layers 1-4). Protocol modules include: TCP, PXP, UDP, IP, Probe, ARP, and an IEEE 802.3/Ethernet Driver. A brief description of each protocol is provided later in this chapter.

Maintenance and Troubleshooting Tools

LAN/9000 provides tools to help with network administration. This includes:

- **Network event logging and tracing**: This utility allows you to log network events and trace record packets as they enter and exit the LAN driver. This utility is implemented with *nettl()*.
- ***ping(1M)***: This utility verifies a connection between systems that support *ping(1M)* (includes most UNIX systems). If the test is successful, *ping(1M)* reports the round-trip time used in the local-to-remote-to-local communication.
- ***netstat(1M)***: This utility reports network and protocol statistics regarding packet traffic and network communications.
- ***rlb(1M)***: This utility tests connectivity through the Transport Layer.
- ***linkloop(1M)***: This utility tests connectivity through the Link Layer.
- ***ifconfig(1M)***: This utility allows you to configure LAN/9000 software.
- ***lanconfig(1M)***: This utility allows you to configure LAN/9000 software.
- ***route(1M)***: This utility allows you to manipulate the network routing table.
- ***nodename(1M)***: This utility allows you to configure and display the official node name of your system.
- ***hostname(1M)***: This utility allows you to configure and display the official host name of your system.
- ***landiag(1M)***: This utility checks LAN card status and resets the LAN card.
- ***lanscan(1M)***: This utility displays information about LAN cards that are successfully bound to the system.

Note For Series 600/800 and Series 700 models only, the HP-UX operating system provides an additional utility called *LANDAD*. *LANDAD* is part of the HP-UX On-line Diagnostic Subsystem. *LANDAD* performs the same functions as *linkloop* and *landiag*. In addition, it provides MAU, AUI and internal loopback tests.

Product Description

Table 1-2 shows the relationship of the LAN/9000 product to the OSI model. For details on the OSI model, refer to the *Networking Overview* manual. The figure also shows the relationship of the LAN product to network services that typically run on it: NS/9000, ARPA/9000 and NFS/9000.

Following is a brief description of LAN/9000 software as it relates to processes within each OSI layer.

Table 1-2. Relationship of LAN/9000 to Services & OSI Model

OSI Model	Network Services (NS)	ARPA Services	Berkeley Services	NFS Services	Link Level Access	Product Structure
7 Application	Network File Transfer (NFT) Virtual Terminal for HP 3000 (VT3k)	File Transfer Protocol (ftp) Telnet (telnet)	Remote Copy (rcp) Remote Login (rlogin) Remote Execution (rexec) Remote Shell (remsh) Remote Who (rwho) Remote Uptime (ruptime) Sendmail	Network File System (NFS) Network Information Systems (NIS) Virtual Home Environment (VHE)		Services
6 Presentation		Simple Mail Transfer Protocol (SMTP)		External Data Representation (XDR)		
5 Session	NetIPC		BSD IPC (Berkeley) Sockets	Remote Procedure Call (RPC)		
4 Transport	Transmissions Control Protocol (TCP)	Transmissions Control Protocol (TCP)	Transmissions Control Protocol (TCP), User Datagram Protocol (UDP)	User Datagram Protocol (UDP)		LAN/9000
3 Network	Internet Protocol (IP)	Internet Protocol (IP)	Internet Protocol (IP)	Internet Protocol (IP)	Link Level Access Applications	
2 Data Link	IEEE 802.3	Ethernet	Ethernet	Ethernet	Ethernet/ IEEE 802.3	
1 Physical	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	

Session Layer (OSI Layer 5)

The LAN/9000 product provides two programmatic interfaces to the Transport Layer:

- NetIPC
- Berkeley Sockets (BSD IPC)

NetIPC

NetIPC enables processes running on HP 9000 nodes on the network to exchange information between other HP 9000s, HP 1000 A-Series, HP 3000 MPE-V and MPE-XL, HP Vectra PC and IBM PC nodes on the network. NetIPC provides an interface between the Application Layer services and the transport protocols in the Transport Layer.

Berkeley Sockets

Berkeley Sockets enables processes running on UNIX nodes on the network to exchange information. HP's implementation of sockets is based on the IPC in the Berkeley Software Distribution of UNIX, version 4.3 (4.3 BSD).

Note For details on NetIPC and Berkeley Sockets, refer to the *NetIPC Programmer's Guide* and *Berkeley IPC Programmer's Guide*, respectively.

Transport Layer (OSI Layer 4)

At the Transport Layer, LAN/9000 provides the following protocol modules:

- TCP
- PXP
- UDP

TCP

Transmission Control Protocol (TCP) is the main Transport Layer protocol for LAN/9000. It is based on the DARPA standard. TCP provides non-duplicated, in-sequence data delivery. It is a stream-based (rather than message-based) protocol. TCP accepts data buffers up to 64 Kilobytes long, divides them into packets and sends each packet separately. TCP keeps track of the bytes sent and retransmits them if they are not acknowledged within a timeout interval. TCP at the receiving node reassembles the packets so that they are delivered to the user (NetIPC or BSD IPC) in order (in-sequence delivery).

Because TCP is a connection-based protocol, it requires more initial overhead than a datagram-based protocol. When the TCPs at two nodes want to communicate, they establish a logical communication channel called a connection. Establishing a TCP connection requires overhead because each node must allocate buffers and other resources to support the connection, and because the TCPs must perform a connection "handshake" before any data is sent. TCP also provides flow control. The amount of data sent can be controlled so that the sender does not overload the receiver.

PXP

Packet Exchange Protocol (PXP) is another Transport Layer protocol for LAN/9000. PXP is an HP proprietary, low-overhead request/reply datagram protocol that is suited for querying data sources. Since PXP does not establish connections, subsequent transactions cannot take advantage of an established connection. PXP retransmits requests that are not acknowledged within a timeout interval. PXP is used internally by NetIPC and is not directly accessible to users.

UDP

User Datagram Protocol (UDP) is an unreliable, connectionless Transport Layer protocol. Unlike TCP, there is no concept of a connection. Messages are sent as a unit with source and destination information in the header. As there is no concept of a connection, there is no way to verify that the message arrived at the destination.

The ARPA/Berkeley Services *rwho(1)*, *ruptime(1)*, and *bind(1)* use UDP. NFS Services primarily uses UDP.

Network Layer (OSI Layer 3)

At the Network Layer, LAN/9000 implements the Internetwork Protocol (IP) based on the DARPA standard. IP is a connectionless delivery mechanism for internetwork packet routing. It defines an internet addressing scheme which can uniquely identify multiple networks as well as a node within a single network.

Physical and Data Link Layers (OSI Layers 1-2)

At the Physical and Data Link Layers LAN/9000 implements:

- IEEE 802.3/Ethernet Driver
- Link Level Access
- Probe
- ARP

IEEE 802.3 Driver

IEEE 802.3 defines a baseband, coaxial bus media with a speed of 10 Megabits per second, CSMA/CD and IEEE 802.2 support.

Under CSMA/CD, all nodes have equal access to the media. Each node listens to network traffic. If there is no traffic on the network, a node can begin to transmit. If two or more nodes transmit at the same time, they detect a collision and stop transmitting. Each node waits for a random period of time to retransmit.

The 802.2 Logical Link Control protocol defines the data link level frame and its associated headers.

Ethernet Driver

Ethernet is a popular de-facto standard, developed before IEEE 802.3 was defined. (IEEE 802.3 has evolved from Ethernet.) Like IEEE 802.3, Ethernet also defines a baseband, coaxial, bus media utilizing CSMA/CD. IEEE 802.3 and Ethernet nodes can coexist on the same cable, but cannot communicate with each other.

The portions of LAN/9000 that implement IEEE 802.3 and Ethernet are the driver, the LAN card, and the remaining hardware that connects the HP/9000 to the LAN cable.

Link Level Access

In addition to the preceding protocols, LAN/9000 provides Link Level Access (LLA), which allows direct access to Link Layer network drivers using standard HP-UX file system calls. Because it provides access to layer 2, LLA allows you to create applications that communicate with other vendors that also implement IEEE 802.3/Ethernet at layers 1 and 2, but that do not implement the same protocols as HP at higher layers. LLA also provides an alternative to using the process-to-process communication services provided by NetIPC and BSD IPC.

Note For details on LLA, refer to the *LLA Programmer's Guide.*

Probe

Probe is an HP protocol that is used by NetIPC. It translates NetIPC node names into physical addresses via a two-step process (name-to-IP address resolution and IP address-to-physical address resolution). Probe multicasts the name of a node to all other nodes in the network. The node that is associated with the node name being broadcast identifies itself by replying to Probe with its IP addresses and protocols supported. Probe also translates IP addresses to hardware addresses (also called station addresses or link-level addresses). Probe, like PXP, has very low overhead. It is not directly accessible to users.

ARP

ARP provides similar functionality to Probe. ARP translates IP addresses to physical addresses via a two-step process (name-to-IP address resolution and IP address-to-physical address resolution. However, ARP does not translate user-defined node names into machine-readable addresses. ARP is directly accessible to users with the *ARP(1M)* command.

Reference Manual Guide

Table 1-3. List of Manuals

For Information on:	Read:
System Administration	*HP 9000 System Administrator's Manual*
Installing LAN Hardware	*LAN Interface Controller (LANIC) Installation and Reference Manual* *LAN Cable and Accessories Manual*
Troubleshooting LAN	*Installing and Administering LAN/9000*
Troubleshooting NS	*Installing and Administering NS*
Troubleshooting ARPA Services	*Installing and Administering ARPA Services*
C Programming Language	*The C Programming Language*, Brian W. Kernighan, Dennis M. Ritchie; © 1978 Bell Telephone Laboratories, Inc., Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632 *HP-UX C Programmer's Guide* *HP-UX C Quick Reference Guide* *HP-UX C Reference Manual Supplement*
General ARPA/Berkeley Services Information	*Using ARPA Services*
General NS/9000 Information	*Using Network Services*
HP-UX Operating System	*HP-UX User's Guide* *HP-UX Reference Manuals* *HP-UX Concepts and Tutorials*

Table 1-3. List of Manuals (cont.)

For Information on:	Read:
HP-UX Reference Manuals	*HP-UX Reference Manuals*
Internetwork Mail Routing	*Sendmail–An Internetwork Mail Router* (Document reference number: UNX11.2.4), Eric Allman, Academic Computing Services Library, University of California at Berkeley, 218 Evans, Berkeley, California 94720
Subnetting	RFC 950
Real-Time Operations	*Real-Time Programming Manual*
General NFS Services Information	*Using and Administering NFS Services* *Programming and Protocols for NFS Services*
Link Level Access	*LLA Programmer's Guide*
BSD Interprocess Communication	*Berkeley IPC Programmer's Guide*
Network Interprocess Communication (NetIPC)	*NetIPC Programmer's Guide*
Protocols: Address Resolution Protocol (ARP)	RFC 826
Domain Requirements	RFC 920
File Transfer Protocol (FTP)	MIL-STD 1780; RFC 959, 765, 678
Internet Control Message Protocol (ICMP)	RFC 792
Internet Protocol (IP)	MIL-STD 1777; RFC 791
Simple Mail Transfer Protocol (SMTP)	MIL-STD 1781; RFC 821
Standard for the Format of ARPA Internet Text Messages	RFC 822

Table 1-3. List of Manuals (cont.)

For Information on:	Read:
Telnet	MIL-STD 1782; RFC 854
Transmission Control Protocol (TCP)	MIL-STD 1788; RFC 793, 813, 814, 816, 817, 179, 889, 896
Sysdiag References	*On-Line Diagnostic Subsystem Manual*

Military Standards and Request for Comment Documents

To obtain information about available RFCs, contact the:

Network Information Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

To obtain information about available MIL-STD specifications, contact:

Department of the Navy
Naval Publications and Forms Center
5801 Tabor Avenue
Philadelphia, PA 19120-5099

2

Networking Concepts

This chapter introduces networking terms and concepts used throughout this manual. It contains the following sections:

- Networking Terminology.
- Network Addresses and Node Names.
- Internet Addresses.
- Subnet Addresses.
- LAN Device Terminology.

Networking Terminology

Following is a description of important networking terms. Become familiar with these terms before attempting LAN procedures.

Nodes

A node or a host is a computer on the network. Local node refers to the node or host to which your terminal is physically attached. A remote node is a computer on the network with which your local node can communicate. A remote node does not have to be directly attached to your terminal.

Routes and Protocols

A route is the sequence of network nodes through which messages travel when sent from a source node to a destination node.

A protocol is a set of rules for a particular communication task. A protocol handler or protocol module is a piece of software that implements a particular protocol.

Network Interface Name and Unit

A network interface configured into a system defines a path through a stack of protocols and, optionally, a hardware device through which packets can be sent and received.

Each network interface is identified by a name and a unit. The name and the unit together form the interface identifier. The unit number can range from 0 to 4 because a maximum of five LAN cards are supported on each system.

A loopback interface does not have a hardware device associated with it. For example, the name and unit of this type of interface might be *lo0*, denoting loopback interface unit 0.

For a network interface associated with a LAN card, the network protocol, e.g. IP, accesses the LAN driver via the network interface. For example, the name and unit of this type of interface might be *lan0*, denoting interface unit 0.

For the Series 600/800, the network interface unit is assigned according to the physical location of the LAN card in the backplane. The LAN card in the lowest hardware module is interface unit number 0; the LAN card in the next higher hardware module is interface unit number 1; and so on. If there is more than one LAN card in a module, e.g. CIO, interface unit numbers are assigned to the LAN cards in that module before numbers are assigned to those in the next higher module.

For the Series 300/400, the network interface unit is assigned according to the order of the select code on the LAN cards. This follows the same scheme as the device logical unit numbers. As a result, the network interface unit is the same as the device lu on each LAN card.

For the Series 700 the network interface unit is assigned in the order in which the LAN cards are detected by the IO subsystem. The LAN Interface Controller on the Core IO card is detected first followed by that on the EISA cards, if any. The network interface unit for the Core IO card is 0; the network interface unit value ranges from 1 to 4 for the EISA card(s).

You can use the *lanscan(1M)* command to display the network interface name and unit of each network interface that is associated with a LAN card.

Gateway

A network gateway is a device used to connect two or more networks. The gateway serves to route information among the networks. An HP 9000 with two or more LAN cards installed may act as a LAN-to-LAN gateway. Such a node may also be referred to as a LAN-to-LAN router or IP router. If a node is a gateway, it affects how you configure and maintain LAN software.

Routing Table

Each node on the LAN has a routing table. The table contains information about the route to nodes on other LANs. The connections are made through gateways. When additional gateways are added, or network addresses change, the routing table must be updated.

Subnets

Subnetting is an addressing scheme that allows you to partition an IP address into discrete subnets. This feature is useful if you have several physical networks sharing the same network address. The IP address format and subnets are described later in this chapter.

Network Addresses and Node Names

Several types of names and addresses are used in networking software. This can be confusing to first time users. Following is a summary of names and addresses used by LAN and the services which run on it.

Table 2-1. Network Addresses and Node Names

Address Type	Description	Recorded In	Used By
link level address	Also referred to as *Ethernet address*, *LAN address*, *IEEE802.3 address*, *local network address*, *network station address*, and *station address*. A link level address is the unique address of the LAN interface card. This value is set at the factory and cannot be changed. An example of a link level address in hexadecimal: 0800090012AB.	interface card; */etc/clusterconf*	*linkloop* diagnostic; internals of networking software to uniquely identify nodes on the LAN; cnode definition in a cluster during reconfig; displayed by *landiag* and *lanscan* diagnostics.
internet address	Also referred to as *IP address*. An internet address is the network address of a computer node. This address identifies both which network the host is on (of all ARPA networks that are registered with the Network Information Center) and which host it is. An example of an internet address: 192.6.23.3	*/etc/hosts*; */etc/netlinkrc*	Internals of the networking software. Many of the services allow the use of the internet address, its corresponding host name or an alias. HP-UX *reconfig* command.
network address	Also, *network number*. The network address is the network portion of an internet address that represents the local network on which a host exists. The network address is the same for all nodes on that network.	*/etc/networks*	Routing facility. Displayed by *netstat -rn*.

Table 2-1. Network Addresses and Node Names (cont.)

Address Type	Description	Recorded In	Used By
host address	The host address is that portion of the internet address that is unique to the network. The host address identifies a particular node on the network.	Combined with network address in */etc/hosts*.	Internals of the networking software in combination with the network address. Many of the services allow the use of the internet address, its corresponding host name or an alias. HP-UX *reconfig* command.
port address	Also referred to as *TCP port number*, *UDP port number*, or simply *port*. A port address is an address within a host that is used to differentiate between multiple communication endpoints with the same internet address and protocol. A port address is associated with a particular service. Port numbers are defined by RFC 923, *Assigned Numbers*.	*/etc/services*	Service requests. Displayed by *netstat -a* or *netstat -an*.
socket address	This address is declared in processes defined by the interprocess communication software. Refer to *Using ARPA Services* for more information on interprocess communication.	socket address variables	Interprocess communication.

Table 2-1. Network Addresses and Node Names (cont.)

Address Type	Description	Recorded In	Used By
system host name	Also referred to as the *HP-UX host name* and *system node name*. This is the name your HP-UX system is known by and is assigned using the HP-UX *hostname* command.	*/etc/rc*	*uucp* facilities.
host name	Also known as the *ARPA host name* and *NFS host name*. A symbolic name associated with an internet address by which a node can be uniquely identified. An example of a host name is: *paul*.	*/etc/hosts*; */etc/hosts.equiv* (optional); *$HOME/.rhosts* (optional); *$HOME/.netrc* (optional); */usr/adm/inetd.sec* (optional).	All ARPA and Berkeley services.
node name	Also known as the *NS node name*. A three-field symbolic name by which a node can be uniquely identified by the Network Services. The syntax for this name is: *node.domain.organization* and is assigned using the *nodename* command. An example of a node name is *paul.mfg.hp*.	*nodename* is recorded by the *nodename* command; it is also recorded in the proxy table via the *proxy* command.	Network file transfer (NFT); *rlb* diagnostic utility; Virtual Terminal for HP 3000 (VT3k).

Internet Addresses

Internet addresses are used extensively by LAN/9000 as well as NS/9000 and ARPA/9000 Services.

An internet address (often referred to as the IP address) consists of two parts:

- Network address.
- Host address.

The network address identifies the network. Host address identifies a node within the network. A network address is concatenated with a host address to form the internet address and uniquely identify a node within a network.

Internet Address Formats

There are three internet address classes, each accommodating a different number of network and host addresses. The address classes are defined by the most significant bits of the binary form of the address as shown in Figure 2-1.

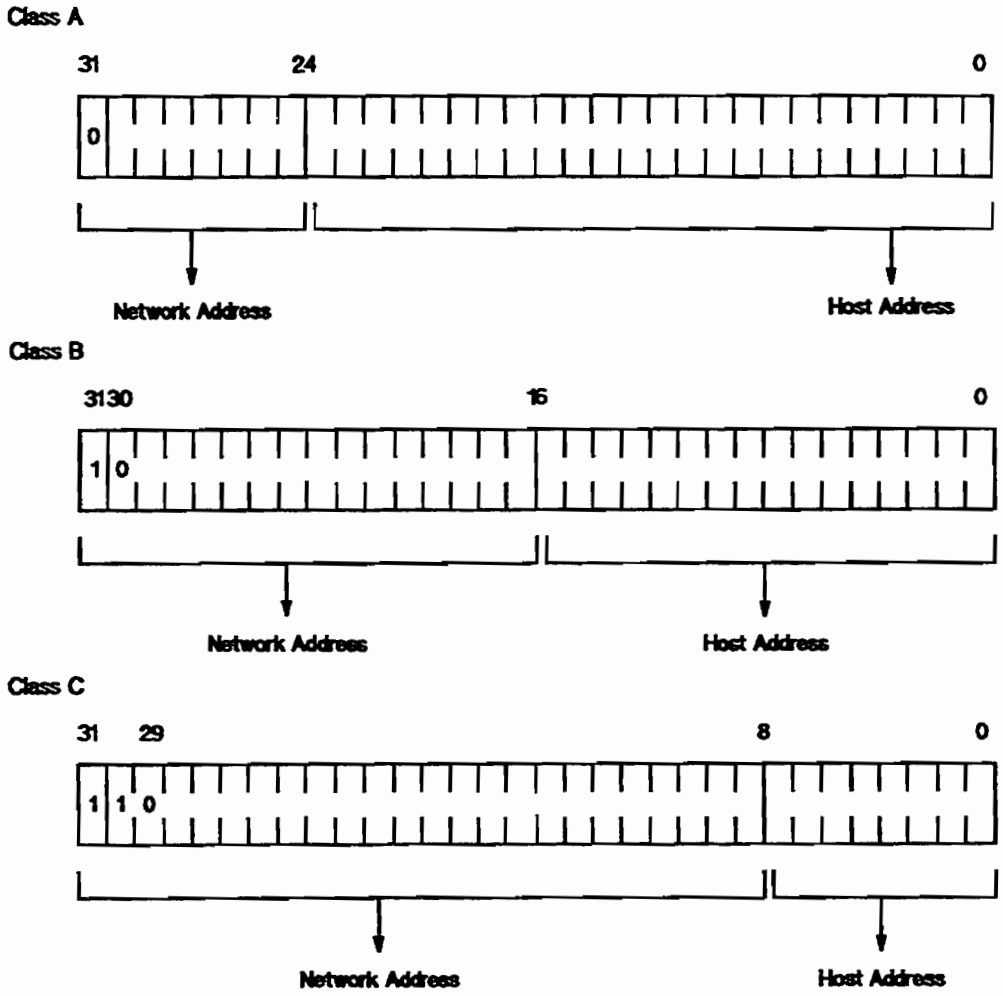

Figure 2-1. Internet Address Classes

The address classes can also be broken down by address ranges. Internet addresses are typically represented by converting the bits to decimal values an octet (8 bits) at a time, and separating each octet's decimal value by a period (.). Therefore, internet addresses are typically of the following form:

n.n.n.n

where *n* is a number from 0 to 255, inclusive. This form is referred to as *decimal dot notation* or *dot notation.*

The following table lists the number of networks and nodes and the address ranges for each address class.

Table 2-2. Internet Address Classes

Class	Networks	Nodes per Network	Address Range
A	127	16777215	0.0.0.1 – 127.255.255.254
B	16383	65535	128.0.0.1 – 191.255.255.254
C	2097151	255	192.0.0.1 – 223.255.255.254
Reserved	–	–	224.0.0.0 – 255.255.255.255

To determine a network address and host address from an internet address, you must separate the network and host address fields. For example, the bit representation of internet address 192.6.1.1 is separated as follows:

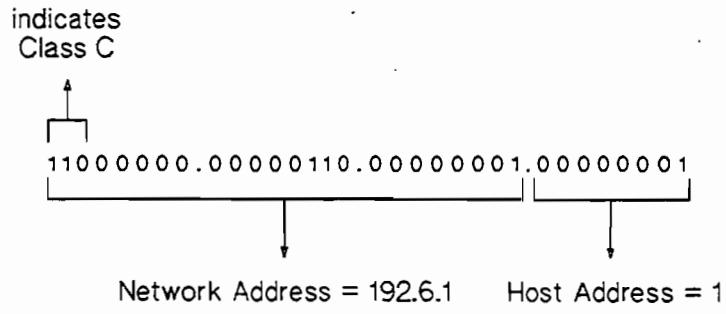

Figure 2-2. Bit Representation of IP Address

Assigning an Internet Address

Each node on the network has at least one internet address. When assigning internet addresses, you must determine network and host addresses as described in this section.

Note When specifying internet addresses, do **not** use leading zeroes within address segments. For example: 192.006.012.023 is **wrong**; 192.6.12.23 is **correct.**

Assigning Network Addresses

To assign network addresses, follow these rules:

- You must have a network address for each logical network.
- All nodes in a network must have the same network address.
- Do not assign any networks the network addresses 0 or 255 (Class A) or 255.255 (Class B) or 255.255.255 (Class C). Those addresses are reserved.

Note HP has obtained a block of Class C network addresses from DARPA to assign to HP customers. You can obtain Class C addresses that are unique within the ARPANET by contacting HP at the following address:

Network Administration Office
Information Networks Division
Hewlett-Packard Company
19420 Homestead Road
Cupertino, California 95014
(408) 447-3444

Assigning Host Addresses

Host addresses must be unique within each network. You can assign host addresses according to your own needs, but they must be within the range for the internet address class that you are using.

Note Do **not** assign any nodes the host addresses 0 or 255.255.255 (Class A) or 255.255 (Class B) or 255 (Class C); **these addresses are reserved.**

Subnet Addresses

Subnetting is an optional addressing scheme that allows you to partition the host address portion of an internet address into discrete subnetworks. The physical networks are connected via gateways. By doing this, several physical networks share the same network address to form one logical network.

For example, if you have a large installation with many interconnected nodes, you could run into hardware configuration restrictions or performance degradation if you tried to place all nodes on the same physical network. With subnetting you can install several smaller physical networks but have them all share the same network address.

Assigning Subnet Addresses

As described previously, an internet address can be represented as four fields separated by a period, each of which represents 8 bits of the overall address.

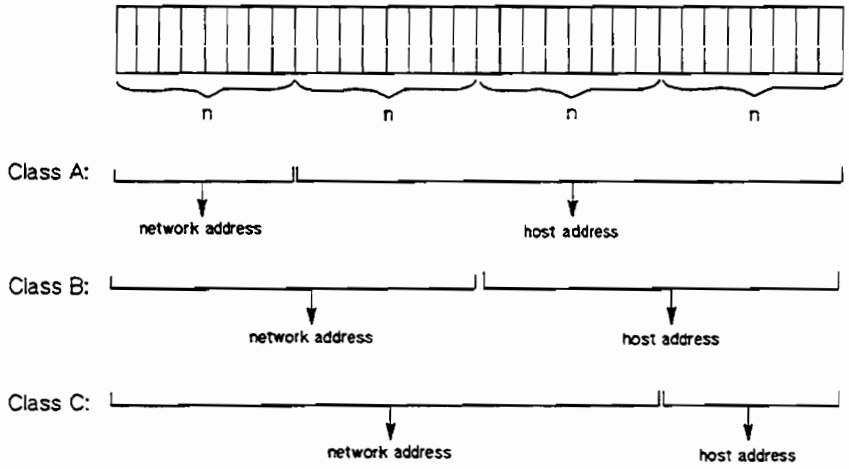

Figure 2-3. Internet Address Fields

Subnet addressing uses the host address portion of the internet address and subdivides it in a way that can accommodate a given number of subnetworks and a given number of nodes per subnetwork. The following rules apply when choosing a subnet addressing scheme and an internet address:

- All subnets on the same network must have the same network address.
- Do not assign a host address where all the bits are 0 or all the bits are 1 (255 base 10); this also implies you cannot have a subnet address of 0.

Using three of the eight bits of the host address portion of a Class C address, the following table lists the valid internet address ranges for up to 7 subnets and 29 nodes per subnet.

Class C internet address: n.n.n._____

Table 2-3. Subnet Addressing

Bitwise Subnet (binary)	Subnet Address (decimal)	Internet Address Range (decimal)
000xxxxx	0	(not allowed)
001xxxxx	1	n.n.n.33 - n.n.n.62
010xxxxx	2	n.n.n.65 - n.n.n.94
011xxxxx	3	n.n.n.97 - n.n.n.126
100xxxxx	4	n.n.n.129 - n.n.n.158
101xxxxx	5	n.n.n.161 - n.n.n.190
110xxxxx	6	n.n.n.193 - n.n.n.222
111xxxxx	7	n.n.n.225 - n.n.n.254 (255 is reserved)

Assigning Subnet Masks

Subnet addressing is implemented by specifying the keyword *netmask* and designating a 32-bit subnet mask in the *ifconfig* command when a LAN interface card is powered up with its internet address. All nodes on a network (with a given network address) must specify the same subnet mask.

The subnet mask is AND'd with the address attached to a message coming across the network to determine if that message should be routed to a node on the local network or ignored. The subnet mask to use with the subnet addresses in the previous table would be:

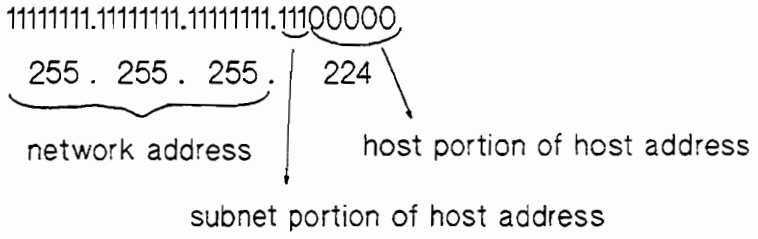

Figure 2-4. Subnet Mask

LAN Device Terminology

Following is a description of terms used by the I/O subsystem to identify LAN cards and device files associated with LAN cards. Become familiar with these terms before attempting LAN installation, administration and diagnostic procedures.

Hardware Path

On Series 600/800 computers, the I/O subsystem identifies each LAN card by its hardware path. The hardware path is assigned according to the physical location (slot) of the card in the hardware backplane. For CIO LAN cards, there are two parts to a hardware path. The first part, the module number, is determined by the location of the channel adapter on the system bus. The second number, the slot number, is determined by the slot number of the CIO LAN card on the CIO adapter module

For example, if you insert a CIO LAN card into one of several slots on a CIO adapter module, e.g. slot 3, and this CIO adapter module is located in one of the hardware modules (also known as slots) on the system bus, e.g. hardware module 4, then the hardware path of the CIO LAN card is 4.3. If you add a second LAN card to slot 7 of the same CIO adapter module, the hardware path of the second LAN card is 4.7.

An NIO LAN card station address is determined in a different way. To determine the address, you multiply the location of the system bus slot number by 4. For example, an NIO LAN card inserted into hardware module 8 would have a hardware path of 32.

For Series 700 systems, the hardware path is composed of three parts: an IO module identifier, a slot identifier, and a card functionality identifier. The module identifier for a Core IO card is always 2, the slot number for Core IO is always 0 and the last field is always 2 for Core IO and 0 for EISA. Thus the hardware path for a Core IO (LAN) card will always be 2.0.2. For add-on EISA cards the module ID is 4, the slot number is 1 through 4, and the card functionality ID is always 2. For example, the hardware path for an add-on EISA card in slot 3 would be 4.3.0.

You can use the *lanscan(1M)* command to display the hardware path of each LAN card that is bound successfully to the I/O subsystem when the system is booted-up.

Select Code

On the Series 300/400, the I/O subsystem identifies each LAN card by its select code. The select code is preset when the system is manufactured, but you can use the dip switch on the LAN card to reset it. The LAN card on a mother board usually has a select code of 21 (15 hex), but you can change the select code before booting the system to another value. When a system has multiple LAN cards, each LAN card has a different select code.

You can use the *lanscan(1M)* command or the *landiag(1M)* command to display the select code of each LAN card.

Device Logical Unit

The device logical unit (lu) is the logical identifier of individual devices within a larger grouping of devices of the same type. For instance, if you have three CIO LAN cards on a Series 835 computer, these cards are all the same type, and each of them has a unique logical unit, e.g. 0, 3 and 2 respectively. The logical unit is unique only within its own category.

The logical unit number of a LAN card is one level of abstraction above the hardware path and the select code. On the Series 600/800, the logical unit of a LAN card is assigned by the I/O subsystem immediately after the system is booted up. The logical unit value can range from 0 to 255. If the system has newly installed LAN cards, the logical unit numbers are assigned one by one according to the order in which each card is bound to the I/O subsystem. The system records each logical unit number that has been assigned to a hardware path.

If the system shuts down and some of the LAN cards are removed, the logical unit numbers assigned to them are still held in reserve when the system is booted up. As a result, depending on the history of the backplane configuration changes when the system was previously booted up, the logical unit number of the LAN cards on a system may not be consecutive. A missing logical unit number (e.g. 1 in the above example) implies that a LAN card was configured in a previous system boot-up and was removed at a later time from the current hardware configuration.

Note Prior to HP-UX release 8.0, the device lu was not assigned by the system. It was specified in the I/O statement in the *uxgen* input file.

On the Series 300/400, the system assigns the logical unit numbers to LAN cards according to the order of their select codes. If three LAN cards with select codes of 21, 27 and 29 are configured when the system is booted-up, the logical units will be 0, 1 and 2 respectively. The system does not recall cards assigned during a previous system boot-up. On the Series 700, the logical unit number is the same as the interface unit number.

You can use the *lanscan(1M)* and *landiag(1M)* commands to display the device logical unit assigned by the system to each LAN card that is successfully bound to the system when the system is booted-up.

LAN Device Files

You use LAN device files to directly access the LAN driver. A device file identifies the LAN card, the LAN driver, and the Data Link protocol (Ethernet or IEEE 802.3) to be used.

By convention, device files are kept in a directory called */dev*. Each device file has a name and a device number to uniquely identify the above characteristics.

The system follows certain conventions when creating LAN device files. The user must enter the correct major number and correct minor number but does not have to follow any other conventions when creating device files. Device files are used by Link Level Access users to access the LAN driver, and some network services and diagnostic tools. These files are described in the section on "Verifying LAN Device File Creation" in Chapter 3.

The device number is composed of a major number and a minor number. On the Series 600/800, the CIO LAN driver has a major number of 50, and the NIO LAN driver has a major number of 51. On the Series 300/400, the DIO LAN driver has a major number of 18 for IEEE 802.3 and 19 for Ethernet. On the Series 700, the LAN driver has a major number of 52.

The minor number is 24 bits wide and consists of various fields. On the Series 600/800, the most significant 8-bit field is not used (zero). The middle 8-bit field is the device logical unit that identifies the LAN card. The least significant 8-bit field indicates IEEE 802.3 (0) or Ethernet (1). After the system is booted up, the system creates two device files for each LAN card, one for IEEE 802.3 and one for Ethernet.

On the Series 300/400, the most significant 8-bit field of the minor number is the select code of the LAN card and the other two 8-bit fields are not used (zero).

In the following explanation of the minor number on Series 700 systems, bit 0 is the right-most bit while bit 23 is the left-most bit of a 24-bit word. The minor number on Series 700 workstations is constructed as follows:

Table 2-4. Series 700 Device File Bit Structure

Bits	Contents
Bits 23-20	Contains the IO module ID (2 for Core IO/4 for EISA)
Bits 19-16	Indicates the slot into which the card is plugged (0 for Core IO/1-4 for EISA)
Bits 15-12	Identifies the IO functionality supported by the card (2 for Core IO/0 for EISA).
Bits 11-1	These bits are always 0
Bit 0	Contains an encoded protocol bit (1 = Ethernet, 0 = IEEE)

You can create LAN device files with the *mknod(1M)* command.

3

Installing LAN

This chapter describes how to load and configure LAN/9000 software. It contains the following sections:

- Overview of Installation.
- Creating a Network Map.
- Connecting LAN Hardware.
- Selecting LAN Filesets.
- Loading LAN Software
- Verifying LAN Device File Creation.
- Configuring LAN Software Using SAM.
- Configuring LAN Software Manually.
- Rebooting the System.
- Verifying Installation.

Overview of Installation

Installation of LAN/9000 includes creating a network map, connecting LAN hardware, loading LAN filesets, configuring LAN software, and system reboot. In some cases, you may also need to rebuild the kernel.

To configure LAN software, you may use either of two methods:

- Automatic process using SAM.
- Manual process.

SAM is the acronym for System Administration Manager. It is a menu-driven utility for system administration tasks, including network configuration.

Note If LAN/9000 has been pre-installed on your system, you may skip the sections in this chapter on "Selecting LAN Filesets" and "Loading LAN Software." If, in addition, you have a pre-configured Series 700 workstation, you may also skip the section, "Configuring LAN software Using SAM." The Series 700 Core IO system contains a pregenerated and preconfigured kernel, a */usr/lib/uxbootlf* file and a copyright file. The kernel is pregenerated with all the proper drivers including the LAN driver.

Creating a Network Map

Be sure to create a network map or update the existing map before attempting installation. An accurate map is essential for administering the LAN. Such a map should include:

- Approximate dimensions of the building or room containing the LAN.
- Location of nodes and node connections.
- Host and NS node name of each node.
- Internet and Station Addresses of each node (in the case of gateways, each LAN card has its own set of Internet and Station Addresses).
- Select Code (Series 300/400 only) or Hardware Path (Series 600/800 and Series 700 only) of each LAN card.
- A list of any other cards contained in each node backplane.
- A note of distributed applications flow from the local node to/from other nodes in the network.
- Version number of the operating system installed on each node.

Figure 3-1 shows an example network map. Figure 3-2 shows a sample worksheet that can be used to gather information for Series 300/400 systems on the map. A worksheet for Series 600/800 and Series 700 systems should have Hardware Path as a column heading in place of Select Code.

Note The "HP OpenView Network Node Manager" product provides dynamic mapping of your network using a graphical user interface. The dynamic mapping facility goes out over your network to learn how it is configured, and the map created is automatically kept up-to-date. Contact your HP Sales Representative for more information on this product.

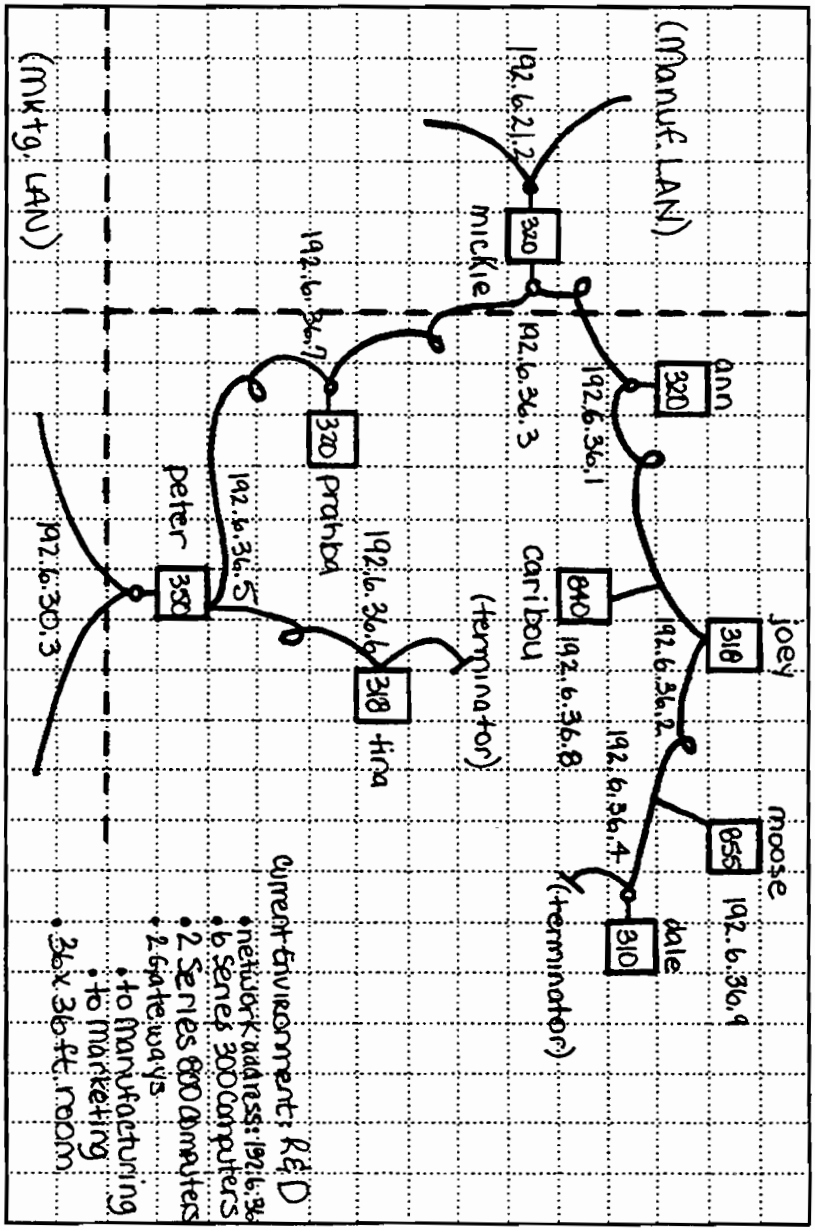

Figure 3-1. Network Map

Network Map Worksheet

ARPA Host Name	NS Node Name	Internet Address	Link Level Address	Select Code	Other Cards in Backplane	Applications	Operating System
dale	dale.lab.hp	192.6.36.4	080009 001484	21			PE 6.0
joey	joey.lab.hp	192.6.36.2	080009 004800	21			PE 6.0
ann	ann.lab.hp	192.6.36.1	080009 00A111	21			HP-UX 6.0
mickie.lab mickie	mickie.mfg.hp	192.6.36.3 192.6.21.2	080009002108 080009 0012A8	22 21	98643(2)		AXE 6.0
prahba	prahba.lab.hp	192.6.36.7	080009 005201	21			PE 6.0
peter peter.mktg	peter.lab.hp	192.6.36.5 192.6.30.3	080009000140 080009 001024	21 22	98643(2) (1 built-in)		HP-UX 6.5
tina	tina.lab.hp	192.6.36.6	080009 001001	21			PE 6.5
moose	moose.lab.hp	192.6.36.8	080009 002125	—			HP-UX 7.0
caribou	caribou.lab.hp	192.6.36.9	080009 001402	—			HP-UX 7.0

Figure 3-2. Network Map Worksheet

Connecting LAN Hardware

If you have Series 600/800 hardware, this chapter assumes that LAN/9000 hardware has been installed and verified. For information about hardware installation, refer to the following:

- *HP Precision Bus Local Area Network Interface Controller (LANIC) Installation and Configuration Guide* (for Model 815).
- *LAN Interface Controller (LANIC) Installation and Reference Manual* (for all other Series 600/800 Models).
- *LAN Cable and Accessories Installation Manual.*

If you have Series 300/400 or Series 700 hardware, you connect to the LAN in either of two ways, depending on your Series 300/400 or Series 700 model number and the options you purchase:

- Direct to ThinLAN cable using a BNC T-connector.
- To Backbone MAU, ThinMAU or Ethertwist MAU using an AUI cable.

The possible connections are shown in Figure 3-3.

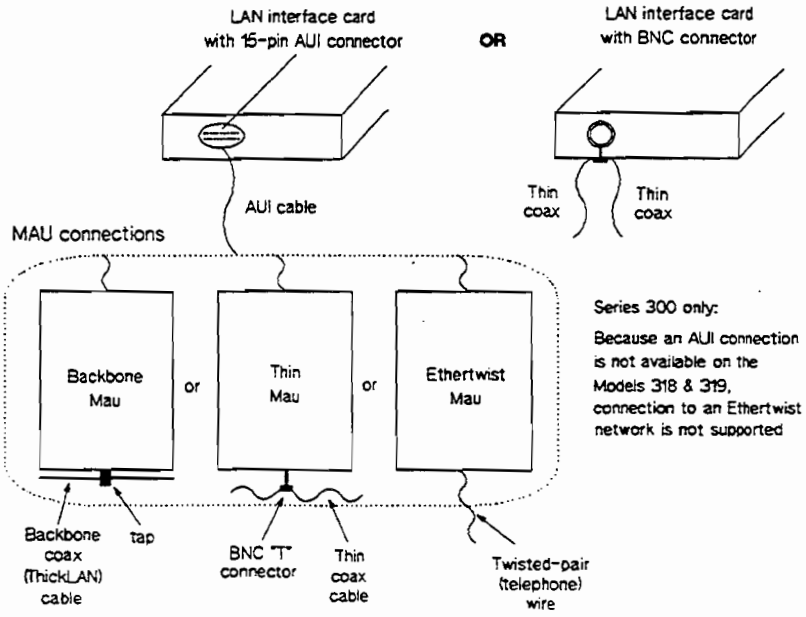

Figure 3-3. LAN/9000 S300/400 and S700 Connections

Note **Series 700 only:** If you add an EISA card(s) to your Series 700 workstation, you must run the *eisa_config* utility prior to installing the board. Refer to the *E/ISA Configuration Guide for HP-UX* for more details.

Selecting LAN Filesets

Note If LAN/9000 software has been preinstalled on your system, skip this section and the section on "Loading LAN Software" and go to "Configuring LAN Software Using SAM."

Prior to running the Update program to load the LAN/9000 software on your system, you must decide whether you want to load ALL networking products or only those products necessary for your configuration. If you decide to load a select group and save space, you must go into the networking partition in the *update* program and select the filesets within the networking partition that are appropriate for your configuration.

Refer to Appendix F for a table showing the correspondence between filesets and subsystem names. Refer to the following two sections in this chapter for more detailed information on using *update* to load files.

You may select from the filesets listed below:

BSDIPC-SOCKET	Unix Domain and Berkeley IPC header files; and Unix Domain and Berkeley IPC kernel code, libraries, and commands.
NETIPC	NetIPC header files and demo programs; NetIPC kernel code, libraries, and commands; and NetIPC MAN pages.
NETINET	Internet header files and demo programs (application development); Internet commands (ping) and kernel code; MIB kernel code; and SLIP/PPP MAN pages and commands.
NET	Network support header files; Network support commands and kernel support; and MAN pages for network support commands and libraries.
LAN	Link Level Access header files and LAN maintenance and diagnostic commands; driver support for Diskless Unix; kernel support for LAN drivers and LLA; and LAN driver MAN pages.

NETTRACELOG	Network tracing and logging support.
APPLTALK	AppleTalk kernel support.

Networking Software	Required Filesets
HP-UX	None required.
Unix Domain Sockets	BSDIPC-SOCKET
IP/TCP/UDP	BSDIPC-SOCKET NETINET NET NETTRACELOG LAN and/or X.25
NETIPC	BSDIPC-SOCKET NETIPC NETINET NET NETTRACELOG LAN and/or X.25
DUX or LLA	BSDIPC-SOCKET NETINET NET LAN NETTRACELOG
AppleTalk	BSDIPC-SOCKET NET LAN NETTRACELOG APPLTALK
PPL (SLIP)	BSDIPC-SOCKET NETINET NET NETTRACELOG

Loading LAN Software

Note If LAN/9000 has been pre-installed on your system, you may skip this section and go to "Configuring LAN Software using SAM."

Note Before you attempt these procedures, use the *uname -a* command to check that the correct version of the HP-UX operating system is installed. The HP-UX version number must match that of the LAN/9000 software to be loaded.

You use the HP-UX *update* program to install the LAN/9000 software. The *update* program is fully documented in the *Installing and Administering HP-UX* manual. Read about *update* before attempting this procedure.

If you are configuring your node for real-time use, refer to the "Installing for Real-Time Use" subsection for procedures on how to customize the *S800* or *dfile* file and change the *netisr* priority.

Using update

Follow this procedure if you are adding LAN/9000 to a HP-UX system for the first time.

1. Run the *update* program in interactive mode.
2. Select and load the appropriate LAN/9000 filesets. (See "Selecting LAN Filesets" in this chapter.) Your LAN fileset version must match the version of your HP-UX operating system. *update (1M)* loads the filesets, runs the customized scripts for the filesets, and builds the kernel.

   If the kernel build is not successful, the *update (1M)* program escapes to a new shell. Note the cause of the failure at the end of */tmp/update.log*.

3. If the kernel build was unsuccessful, correct the problem and press Cntrl D to return to the *update* program.

   Update (1M) attempts to build the kernel again and continues to retry until the kernel build is successful or the *update* process is aborted. When the kernel build is successful, *update (1M)* loads the remaining filesets, and reboots the system.

   The system runs the remaining customized scripts, reboots, and prompts for a login.

4. Log in as root, check for error messages at the end of */tmp/update.log*, and refer to Appendix A, "Installation Error Messages" in this manual to correct any unresolved problems.

5. If *update* installs the LAN/9000 software successfully, proceed to "Verifying LAN Device File Creation."

6. Configure and run the */etc/netlinkrc* initialization script to start networking. Refer to "Editing and Installing /etc/netlinkrc" in this chapter.

Note If you previously installed LAN/9000 and have existing LAN configuration files in */etc* and */usr/adm* directories, the *update* program will copy the new configuration files to */etc/newconfig*. You should compare your current files to the new files in */etc/newconfig* to ensure that your configuration information is current.

Creating a New Kernel (Series 600/800)

This procedure applies if you wish to create a new kernel manually. Alternatively, you can also create a new kernel by running SAM.

Note Before attempting this procedure, familiarize yourself with the system reconfiguration information in the *uxgen(1M)* manual reference page and HP-UX system literature.

---

1. Move to the */etc/conf/gen* directory:

   ```
   cd /etc/conf/gen
   ```

2. Edit the *uxgen* input file, usually called *S800*, by:

   - Copying the existing file and saving it.
   - Removing the comment delimiters /* and */ from the following lines of the *uxgen* input file according to the filesets that you have loaded. If you are creating a CIO-based kernel, you must remove the comment delimiters from `/*include lan0 */`. If you are creating an NIO-based kernel, you must remove the comment delimiters from `/*include lan1 */`. Refer to Appendix F for a listing of filesets and corresponding subsystem names. The file, *netdiag1*, will always be in the *uxgen* input file by default.

     ```
     /*include uipc;*/
     /*include nipc;*/
     /*include inet;*/
     /*include ni;*/
     /*include atalk;*/
     /*include nm;*/
     /*include nfs;*/
     /*include lan */
     /*include lan0 */
     /*include lan1 */
     ```

3. Create a new kernel:

   ```
   uxgen S800
   ```

4. Move to the *../S800* directory:

   ```
   cd ../S800
   ```

5. Save the old kernel:

   ```
   mv /hp-ux /SYSBCKUP
   ```

6. Move the new kernel to the root directory:

   ```
   mv hp-ux /
   ```

7. Check file system integrity:

   ```
   sync; sync
   ```

8. Reboot the system:

   ```
   reboot
   ```

Installing for Real-Time Use

The following applies if you are configuring the node for real-time use.

In each case, you need to edit the *uxgen* input file.

1. Use *update* to load LAN/9000 Series 600/800 files as described previously in "Using update to Load Files on the Series 600/800."

2. Edit the *uxgen* input file as required for your application. For details, refer to "Editing the uxgen Input File for Real-Time Operation" in the next section.

3. After the *uxgen* input file edits, follow Steps 3 through 8 of the previous subsection, "Creating a New Kernel (Series 600/800)." This creates the new kernel and reboots the system.

Editing the uxgen Input File for Real-Time Operation

Note Perform this step only if you plan to use your LAN/9000 Series 600/800 product for real-time operation.

The network interface daemon, *netisr*, is a packet dispatcher between the LAN driver and the IP protocol layer. By default, the *netisr* daemon runs as an interrupt at priority -1 on both the Series 300/400, Series 700 and the Series 600/800. Running *netisr* as an interrupt substantially improves the throughput and performance of network related software. To run *netisr* as a process, you must reset its priority to a value between 0 (high) and 127 (low) inclusive. You can edit the *netisr_priority* entry in the *uxgen* input file to change *netisr's* priority. Once the LAN product is installed, you may alter *netisr's* priority using the HP-UX *rtprio(1M)* command.

Note *netisr* must run at higher priority than other network services on the same node.

Caution Care must be taken when using the *rtprio(1M)* command to set the real-time priority of other processes ahead of netisr. Doing so may shutdown the system. Refer to the *HP-UX Real-Time Programming Manual* for more information on choosing real-time priorities.

Creating a New Kernel (Series 300/400 and Series 700)

This step is necessary only if you are not using SAM to configure and initialize LAN files and either of the following is true:

- LAN/9000 has not been configured previously on your system.
- LAN/9000 has been configured previously, but you are adding LAN cards which bring the total to three or more.

If you are using SAM, skip this step. Similarly, if you are not using SAM, and neither of the above two conditions is true, skip this step.

Using Existing dfile

If your kernel is highly customized for special system requirements, you are installing LAN/9000 only, or both, use the existing *dfile* to rebuild the kernel.

1. Using the */etc/shutdown* command, put the system in single-user mode. For details on this procedure, refer to the *Installing and Updating HP-UX* manual.
2. Save the old *dfile*:

   ```
   cp dfile  dfile.save
   ```

3. Enter:

   ```
   vi dfile
   ```

4. If you have not installed LAN/9000 previously, add the following lines to the *dfile*:

   ```
   lan01
   lla (Series 300/400 only)
   ```

5. Depending on which filesets you have loaded, you will also need to add the following lines to the *dfile*:

```
uipc
nipc (optional)
inet
netman
ni
dskless (optional)
nfs (optional)
netdiag1
```

Refer to Appendix F for a table showing the correspondence between filesets and subsystem names. Neither *nfs*, *nipc*, or *dskless* are covered in this manual. Refer to *HP-UX System Administration Tasks* for the Series 300, Chapter 10, for more information on *dskless*, and *Using NFS Services* for more information on *nfs*.

6. **Series 300 only:** If you want to add a third or more LAN cards to your system (up to a total of five), add the following line:

```
num_lan_cards n
```

where n is the total number of LAN cards to be supported by the kernel.

7. Save the new *dfile*.

8. Create the kernel configuration files, *conf.c* and *config.mk*:

```
/etc/config dfile
```

9. Using *config.mk*, create the new kernel file, */etc/conf/hp-ux*:

```
make -f config.mk
```

10. Save the old kernel:

```
mv /hp-ux  /SYSBCKUP
```

11. Move the new kernel to the root:

```
mv  hp-ux /hp-ux
```

12. Reboot on the new kernel:

```
/etc/reboot
```

Using dfile.full.lan

If you are installing other networking software as well as LAN/9000, use *dfile.full.lan* to rebuild your kernel. It contains additional lines necessary to update your system for all networking products.

1. Using the */etc/shutdown* command, put the system in single-user mode. For details on this procedure, refer to the *Installing and Updating HP-UX* manual.
2. Save the old *dfile*:

```
cp dfile  dfile.save
```

3. Enter:

```
cp  dfile.full.lan  dfile
```

4. Enter:

```
vi dfile
```

5. Edit *dfile* to include any customization from your previous *dfile* (now *dfile.save*).
6. **Series 300 only:** If you want to add a third or more LAN cards to your system (up to a total of five), add the following line:

```
num_lan_cards n
```

   where n is the total number of LAN cards installed.

7. Save the new *dfile*.
8. Perform Steps 7 through 11 of "Using the Existing dfile."

Verifying LAN Device File Creation

Once your system is rebooted, log on and use the *lanscan(1M)* command to find the device logical unit number (lu) of each LAN card. Refer to Chapter 6 for a detailed explanation of the *lanscan(1M)* command.

Device Files on the Series 600/800

For each LAN card that is bound successfully to the I/O subsystem at boot-up, two device files, */dev/lanx* and */dev/etherx*, are created by the system with x the device lu of each card.

Example 1:

For a system with three CIO LAN cards with device lu numbers of 0, 2, and 3, you issue the command `ls -l /dev/lan0 /dev/lan2 /dev/lan3 /dev/ether0 /dev/ether2 /dev/ether3`

The display will be as follows:

```
crw-rw-rw-   1  bin   bin  50 0x000000  Jan 28 08:58  /dev/lan0
crw-rw-rw-   1  bin   bin  50 0x000200  Jan 28 08:58  /dev/lan2
crw-rw-rw-   1  bin   bin  50 0x000300  Jan 28 08:58  /dev/lan3
crw-rw-rw-   1  bin   bin  50 0x000001  Jan 28 08:58  /dev/ether0
crw-rw-rw-   1  bin   bin  50 0x000201  Jan 28 08:58  /dev/ether2
crw-rw-rw-   1  bin   bin  50 0x000301  Jan 28 08:58  /dev/ether3
```

The fifth column is the major number (50 for the CIO LAN driver). The sixth column is the minor number consisting of three two-digit fields. The middle two-digit field is the device lu number, and the last two-digit field indicates the Data Link protocol (0 for IEEE 802.3 and 1 for Ethernet).

Example 2:

For a system with two NIO LAN cards with lu numbers of 1 and 2, you issue the command `ls -l /dev/lan1 /dev/lan2 /dev/ether1 /dev/ether2.`

The display should be as follows.

```
crw-rw-rw-   1  bin   bin  51 0x000100  Jan 28 08:58  /dev/lan1
crw-rw-rw-   1  bin   bin  51 0x000200  Jan 28 08:58  /dev/lan2
crw-rw-rw-   1  bin   bin  51 0x000101  Jan 28 08:58  /dev/ether1
crw-rw-rw-   1  bin   bin  51 0x000201  Jan 28 08:58  /dev/ether2
```

The fifth column is the major number (51 for the NIO LAN driver). The sixth column is the minor number consisting of three two-digit fields. The first two-digit field of the minor number is always 0. The middle two-digit field is the device lu number, and the two-digit field indicates the Data Link protocol (0 for IEEE 802.3 and 1 for Ethernet).

Device Files on the Series 300/400

After boot-up, the system creates three LAN device files with select code 21 (15 hex). If the select code of the on-board LAN is different than 21, you will have to remove these files and create new ones with the proper select code. The system does not create device files for add-on LAN cards until you run SAM.

Example 1:

For any system with one or more DIO LAN cards, issue the command `ls -l /dev/lan /dev/ieee /dev/ether`.

The display will be as follows.

```
crw-rw-rw-   1 bin   bin  18 0x150000  Jan 28 08:58  /dev/lan
crw-rw-rw-   1 bin   bin  18 0x150000  Jan 28 08:58  /dev/ieee
crw-rw-rw-   1 bin   bin  19 0x150000  Jan 28 08:58  /dev/ether
```

The fifth column is the major number (18 for DIO LAN driver using the IEEE 802.3 protocol and 19 for DIO LAN driver using the Ethernet protocol). The sixth column is the minor number. The first two-digit field in the minor number is the select code (21 or 15 hex) and the other two two-digit fields are always 0.

After boot-up, if you run SAM to configure LAN software, SAM will find the select code and the lu(x) of each LAN card, and create three device files, */dev/lanx*, */dev/ieeex*, and */dev/etherx*, for you.

These device files are used by Link Level Access (LLA), by the *rbootd(1M)* command for Diskless, and by the *landiag(1M)* and *LANDAD* commands for LAN diagnostics.

If the major numbers, minor numbers, or device file names are not correct, delete the device file entries from your */dev* directory and recreate them with the correct numbers using the *mknod(1M)* command.

Device Files on the Series 700

After boot-up, the system creates three LAN device files by default: `/dev/lan0`, `/dev/ieee0`, and `/dev/ether0` where `1282 (0x202)` corresponds to the most significant 12 bits of the minor number of the Core IO Lan card. The device LU number is concatenated to the device files.

The display will be as follows:

```
crw-rw-rw-   1  bin   bin  52 0x202000  Mar 14  1990   /dev/lan0
crw-rw-rw-   1  bin   bin  52 0x202000  Mar 14  1990   /dev/ieee0
crw-rw-rw-   1  bin   bin  52 0x202001  Mar 14  1990   /dev/ether0
```

Configuring LAN Software Using SAM

Once you have installed LAN software, you can use SAM to automatically configure networking.

SAM stands for System Administration Manager, a menu-driven utility for system administration tasks, including configuration of networking software.

Note Using SAM is the preferred method for LAN/9000 configuration. However, SAM currently does not support domain-style naming of your system. Domain-style naming is used with the BIND name service provided with ARPA Services/9000. If you are using the BIND name service, you must configure LAN/9000 manually. Skip to the following section, "Configuring and Initializing LAN Manually."

Tips for Using SAM

When using SAM, remember the following:

- Use your keyboard's cursor control and editing keys to navigate and edit forms.
- You may select a menu item by typing enough of its first word to uniquely identify it. In some cases, this is simply the first letter of the menu item. This method does not work for menu items that start with the same word.
- Access the on-line help screens whenever you need more information, such as how or where to obtain a required configuration value. Note that the RESULT section of the on-line help screens explains what SAM does "behind the scenes," such as what files SAM creates or modifies, or what commands SAM executes automatically.

Using SAM, configuring LAN can be divided into two procedures:

- Configuring LAN hardware.
- Configuring Network Connectivity.

Configuring LAN Cards

1. At the HP-UX prompt, type:

   `sam`

   and wait for SAM's main menu to appear.

2. Select the `Networks/Communications` menu item.

3. Select the `LAN Hardware and Software (Cards and Services)` menu item.

4. Select the `Add a New LAN Card` menu item or the `View/Modify a LAN Card's Configuration`, except for the S700 Core IO card, (under `Networks/Communications`).

5. Fill in the form according to instructions. View the help screens for information about filling in the form.

6. Press the `Perform Task` softkey. Note that, before you exit SAM entirely, SAM automatically does what is necessary to configure your LAN card. You do not need to configure the card via SAM's other LAN card menu options.

7. Repeat steps 5 and 6 to configure additional LAN cards.

8. Press the `Main Menu` softkey when you are finished.

9. To configure your system for network communication, continue with the steps provided in the following subsection, "Configuring Network Connectivity." If you wish to stop after configuring the LAN card(s), use the `Exit SAM` softkey to exit SAM. If you are adding new LAN cards, you may be told that SAM needs to reconfigure a new kernel and reboot your system to activate the new configuration.

Configuring Network Connectivity

Your system will not be able to communicate with other systems until you do the following additional steps:

- Add entries for remote systems to your system's */etc/hosts* file.
- If you need to use gateways, you must add */etc/route* entries for them to your */etc/netlinkrc* initialization script.

Note You need to have ARPA Services/9000 or NFS Services/9000 installed for the following steps.

You can use SAM to do each of these tasks automatically.

1. At the Main Menu, select the `Networks/Communications` menu item.
2. Select the `LAN Hardware and Software (Cards and Services)` menu item.
3. Select the `ARPA Services Configuration` or the `NFS (Network File System) Configuration` menu item.
4. Select the `Add/Modify Connectivity Info About a Remote System` menu item.

Note The `View/Remove Connectivity Info About a Remote System` menu item lets you view or delete */etc/hosts* file entries. If you had to reach the remote system through a gateway, this menu item also removes the associated */etc/route* command from the */etc/netlinkrc* file.

---

5. Fill in the form according to its instructions. View the help screens for information about filling in the form.
6. Press the `Perform Task` softkey.

Note If you must go through a gateway to reach the remote system to which you are adding connectivity, SAM prompts you for the gateway's hostname and IP address. With this information, SAM automatically configures the necessary routing (by executing an */etc/route* command and adding it to the */etc/netlinkrc* file) to reach that remote system through a gateway.

If there is just one gateway you use to reach many or all systems on other parts of the network, use the `Specify the Default Gateway` form (under the ARPA `Services Configuration` menu) to avoid having to enter the same gateway information every time SAM prompts you for it.

7. Repeat Steps 5 and 6 to add connectivity information about more remote systems.
8. Press the `Main Menu` softkey when you are finished.
9. Press the `Exit SAM` softkey to exit from SAM. If you have not previously initialized LAN card configuration, you may be told to use SAM to reconfigure a new kernel and reboot your system to activate the new configuration.

Verifying Remote Systems

To view the list of remote systems you can communicate with, type the following command at the HP-UX prompt:

```
more  /etc/hosts
```

To view the destinations reached through gateways and the gateways used to reach those destinations, type the following command at the HP-UX prompt:

```
netstat -r
```

The listing from this command may appear slowly, as it attempts to find the names associated with the network addresses used to perform routing.

To verify that you can communicate with a remote system via the LAN/9000 product, refer to the "Verifying Installation" subsection at the end of this chapter.

Undoing: Deleting a Default Gateway

To delete a default gateway that you have added with SAM's Specify a Default Gateway form, do the following:

1. Enter the following command at the HP-UX prompt:

   ```
   /etc/route delete default gateway_hostname
   ```

   where *gateway_hostname* is the name of the default gateway you want to delete.

2. Edit the */etc/netlinkrc* file to remove the corresponding */etc/route* add default entry for the gateway.

Configuring LAN Software Manually

If you are not using SAM, do the following to configure and initialize LAN:

- Create */etc/hosts*.
- Edit and install */etc/netlinkrc*.

You may also do the following optional steps:

- Create */etc/networks*.
- Modify */etc/services*.
- Modify */etc/protocols*.

Creating the /etc/hosts File

The */etc/hosts* file associates IP host addresses with mnemonic host names and alias names. It contains the names of other nodes in the network with which your system can communicate. LAN/9000 diagnostics *netstat* and *ping* use */etc/hosts*. If you install ARPA Services/9000 or NFS/9000, those products also use the */etc/hosts* file.

You can create an */etc/hosts* file three ways:

- From scratch, entering the known nodes in the format shown below.
- By copying the file from another node.
- If you are installing ARPA Services/9000, you may copy the official host data base maintained at the Network Information Control Center (NIC) for ARPA Internet networks. (Refer to "Military Standards and Request for Comments Documents" section of Chapter 1 for more information on how to contact the NIC.)

If you copy an */etc/hosts* file from another host, you may need to bring it up-to-date by adding unofficial aliases or unknown hosts, including your own host.

/etc/hosts

Each node in the */etc/hosts* file has a one line entry. Each entry in the file must be in the following form:

Syntax

```
IP_address host_name[alias(es)]
```

Parameters

IP_address	The IP address that uniquely identifies the node. Refer to Chapter 2 for details on IP addresses. *IP_address* must be in internet "dot" notation.
host_name	Name of the node. Host names can contain any printable character except spaces, newline, or the comment character (#). Naming Convention: The first nine characters should be unique for each network host.
alias(es)	Common name or names for the node. An *alias* is a substitute for *host_name*. *Alias* names are optional. Naming Convention: The first nine characters should be unique for each network host.

Note HP recommends that the host name should be the same as the node field assigned by the NS *nodename* command. (You assign a node name to your system when you initialize */etc/netlinkrc*, as described later in this chapter.) If you are using domain-style naming, try to keep the various types of names assigned to your system as consistent as possible, within their limitations.

/etc/hosts Format

When creating the */etc/hosts* file, follow these rules:

- Lines cannot start with a blank or tab character.
- Fields can have any number of blanks or tab characters separating them.
- Comments are allowed; they are designated by a pound sign (#) preceding the comment text.
- Trailing blank and tab characters are allowed.
- Blank line entries are allowed.
- Only one host entry per line is allowed.

/etc/hosts Permissions

The */etc/hosts* file should be owned by user *root*, group *other*, and have 0444 (-r—r—r—) access permission. For more information on */etc/hosts*, refer to the hosts(4) entry in the *LAN/X.25 Reference Pages*.

/etc/hosts Example

The */etc/hosts* entry for a node with:

- The IP address *192.6.1.1*.
- The node field of an NS hostname *node3*.
- The alias name *grace*.

looks like:

```
192.6.1.1 node3 grace
```

Editing and Installing /etc/netlinkrc

To configure and initialize LAN manually, you must also edit and install the LAN/9000 initialization script, */etc/netlinkrc*. To do so, you must be logged on as super-user. Once edited and installed, the */etc/netlinkrc* script does the following when you reboot:

- Starts network logging.
- Configures the network interface with an IP address.
- If the NetIPC fileset has been loaded, assigns a network (NS) node name to be used by *rlb*(1M) and NS/9000.
- If the NetIPC fileset has been loaded, starts the *rlb(1M)* and NetIPC daemon, *rlbdaemon*, respectively.
- Configures the network routing table if your node is a gateway or on a LAN with a gateway.
- Starts the Internet daemon (*inetd*).
- Starts NFS/9000 (if it is installed) by invoking the NFS initialization script */etc/netnfsrc*.
- Starts ARPA Services/9000 (if it is installed) by invoking the */etc/netbsdsrc* initialization script.
- Starts NS/9000 (if it is installed) by invoking the */etc/netnssrc* initialization script.

Note You must initialize LAN/9000 (reboot with */etc/netlinkrc* installed) to use NFS/9000, ARPA Services/9000 or NS/9000.

Note You use the *lanconfig* command to change the default encapsulation. Refer to *lanconfig* in the next section for a detailed description of this command.

Editing /etc/netlinkrc

Before installing */etc/netlinkrc*, edit it to contain the following information:

- The network interface name that identifies your LAN card.
- The IP address that identifies your system as part of the network.
- Network routing table information if your node is on a LAN with a gateway.
- The NS node name that is assigned to your system.

Entering this information is described in following subsections.

Assigning a Network Interface Name and IP Address

To assign a network interface name and IP address, you edit the */etc/ifconfig IFCONFIG_OPTIONS* entry in the */etc/netlinkrc* script. A detailed explanation of */etc/ifconfig* is provided in Chapter 4. Following is a sample */etc/netlinkrc* entry:

```
/etc/ifconfig lan0 192.6.21.2
```

where lan0 is interface name, and 192.6.21.2 is IP address.

Note If you configure your system as a gateway, you must include one */etc/ifconfig* entry for each LAN interface (LAN card). Each entry must have a separate interface name and IP address.

Adding Entries to the Routing Table

If you intend to use your system as a gateway, or to communicate with gateways, you must edit the */etc/route* entry in the */etc/netlinkrc* script.

Note This step is required only if your node is a gateway or you intend to use gateways from your node. If you have no gateways on your network, leave this entry commented out and go on to the next step, "Assigning a Node Name."

Note When the LAN/9000 software is loaded, the only entry in the routing table is the loopback interface, called *lo0*. This corresponds with the *loop* entry in the */etc/networks* file. When the software is initialized, other entries are created for each LAN card installed: lan0, lan1, etc. Before adding additional entries to the routing table, you should contact your HP representative for supported gateway configurations.

To add entries to the network routing table, remove the comment delimiter (#) from the */etc/route ROUTE_OPTIONS* entry, then edit this entry and create other entries for each route you intend to use.

A detailed explanation of */etc/route* is provided in Chapter 4. Following is a sample /etc/netlinkrc entry:

```
/etc/route  add  192.6.12.33  196.6.12.132
```

where 192.6.12.33 is the IP address of the destination network, 196.6.12.132 is the IP address of the gateway to that destination.

Assigning an NS Node Name

To assign an NS node name to your system, remove the comment delimiter from the */bin/nodename* entry in */etc/netlinkrc*.

When assigning a node name, follow these rules:

- The node name must be in the form *node.domain.organization*. The three fields must be separated by periods and each field can contain up to 16 alphanumeric characters plus underscores and dashes (hyphens). The first character of each field must be alphabetic. The *domain* and *organization* fields are arbitrary labels that can be useful for grouping nodes and collections of nodes.
- The node name you assign must be unique on the network.

For more information on node names, refer to Table 2-1. Following is a sample entry:

```
/bin/nodename hpindda.ind.hp
```

where hpindda.ind.hp is the assigned node name.

You can use SAM to assign, view, or modify the NS node name of your system.

1. At the Main Menu, select the `Networks/Communications` menu item.
2. Select the `LAN Hardware and Software (Cards and Services)` menu item.
3. Select the `View/Modify This System's NS Node Name menu item.`
4. Fill in the form according to instructions. View the help screens for information about filling in the form.
5. Press the `Main Menu` softkey.

Note HP recommends that the node field of your node name is the same as the host name that you configured in the */etc/hosts* configuration file. If you are using domain-style naming, try to keep each of these names as consistent as possible, within their limitations.

Installing /etc/netlinkrc

Once you have edited the */etc/netlinkrc* script, you must install it. To install the script, edit the */etc/rc* file to add the following line after the line containing /etc/cron:

```
/etc/netlinkrc
```

Note Your */etc/rc* file must also contain a line for your system's host name. If the line is not there, edit */etc/rc* to include it. For details on */etc/rc*, refer to your HP-UX system reference manuals.

Activating Optional Network Features

To activate special network features, you may also want to configure */etc/networks, /etc/services* and */etc/protocols*. Each of these steps is optional.

Creating the /etc/networks File

The */etc/networks* file associates network addresses with mnemonic names and alias names. The */etc/networks* file contains the name and address of known internet networks with which your host can communicate. The LAN/9000 diagnostic *netstat* and the *route* command use the */etc/networks* file. You must configure this file for your host if you want *route* or *netstat* to use symbolic network names instead of addresses.

You can create an */etc/networks* file three ways:

- From scratch, entering the known nodes in the format shown below.
- By copying the file from another node.
- If you are installing ARPA Services/9000, you may copy the official host data base maintained at the Network Information Control Center (NIC) for ARPA Internet networks. (Refer to "Military Standards and Request for Comments Documents" in Chapter 1 for more information on how to contact the NIC.)

If you copy an */etc/networks* file from another host, you may need to bring it up to date by adding unofficial aliases or unknown networks, including your own network.

/etc/networks

Each network has a one line entry in the */etc/networks* file. Each entry in */etc/networks* file takes the following form:

Syntax

```
network_name network_address [alias(es)]
```

Parameters

network_name	Name of the internet network. Network names can contain any printable character except spaces, newline, or the comment character (#).
network_address	Network address that uniquely identifies the network. *network_address* must be in dot notation. See Chapter 2 for details on network addresses.
alias(es)	Common name or names for the network. An *alias* is a substitute for *network_name*. *Alias* names are optional.

/etc/networks Format

- Lines cannot start with a blank or tab character.
- Fields can have any number of blanks or tab characters separating them.
- Comments are allowed; they are designated by a pound sign (#) character preceding the comment text.
- Trailing blank and tab characters are allowed.
- Blank line entries are allowed.
- Only one entry per line is allowed.

/etc/networks Permissions

The */etc/networks* file should be owned by user *root*, group *other*, and have 0444 (-r—r—r—) access permission.

For more information on */etc/networks*, refer to the *networks(4)* entry in the *LAN/X.25 Reference Pages*.

/etc/networks Example

The */etc/networks* entry for a node with:

- The network name *neta*.
- The network address *192.6.1*.
- The alias name *testlan*.

looks like:

```
neta 192.6.1 testlan
```

Modifying the /etc/services File

The */etc/services* file associates port numbers with mnemonic service names and alias names. The */etc/services* file contains the names, protocol names, and port numbers of all services known to your local host. The *netstat* diagnostic uses the */etc/services* file.

If you install ARPA Services/9000 or NFS/9000, those products also use the */etc/services* file.

Note You can modify this file if you have special requirements, but it is properly configured when you receive LAN/9000.

/etc/services

Each service has a one line entry in the */etc/services* file. Each entry in */etc/services* file takes the following form:

Syntax

```
service_name port_num/protocol [alias(es)]
```

Parameters

service_name	Name of the service. Service names can contain any printable character except spaces, newline, or the comment character (#).
port_num/protocol	*port_num* is the protocol port number assigned to this service. All requests for this service must use this port number. *protocol* is the protocol name, as listed in */etc/protocols*, that the service uses.
alias(es)	Common name or names for the service. An *alias* is a substitute for *service_name*. *Alias* names are optional.

/etc/services Format

- Lines cannot start with a blank or tab character.
- Fields can have any number of blanks or tab characters separating them.
- Comments are allowed; they are designated by a pound sign (#) character preceding the comment text.
- Trailing blank and tab characters are allowed.
- Blank line entries are allowed.
- Only one entry per line is allowed.

/etc/services Permissions

The */etc/services* file should be owned by user *root*, group *other*, and have 0444 (-r—r—r—) access permission.

Refer to the */etc/services* file for examples of actual format and contents. For more information on */etc/services*, refer to the *services(4)* entry in the *Network Services Reference Pages*.

/etc/services Example

The */etc/services* entry for a service with:

- The service name *shell*.
- The port number 514.
- The protocol name *tcp*.
- The alias name *cmd*.

looks like:

```
shell 514/tcp cmd
```

Modifying the /etc/protocols File

The */etc/protocols* file associates port numbers with mnemonic names and alias names. The */etc/protocols* file contains the names and protocol numbers of all protocols known to your local host. The *netstat* diagnostic uses the */etc/protocols* file. If you install ARPA Services/9000 or NFS/9000, those products also will use the */etc/protocols* file.

Note You can modify this file if you have special requirements, but it is properly configured when you receive the LAN/9000.

/etc/protocols

Each protocol has a one line entry in the */etc/protocols* file. Each entry in */etc/protocols* file takes the following form:

Syntax

```
protocol_name  protocol_num [alias(es)]
```

Parameters

protocol_name	Name of the protocol. Protocol names can contain any printable character except spaces, newline, or the comment character (#).
protocol_num	Protocol number that identifies this protocol.
alias(es)	Common name or names for the protocol. An *alias* is a substitute for *protocol_name*. *Alias* names are optional.

/etc/protocols Format

- Lines cannot start with a blank or tab character.
- Fields can have any number of blanks or tab characters separating them.
- Comments are allowed; they are designated by a pound sign (#) character preceding the comment text.
- Trailing blank and tab characters are allowed.
- Blank line entries are allowed.
- Only one entry per line is allowed.

/etc/protocols Permissions

The */etc/protocols* file should be owned by user *root*, group *other*, and have 0444 (-r—r—r—) access permission.

Refer to the */etc/protocols* file for examples of actual format and contents. For more information on */etc/protocols*, refer to the *protocols(4)* entry in the *LAN/X.25 Reference Pages*.

/etc/protocols Example

The */etc/protocols* entry for a protocol with:

- The protocol name *tcp*.
- The protocol number *6*.
- The alias name *TCP*.

looks like:

```
tcp 6 TCP
```

Rebooting the System

If you have added new LAN cards, you need to reboot. Once rebooted, the LAN product will be running with full network capability.

To reboot your system, enter:

```
/etc/shutdown -r
```

Verifying the Installation

Once your LAN software is installed, fully configured and running, do the following to check the installation.

1. To view your system's name and LAN card information, enter the following commands at the HP-UX prompt and view their output:

   ```
   lanscan
   nodename
   hostname
   more /etc/hosts
   ```

2. To check the status of your system's LAN cards, enter the following commands at the HP-UX prompt and view their output:

   ```
   ifconfig lan0
   ifconfig lan1
   ```

   and so on for each LAN card. Also enter:

   ```
   netstat -i
   ```

Running the LAN Verification Script

LAN/9000 Series 300/400, Series 700, and 600/800 software provides a script for verification of LAN hardware and software. You should run this script after installation/configuration of LAN/9000 and after installation of additional LAN cards. It may also be helpful to run this verification script when you encounter problems with the LAN.

This script will perform the following verification tests:

- Check that the backplane contains the supported number of LAN cards.
- Check the state of the LAN card hardware.
- Check the state of the LAN interface.
- Verify the link encapsulation.
- Check for the existence of device files.
- Test for loopback link level connectivity.
- Test for network level connectivity to remote node (-h option).
- Test for transport level connectivity to remote node (-r option).
- Check for nodename configuration.

/usr/nettest/ver_link

Syntax is as follows:

Syntax

```
/usr/nettest/ver_link [-h hostname][-r nodename] [ -k kernel]
```

-h *hostname* is an optional parameter used to test network level connectivity to other UNIX Internet machines. hostname specifies the name of the local (for loopback testing) or remote host to which connectivity testing is desired.

-r *nodename* is an optional parameter used to test transport level connectivity to other Series 300/400, Series 700, and 600/800 computers that have installed

NetIPC and the rlbdaemon. nodename specifies the name of the local (for loopback testing) or remote node to which connectivity testing is desired.

-k *kernel* is an optional parameter used to specify the name of the HP-UX kernel if the name is other than the default, `/hp-ux`.

If the LAN verification script encounters a problem, then either a warning or error message will appear on your terminal screen. Take note of the message and follow the recommended corrective action.

Manually Testing the Installation

If you cannot find the LAN verification script, you can do the following verification tests manually. If you are adding your system to an existing network, you can exercise remote loopback tests with the remote systems.

- To test the NetIPC-level connection to an HP 9000 Series 600/800, Series 700 or Series 300/400, use the *rlb(1M)* remote loopback test. *rlb(1M)* exchanges a message with a remote node from the NetIPC layer to the Physical Layer (OSI Layer 1). For a detailed description of *rlb(1M)*, refer to Chapter 6.
- To test connectivity to an HP 9000 Series 600/800, Series 700, or 300/400, or an HP 1000 A-Series computer, use the *ping(1M)* command. Refer to Chapter 6 for a detailed description of *ping(1M)*.
- First check that the NetIPC fileset has been installed by checking for it in */etc/filesets*. If the diagnostic *rlb(1M)* fails, test that the network daemons are running. Issue the command */bin/ps -ef | grep d*. You should see one entry per network daemon in the table of statistics returned to standard output. If you don't see an entry for a daemon, start it by typing the daemon name (as an absolute pathname) on the command line. You must be a super-user to start a network daemon. Required LAN Daemon: rlbdaemon. Then try *rlb(1M)* again.
- **Series 800 only:** If the diagnostic *rlb(1M)* fails and its daemon is running, you can test the Link Layer (OSI Layer 2) and Physical Layer (OSI Layer 1) with the *LANDAD* section of *sysdiag*. *sysdiag* is the system hardware diagnostic tool for the Series 600/800. *LANDAD* tests the LAN link on a Series 600/800. The *LANDAD* section executes a Link Layer (OSI Layer 2) loopback test with a remote LAN/9000 Series 300/400, 500, 600/800 or HP 1000 A-Series node. Refer to the appropriate link installation manual and the *On-Line Diagnostic Subsystem Manual* for a detailed description of the *LANDAD* diagnostic.

- *linkloop(1M)* can also execute a Link Layer (OSI Layer 2) loopback test with a remote LAN/9000 Series 300/400, Series 700, or Series 600/800 node. The link loopback test exercises all the hardware components including the LAN cable, the backplane, and the IEEE 802.3 driver.
- If the *rlb(1M)* remote communications diagnostic fails and the *rlb* daemon is running, use *rlb(1M)* to test the LAN interface. The LAN interface test executes a link level (OSI Layer 2) loopback with a remote LAN/9000 Series 300/400, 500, 700, 600/800 or HP 1000 A-Series node. linkloop(1M) can also be used to execute the Link Layer (OSI Layer 2) loopback test with a remote LAN/9000 Series 300/400, Series 700, or 600/800 node. The link loopback test exercises all the hardware components including the LAN cable, the backplane, and the IEEE 802.3 driver.

If the Series 600/800 or Series 700 is the first node configured onto a network, you can exercise the *rlb(1M)* test locally. This tests LAN/9000 from NetIPC level down to the IP Network Layer (OSI Layer 3). Then you can use the *LANDAD* section of *sysdiag* or *linkloop* to execute a local Link Layer (OSI Layer 2) loopback test, which tests all of the hardware pieces down to the LAN cable on your local node.

The *rlb(1M)* diagnostic is described in detail in Chapter 6. *sysdiag* is explained in the *On-Line Diagnostic Subsystem Manual.*

Note HP recommends that the On-Line Diagnostic Subsystem be used by HP Customer Engineers or by trained customers only.

4

Maintaining LAN

This chapter provides information on maintaining LAN/9000. It contains the following sections:

- Modifying LAN Hardware Configuration.
- Modifying LAN Software Configuration.
- Modifying the Routing Table.
- Subnetting Example.
- Overview of Network Daemons and Library Routines.

Modifying LAN Hardware Configuration

Follow the procedure below to add or replace LAN cards.

- Shut down the system.
- Add the new LAN cards or replace existing LAN cards which are not in order.
- Reboot the system.
- Configure the new LAN cards manually or by running SAM to assign an IP address and host name for each LAN card, and then add network connectivity by configuring */etc/netlinkrc*.

  Series 300/400 Only: If you are adding a second LAN card, you do not need to rebuild the kernel. You must run SAM, however, to create device files for the new LAN card or create them manually. If you are adding a third, fourth, or fifth LAN card, you may need to rebuild the kernel depending on whether the *dfile* contains the line `num_lan_cards n` where `n = 3, 4,` or `5`. If you configure the new card with SAM, SAM will add this line into the *dfile*, rebuild the kernel, and reboot the system. Run SAM again to create LAN device files (SAM does this silently) for the new LAN cards or create the device files manually. If you configure the new card manually, you must edit the *dfile* to add the line, rebuild the kernel, and reboot the system.

Modifying LAN Software Configuration

You may modify software configuration of the network interface anytime using *ifconfig(1M)* and *lanconfig(1M)* commands. *ifconfig* can be used to assign a new IP address to the interface and to change operating parameters. *lanconfig* is used to specify what protocol runs on the interface.

Note Each *ifconfig* entry should be followed by an entry for *lanconfig.*

ifconfig(1M)

The *ifconfig* command takes the following form:

Syntax

```
ifconfig interface address_family [address[dest_address]][parameters]
```

Parameters

interface A string of at most four alphabetic characters followed by an integer. The alphabetic characters denote the network interface. The integer denotes the network interface unit for the device which connects to the network. To use LAN device as the interface, the interface string is *lan*, and the interface unit number is determined as follows:

The LAN card in the lowest hardware module in the backplane is interface unit number 0; the LAN card in the next higher hardware module is interface unit number 1; and so on. If there is more than one LAN card in a module (e.g. CIO), the interface unit numbers will be assigned to the LAN cards in that module before numbers are assigned to those in the

next higher module. For example, if a system has two LAN cards in CIO module 4 (slot 3 and slot 7) and one LAN card in CIO module 8 (slot 5), the LAN cards will be assigned the interface unit numbers 0, 1, and 2 respectively.

You can use the *lanscan(1M)* command to display the string and unit of each interface that is associated with a LAN card.

address

The address is either a host name present in the host name database, hosts(4), or a DARPA Internet address expressed in dot notation. The host number may be omitted on 10 Mb/s Ethernet interfaces, which uses the hardware physical address, and on interfaces other than the first.

address_family

The address format used to interpret addresses specified in socket operations. The internet address family (AF_INET) is supported.

up

Mark an interface "up." This may be used to enable an interface after an "ifconfig down." It happens automatically when setting the first address on an interface. If the interface was reset when previously marked down, the hardware will be re-initialized.

down

Mark an interface "down." When an interface is marked "down," the system will not attempt to transmit messages through that interface. If possible, the interface will be reset to disable reception as well. This action does not automatically disable routes using the interface.

trailers

Request the use of a "trailer" link level encapsulation when sending. If a network interface supports trailers, the system will, when possible, encapsulate outgoing messages in a manner which minimizes the number of memory-to-memory copy operations performed by the receiver. On networks

that support the Address Resolution Protocol this flag indicates that the system should request that other systems use trailers when sending to this host. Similarly, trailer encapsulations will be sent to other hosts that have made such requests. Currently used by Internet protocols only.

`-trailers` — Disable the use of a "trailer" link level encapsulation (default).

`arp` — Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between DARPA Internet addresses and 10Mb/s Ethernet addresses.

`-arp` — Disable the use of the Address Resolution Protocol.

`metric` *n* — Set the routing metric of the interface to n, default 0. The routing metric is used by the routing protocol (see *gated(1M)*). Higher metrics have the effect of making a route less favorable; metrics are counted as addition hops to the destination network or host.

`debug` — Enable driver dependent debugging code; usually, this turns on extra console error logging.

`-debug` — Disable driver dependent debugging code.

`netmask` *mask* — (Inet only) Specify how much of the address to reserve for subdividing networks into sub-networks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or with a pseudo-network name listed in the network table *networks(4)*. The mask contains 1's for the bit positions in the 32-bit address which are to be used

for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion.

dest_address — Specify the address of the correspondent on the other end of a point-to-point link.

broadcast — (Inet only) Specify the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's.

ipdst — (NS only) This is used to specify an Internet host who is willing to receive ip packets encapsulating NS packets bound for a remote network. In this case, an apparent point-to-point link is constructed, and the address specified will be taken as the NS address and network of the destinee.

Description

ifconfig is used to assign an address to a network interface and configure network interface parameters. *ifconfig* must be used at boot time to redefine an interface's address or other operating parameters.

ifconfig displays the current configuration for a network interface when no optional parameters are supplied. If a protocol family is specified, *ifconfig* will report only the details specific to that protocol family.

Only the super-user may modify the configuration of a network interface.

Following is a typical example of the use of *ifconfig*.

To assign the Class C IP address *192.6.1.17* and the subnet mask *255.255.255.240* to the network interface *lan0*, issue the following command:

```
ifconfig lan0 192.6.1.17 netmask 255.255.255.240 up
```

lanconfig (1M)

The *lanconfig* command takes the following form:

Syntax

```
lanconfig interface [ether][ieee]
                    [-ether][-ieee]
```

Parameters

interface	A string of the form "*name unit*," e.g. "*lan0*."
ieee	Enables IEEE 802.3 protocol over the network interface.
-ieee	Disables IEEE 802.3 protocol over the network interface.
ether	Enables Ethernet protocol over the network interface.
-ether	Disables Ethernet protocol over the network interface.

lanconfig displays the current configuration for a network interface when no optional parameters are supplied.

Description

lanconfig is used to define the packet encapsulation method for a network interface. The default encapsulation is Ethernet only. 802.3 packet encapsulation is needed only when an HP 9000 interacts using HP proprietary NFT(DSCOPY) with an HP 3000, HP 1000, or Vectra PC that does not support Ethernet. In these situations you should modify the *lanconfig* command line in */etc/netlinkrc* to include 802.3 encapsulation.

lanconfig must be used at boot time to configure each interface present on a machine. It may also be used at a later time to redefine an interface's configuration.

Following is a typical example of the use of *lanconfig*.

```
lanconfig lan0 ieee
```

Modifying the Routing Table

The routing table allows your system to communicate through a gateway. You create routing table entries with the *route(1M)* command.

Before you bring up the network, the only entry in the routing table is *lo0*, the loopback interface This corresponds to the *loop* entry in the */etc/networks* file. If your system is a gateway, or if it uses gateways, you make additional entries at installation time using *route(1M)* in the */etc/netlinkrc* file (refer to "Editing and Installing /etc/netlinkrc" in Chapter 3).

After system boot up, you may add or delete a route anytime using *route(1M)* at the command line.

route(1M)

The *route* command takes the following form:

Syntax

```
/etc/route [-f] command [net] destination gateway [count]
                        [host]
```

Parameters

`-f` Specifies that *route* will "flush" the routing table of all gateway entries. If this is used with one of the commands described below, the tables are flushed before the command's application.

command Specifies which *route* command to use: *add* or *delete*. *add* adds the specified host or network to the network routing table. *delete* deletes the specified host or network entry from the network routing table.

`net`	Specifies that *destination* is a network. Use *net* when you are using subnetting and *destination* is ambiguous, that is, *destination* could be interpreted as either a network or a host if you didn't specify it as a *network*.
`host`	Specifies that *destination* is a host. Use *host* when you are using subnetting and *destination* is ambiguous, that is, *destination* could be interpreted as either a network or a host if you didn't specify it as a *host*.
`destination`	Specifies the host or network where packets will be routed. *destination* may be either a host name (or alias as listed in */etc/hosts*), a network name (or alias as listed in */etc/networks*), an internet address in "dot" notation (see *inet(3N)* in the *LAN/X.25 Reference Pages*) or the keyword *default*. If the keyword *default* is specified for *destination*, the default gateway entry is changed to *gateway*. The default "wild-card" gateway is where packets are routed if they match no other destination in the route table.
`gateway`	Specifies the gateway node through which *destination* is reached. A gateway node must be specified in the */etc/hosts* file or as an internet address in "dot" notation. See the *inet(3N)* entry in the *LAN/X.25 Reference Pages* for details on internet "dot" notation.
`count`	Integer indicating whether the gateway is a local or remote host. If *count* is greater than 0, the gateway is a remote host. If *count* equals 0, the gateway is the local host. Default: 0.

The routing table can be displayed with the *netstat -r* command.

For more information on *route*, refer to the *route(1M)* and *routing(7)* entries in the *LAN/X.25 Reference Pages*.

Subnetting Example

The following example shows how to use *ifconfig(1M)* and *route(1M)* to subnet on a Class C address. The network map is shown in Figure 4-1.

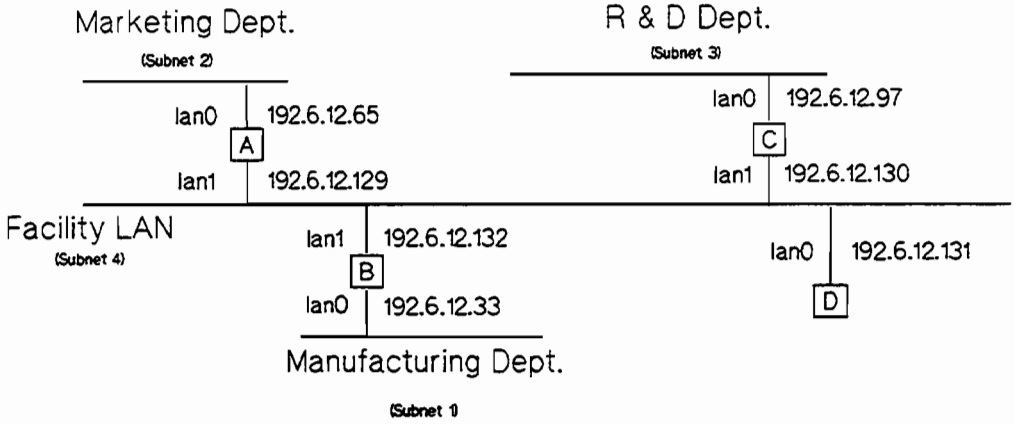

Figure 4-1. Network Map for Subnetting

Company network address = 192.6.12
  Netmask: 255.255.255.224

Manufacturing Department subnet number = 1
  Host address range: 33 to 62
  Host B internet address: 192.6.12.33 for lan0

Marketing Department subnet number = 2
  Host address range: 65 to 94
  Host A internet address: 192.6.12.65 for lan0

R & D Department subnet number = 3
  Host address range: 97 to 126
  Host C internet address: 192.6.12.97 for lan0

Facility LAN subnet number = 4
  Host address range: 129 to 158
  Host A internet address: 192.6.12.129 for lan1
  Host B internet address: 192.6.12.132 for lan1

Host C internet address: 192.6.12.130 for lan1
Host D internet address: 192.6.12.131 for lan0

To set the subnet masks, you include them in the *ifconfig* command that starts up the LAN interface card for each host. The hosts in the example above would require the following *ifconfig* commands:

Host A:

```
/etc/ifconfig lan0 192.6.12.65 netmask 255.255.255.244
/etc/ifconfig lan1 192.6.12.129 netmask 255.255.255.244
```

Host B:

```
/etc/ifconfig lan0 192.6.12.33 netmask 255.255.255.244
/etc/ifconfig lan1 192.6.12.132 netmask 255.255.255.244
```

Host C:

```
/etc/ifconfig lan0 192.6.12.97 netmask 255.255.255.244
/etc/ifconfig lan1 192.6.12.130 netmask 255.255.255.244
```

Host D:

```
/etc/ifconfig lan0 192.6.12.131 netmask 255.255.255.244
```

In addition, every other host on each subnetwork would require the subnet mask 255.255.255.224 in their *ifconfig* command.

Besides using the appropriate subnet masks, each gateway to a subnet needs to identify other gateways to subnets in its routing table. To initiate proper routing between the subnets, you would add the following entries to each host's routing tables:

Host A:

```
/etc/route add net 192.6.12.33 192.6.12.132 1
/etc/route add net 192.6.12.97 192.6.12.130 1
```

Host B:

```
/etc/route add net 192.6.12.65 192.6.12.129 1
/etc/route add net 192.6.12.97 192.6.12.130 1
```

Host C:

```
/etc/route add net 192.6.12.65 192.6.12.129 1
/etc/route add net 192.6.12.33 192.6.12.132 1
```

Host D:

```
/etc/route add net 192.6.12.65 192.6.12.129 1
/etc/route add net 192.6.12.33 192.6.12.132 1
/etc/route add net 192.6.12.97 192.6.12.130 1
```

subnetconfig(1M)

The *subnetconfig* command takes the following form:

Syntax

```
subnetconfig [local|remote]
```

Parameters

`local`	Instructs the system to treat all subnets belonging to the same network as being "local."
`remote`	Instructs the system to treat only the directly attached subnet as being "local."

When no parameters are specified, *subnetconfig* displays the current status of an internal flag which controls subnet behavior.

Description

You use the *subnetconfig* command to set/reset the value of the internal flag which controls the subnet behavior of a host. The default setting of the flag considers all subnets on the same network to be local. The command, *subnetconfig remote* specifies that only the directly attached subnet should be considered as local. The setting of the internal flag affects the maximum size of TCP packets sent out on the network.

TCP chooses the maximum segment size on a per connection basis at connection setup time. When connecting between hosts on local subnets, TCP's choice of maximum segment size is limited only by the size of the MTU of the interface being used to send packets. When connecting between hosts on remote subnets, TCP always chooses a maximum segment size of 512 bytes. You may change the definition of local and remote subnets within the same network with the *subnetconfig* command. For connections between hosts which do not belong to the same network, the size of the maximum segment size is always 512 bytes.

In the example shown in Figure 4-1, all the subnets on the 192.6.12 network are considered local subnets by default. If on Host A, the system administrator configures remote subnets using the *subnetconfig* command, then only the hosts belonging to subnets 192.6.12.65 and 192.6.12.129 will be considered local subnets.

The effect of choosing smaller packet sizes between hosts on the same networks, but different remote subnetworks, may result in a noticeable performance degradation of TCP. On the other hand, if the connection between two hosts involves significant fragmentation at gateways, the use of 512 bytes may actually improve performance because of lesser overhead at the gateways.

Overview of Network Daemons and Library Routines

This section provides a quick reference of the daemons and library routines that are provided and used by the LAN/9000 product.

Daemons

When the system is brought up, the */etc/netlinkrc* initialization script starts the *netisr, rlbdaemon,* and *inetd* LAN/9000 daemons (if they are executable).

netisr	The network interface daemon. It allows for system wide performance improvements, particularly real-time responses.
rlbdaemon	The daemon used by the *rlb* diagnostic.
inetd	The daemon that supports the ARPA Services/9000. Each time a service command is invoked, *inetd* initiates the server for the specific service.

Library Routines

The following library routines are provided by the LAN/9000 product.

byteorder(3N)	Converts values between host and network byte order.
gethostent(3N)	Gets network host entries.
getnetent(3N)	Gets network entries.
getprotoent(3N)	Gets protocol entries.
getservent(3N)	Gets service entries.
inet(3N)	Provides internet address manipulation routines. Used by ARPA Services/9000 and NFS Services only.
rcmd(3N)	Provides routines for returning a stream from a remote command. *rcmd* is reserved for

	super-user use. Used by ARPA Services/9000 only.
rexec(3N)	Returns a stream to a remote command. Used by ARPA Services/9000 only.

5

Troubleshooting LAN

This chapter provides guidelines for troubleshooting LAN/9000. It contains the following sections:

- Troubleshooting Overview.
- Identifying the Problem.
- Using Diagnostic Flowcharts.
- Contacting your HP Representative.

Troubleshooting Overview

Troubleshooting LAN problems can be difficult. A variety of hardware and software components may be involved. If there is a gateway on the LAN, the problem may originate in another network.

As with any troubleshooting, a systematic approach is helpful. Following is the recommended sequence of steps for troubleshooting LAN:

1. Identify the problem as specifically as you can.
2. Using the diagnostic flowcharts provided in this chapter, verify your assumptions and correct the problem.
3. If you can not solve the problem on your own, contact your HP representative. Use the guidelines at the end of this chapter to help you effectively communicate what is wrong.

Note This chapter contains references to diagnostic utilities and messages described elsewhere in this manual:

Chapter 6 – Using Network Diagnostics
Appendix A – Installation Error Messages
Appendix B – Diagnostics Error Messages
Appendix C – Network Event Logging Messages
Appendix D – LAN Interface Card Statistics
Appendix E – LAN Interface Card Self-test Codes (Series 300/400 only)

Identifying the Problem

Start by identifying a problem as specifically as you can. That will help you confirm that the problem is related to the network. It also helps you decide which diagnostics are appropriate to verify and/or correct the problem.

To help you identify a problem, ask the following questions:

1. What is the scope of the problem? Is it limited to one user or one host? Is it network-wide?

   If the problem is limited to one user, it is likely caused by the user's serial device, the device cabling or the terminal I/O port serving that device. If so, this is not considered to be a network problem. For the appropriate diagnostics, refer to documentation provided with the serial device or terminal I/O port involved.

   If the problem is limited to one host, verify that it is a network problem. Check that a network application was being attempted, either interactively or programmatically, at the time of difficulty. A network application is considered to be any of the services provided by NS/9000, NFS/9000 or ARPA/9000, or a custom application based on NetIPC or BSD IPC programmatic interfaces.

   If it is a network problem, and is limited to one host, the problem is likely caused by improper LAN/9000 software configuration within the host, a faulty LAN card or improper or faulty LAN connections. Flowcharts within this chapter help to isolate each of these conditions.

   If the problem is not limited to one user or host, it may be isolated to a segment of a LAN. In this case, suspect improper software configuration or faulty hardware within hosts in that segment.

   Finally, if the problem is network-wide, suspect a faulty LAN cable, gateway or repeater.

2. Has there been a configuration change recently?

   Many times problems result from a change in hardware or software configuration of a host on the LAN. The network interface may be configured improperly or a cable may be installed wrong. Such a problem can effect the entire network if the host is used as a gateway.

3. Is the problem hardware- or software-related?

   Symptoms that indicate hardware problems are intermittent errors and link level errors logged from the LAN driver. A hardware fault is also indicated by a network-wide problem experienced after no changes in software. Finally, a hardware problem may be indicated by a link level trace that shows data is sent without error but is corrupt or lost at the receiver.

   Symptoms of software problems include error and logging messages from software modules. Software faults are also indicated by link level traces that show data is corrupt at the link level of both sending and receiving hosts.

4. If a software problem is indicated, what module is involved?

   Most error and logging messages identify which module is sending the message. For instance, each LAN/9000 logging message has a header such as this:

   ```
   Oct 19 15:18:53:96328: Network Probe Error 2012, pid 86
   ```

   Here, "Network Probe Error 2012" indicates this message comes from the Probe module. The message is Probe log error number 2012. Other LAN/9000 modules are: Buffer Manager, IPC, IP, LAN, Logging, NetISR, NFS, NFT, and NS. For more information on reading log messages, refer to Chapter 7. For a complete listing of LAN/9000 logging messages, refer to Appendix C.

Error messages usually appear during interactive use, and it is easy to identify their source. For instance, if you encounter a problem using the telnet service of ARPA/9000, the following may appear:

```
telnet/tcp: Unknown service
```

Note This manual contains diagnostics for LAN/9000 software modules only. For details on troubleshooting network services such as those provided by NFS/9000, NS/9000 or ARPA/9000, refer to the following documentation:

Using ARPA Services
Using Network Services
Using NFS Services

Be aware that, although an error message is sent from a network service, the problem may be in LAN/9000 or network components underlying that service. Flowcharts within the ARPA/9000, NS/9000 or NFS/9000 manuals cover this possibility.

Other error messages may not identify their source, but it is explicit given what you were doing when the error occurred. For instance, suppose you are using *ping(1M)* for a diagnostic loopback to a remote host whose alias is joey. Further suppose that the alias joey is not recorded in your system's */etc/host* file. When you attempt to ping to joey, the following message appears:

```
unknown host joey
```

Using Diagnostic Flowcharts

After you have identified the problem, use the diagnostic flowcharts in this section to verify and correct it. The flowcharts are listed in Table 5-1.

Table 5-1. Diagnostic Flowcharts

Flowchart(s)	Description
1, 2 & 3	Configuration Test
4 & 5	Network Level Loopback Test
6	Transport Level Loopback Test (using rlb)
7	Transport Level Loopback Test (using ARPA)
8	Link Level Loopback Test
9 & 10	LAN Card Test (Series 300/400 only)
11	LAN Card Test (Series 600/800 and Series 700)
12	LAN Connections Test
13	Gateway Configuration Test
14	Gateway Loopback Test
15	Probe Proxy Server Test
16	Subnet Test

The following paragraphs describe each flowchart and its recommended use. This is followed by an explanation of flowchart conventions and the fourteen flowcharts themselves.

Flowchart Descriptions

There are many ways to use the flowcharts. As you gain experience, you will find which flowcharts suit the majority of problems at your installation. The following paragraphs are general recommendations only.

Configuration Test

This verifies configuration of the network interface on a host. The network interface consists of the LAN card and supporting LAN/9000 software.

The Configuration Test is implemented with the *ifconfig(1M)* command. It requires super-user privileges. Use this test if you experience a problem following a configuration change on the host.

Network, Transport, and Link Level Loopback Tests

These are three different loopback tests to help isolate a network communication problem to a specific OSI layer. Each checks roundtrip communication between peer layers on two different hosts. The tests differ in how they are implemented and in what layers they check. Figure 5-1 shows how the tests relate to OSI layers.

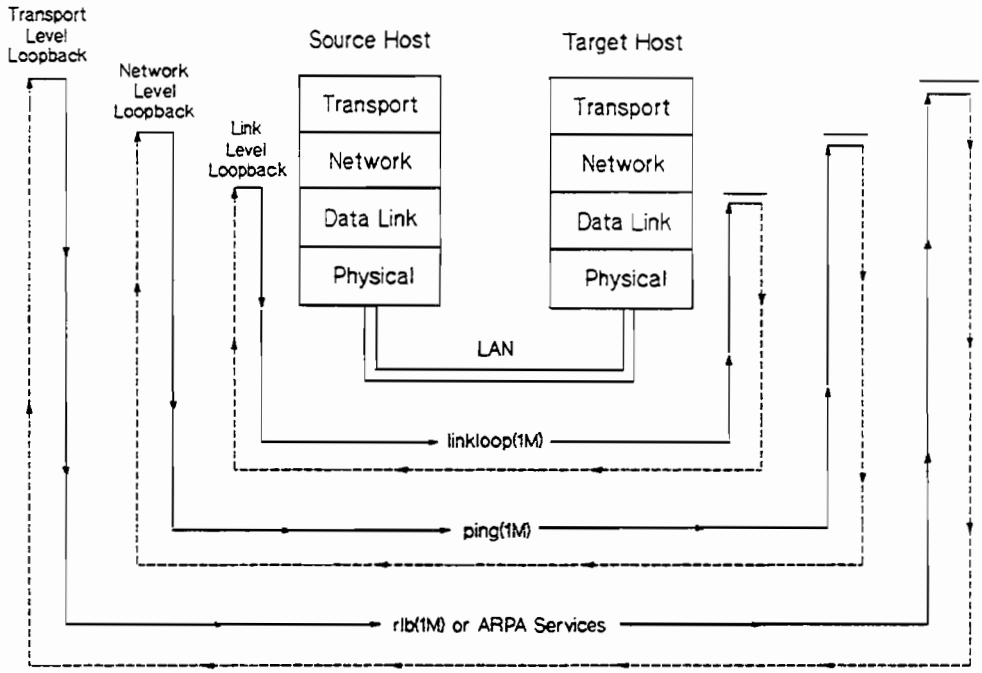

Figure 5-1. Loopback Tests

As shown, the Network Level Loopback Test operates between Network Layers on the source and target hosts. It is implemented with *ping(1M)*. A test packet is sent by the source host, through the network, up to the corresponding layer on the target host. After the message is received, a return packet is sent back to the source.

The Transport Level Loopback Test is similar, but packets are exchanged between Transport Layers. Again, the source host transmits a test packet, then waits for a response by the target. This is implemented either with *rlb(1M)* or by using ARPA/9000 services.

The Link Level Loopback Test is similar to the other two, but it operates through the Link Layer. It is implemented with *linkloop(1M)*.

Typically, the three loopback tests are used to isolate a network communication problem that may be software- or hardware-related. In any case, you should first have checked that the problem is not due to a recent configuration change.

The Network Level Loopback Test commonly is tried first. It is fast, efficient, and it does not require super-user privileges. If the connection passes this test, you know the problem is at OSI Layer 4 or above. Go on to the Transport Level Loopback Test.

As mentioned, the Transport Level Loopback Test can be implemented two ways. The first utilizes *rlb(1M)* as the loopback command. Note that, to use *rlb*, NetIPC must be installed on your host. In addition, the *rlbdaemon* must be executing.

If you have not installed NetIPC, you may do a Transport Level Loopback Test using ARPA/9000 services. In this case, you use telnet and ftp to systematically focus on a problem.

If the Network Level Loopback Test failed, the problem is in OSI Layer 3 or below. In this case, continue with the Link Level Loopback Test to isolate a problem to OSI Layer 2 or below.

LAN Card Tests

If the Link Level Loopback Test fails, or if a LAN card problem is indicated by the Configuration Test, test the LAN card hardware. Two flowcharts are provided for the Series 300/400 LAN card and one for the Series 600/800 and Series 700 LAN card.

The first Series 300/400 flowchart is a hardware check using *landiag(1M)*. Some commands of the *landiag(1M)* interface require super-user privileges. The second Series 300/400 flowchart allows you to check that the correct select code switches are set on the card.

LAN Connections Test

If there is a network communication problem, but each host appears to be properly configured and operational, suspect LAN connections. In this case, LAN connections may refer to the LAN media itself, and any component between the LAN media and individual LAN cards. This includes the coaxial cable, MAUs and any associated connections and taps.

Gateway and Repeater Tests

You may have a problem that effects a LAN segment, or perhaps the entire LAN. Such a problem may be caused by improper use of repeaters, gateways or subnets. Four flowcharts are provided that relate to use of gateways and subnets. The following recommendation is provided for repeaters.

If you have a problem with a LAN segment, check if a repeater is involved. A typical installation is shown in Figure 5-2.

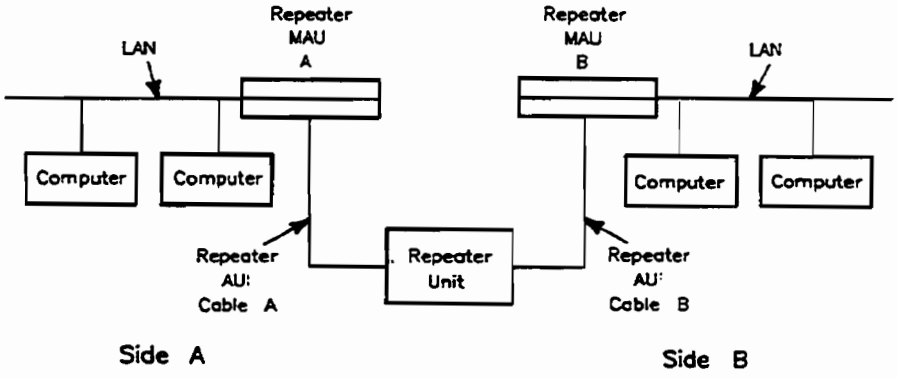

Figure 5-2. LAN with Repeater

As shown, the repeater divides the LAN into two segments, labeled A and B. Suppose hosts on the A side can communicate with each other, but not with hosts on the B side. Further, suppose B side hosts can communicate with each other but not with A side hosts. In this case, the repeater is likely at fault. Replace the unit or perform further diagnostics as recommended by the manufacturer.

If a problem is network-wide, improper use of gateways and subnets may be involved. In this case, it is assumed LAN hardware and connections are operational, and that there is no problem with local LAN traffic. The problem is that either hosts on your LAN can not communicate with a host on a remote network, or a remote host is having trouble connecting with your LAN.

The gateway and subnet flowcharts cover a number of different circumstances. The Gateway Configuration Test, Flowchart 13, is used to check configuration

of multiple network interfaces on a host. This flowchart is a supplement to the general configuration test, Flowchart 1.

The Gateway Loopback Test, Flowchart 14, provides a general check of network connections through a gateway. It is similar to the Network Layer Loopback Test, only this time you are going through a gateway.

Flowchart 15 is useful if you are having problems operating NS/9000 between hosts on separate networks. In this case, you may have a problem with the Probe Proxy Server. The server provides information on local station addresses to a remote host trying to establish a connection.

Finally, Flowchart 16 is used to verify correct use of subnets. If you are using subnetting and have ruled out other causes of LAN difficulty, use Flowchart 16.

Flowchart Conventions

The flowcharts use a series of symbols to indicate which steps to perform next. The symbols are explained in Figure 5-3.

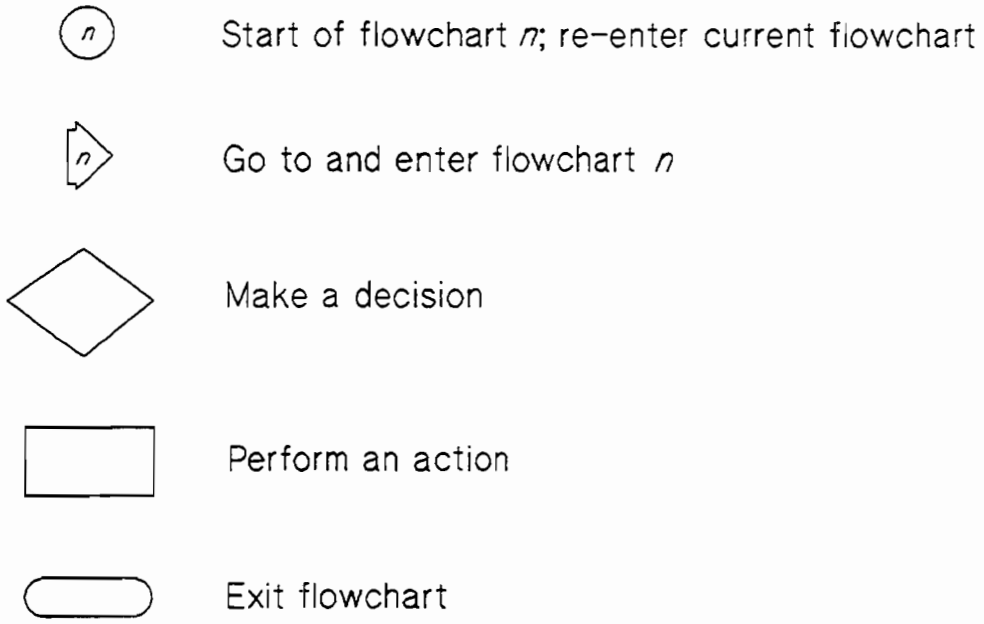

Figure 5-3. Flowchart Conventions

Flowchart 1: Configuration Test

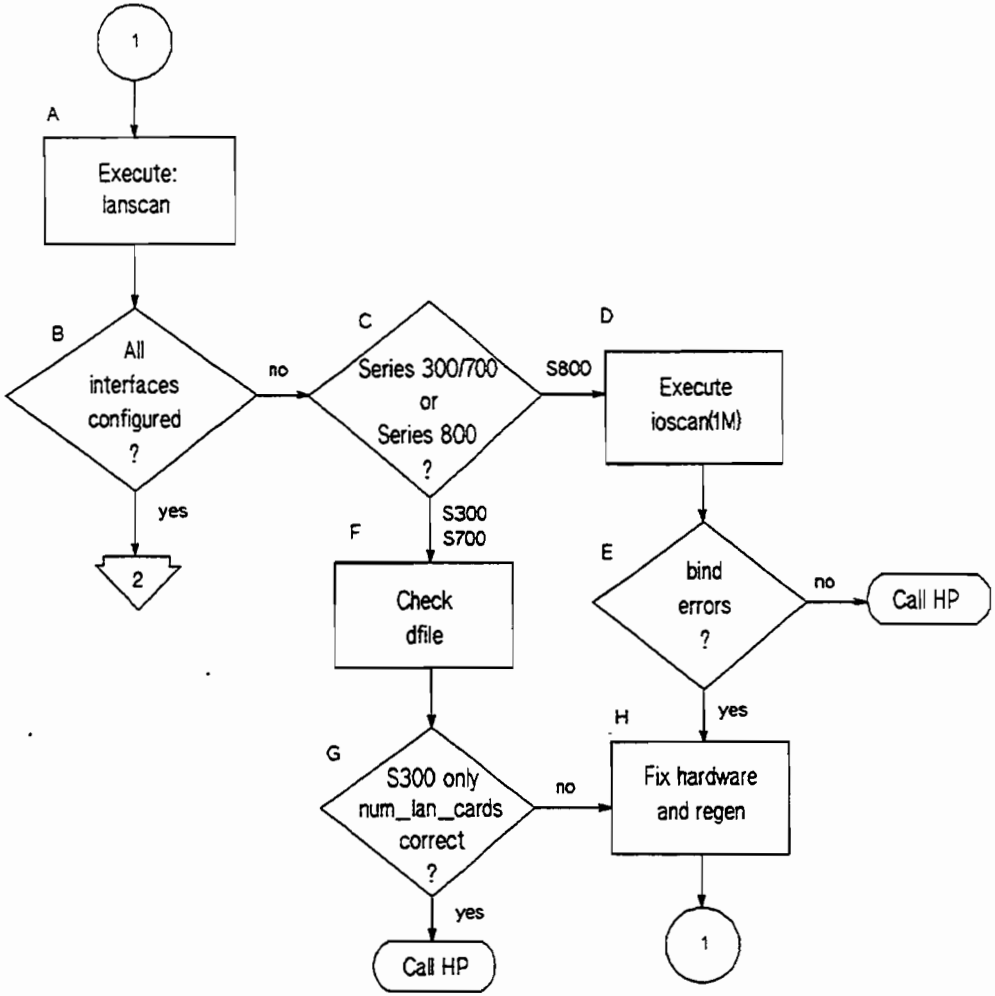

Figure 5-4. Flowchart 1

Flowchart 1 Procedures

A. **Execute: lanscan.** Execute *lanscan* to display information about LAN cards that are successfully bound to the system. For example, to check the cards on /hp-ux, enter:

```
/etc/lanscan /hp-ux
```

B. **All interfaces configured?** *lanscan* is successful if the output shows information about every card in the hardware backplane.

C. **Series 300/400, Series 600/800 or Series 700?** Determine the type of system you are working on and proceed.

D. **Execute ioscan.** Execute the *ioscan(1M)* command to check for bind errors.

E. **bind errors?** If a bind error exists, check that the hardware has been properly installed.

F. **Check dfile.** If there are three or more LAN cards, check that the value of *num_lan_cards* in the *dfile* is correct.

G. **num_lan_cards correct?** If not, correct the value.

H. **Fix hardware and regen.** Make sure hardware is properly installed and regen the kernel if necessary.

Flowchart 2: Configuration Test — cont.

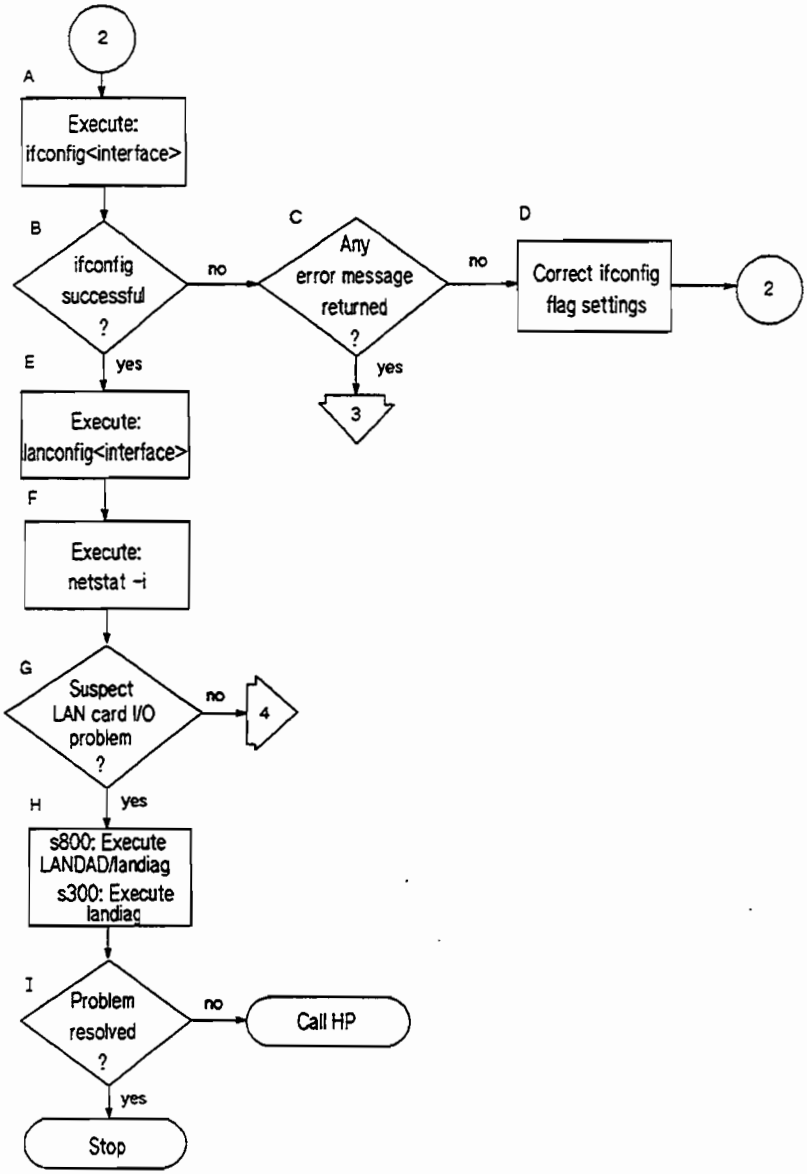

Figure 5-5. Flowchart 2

Flowchart 2 Procedures

A. **Execute: ifconfig <interface>.** Execute *ifconfig* on the interface you want to test. For example, to check LAN interface lan0, enter:

```
/etc/ifconfig lan0
```

B. **ifconfig successful?** *ifconfig* is successful if the output shows the correct Internet address and the flags: *<UP, BROADCAST,ROUTE,NOTRAILERS, RUNNING>*.

C. **Any error message returned?** If *ifconfig* is not successful, and an error message appears, go to Flowchart 3. Flowchart 3 shows common error messages and what to do for each.

D. **Correct ifconfig flag settings.** If *ifconfig* returns an incorrect flag setting, re-execute the command with the proper setting. For more information, refer to the *ifconfig(1M)* manual page. Start again with Flowchart 1, as necessary.

E. **Execute: lanconfig <interface>.** Execute *lanconfig* on the interface without any optional parameters to display the current configuration. For example, to check LAN interface lan0, enter:

```
lanconfig lan0
```

F. **Execute: netstat -i.** If *ifconfig* is successful, you know the network interface has been configured correctly. Using *netstat*, you can return statistics which show the interface is operational.

Attempt a file transfer to a remote note, then enter the following:

```
/usr/bin/netstat -i
```

netstat statistics give a quick check of key operating parameters. For instance, if the *opkts* value does not change after attempting the file transfer, packets are not being transmitted. Similarly, if the *ipkts* value does not change, packets are either not being received by the local node or

are not being sent by the remote node, which may not be receiving your transmissions. If the values of the *ierrs* and *oerrs* fields increase substantially during a file transfer attempt, this can indicate transmission or reception problems.

Finally, *netstat -i* can indicate, by not printing any information for a particular interface, that an I/O card is missing.

G. **Suspect LAN card I/O problems?** If the statistics indicate possible LAN card problems, go to G, otherwise go to Flowchart 4.

H. **S600/800 and S700: Execute LANDAD/landiag; S300/400:** Execute landiag. Use *landiag (Series 300/400, Series 600/800* or *Series 700)* or *LANDAD (Series 600/800,* or *Series 700)* to ensure the LAN card is operational.

I. **Problem resolved?** If you have found and corrected the LAN card problem, stop. If not, call your HP representative for help. Be prepared to discuss the problem as described in "Contacting your HP Representative" at the end of this chapter.

Flowchart 3: Configuration Test — cont.

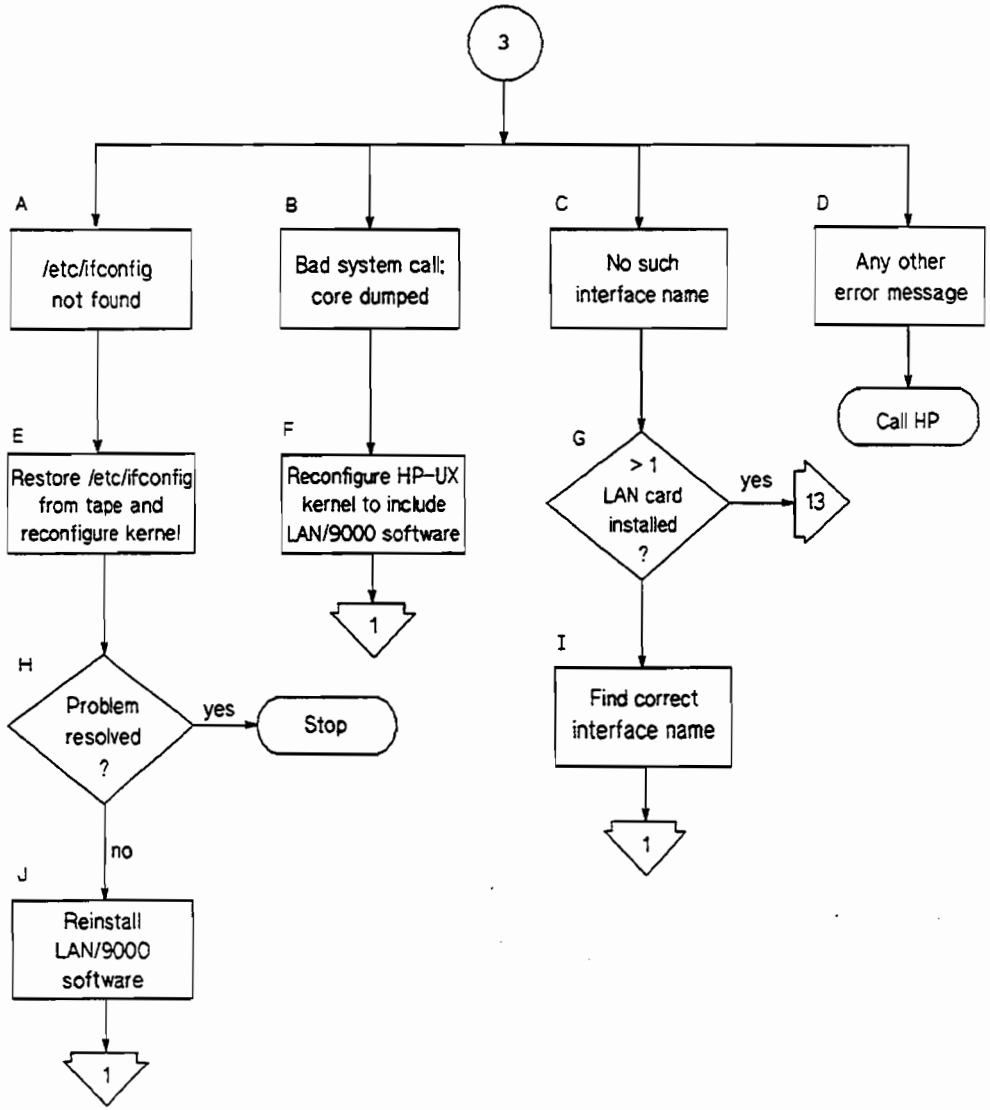

Figure 5-6. Flowchart 3

Flowchart 3 Procedures

A. **/etc/ifconfig not found.** The command has been relocated on the system or deleted.

B. **Bad system call; core dumped.** Networking is not configured into the HP-UX kernel.

C. **No such interface.** The interface name passed to *ifconfig* does not exist on the system. Check spelling and names of interfaces on the system using *netstat -i*.

If you have more than one LAN card, make sure the number of LAN cards has been configured into the kernel and that an *ifconfig* command has been executed for each.

D. **Any other error message.** If you received an error message not listed on this flowchart, interpret the message and take the appropriate action. If you need assistance, call your HP representative. Be prepared to discuss the problem as described in "Contacting Your HP Representative" at the end of this chapter.

E. **Restore /etc/ifconfig from tape and reconfigure kernel.** You can restore *ifconfig* from the last good backup tape or your install/update tape.

F. **Reconfigure HP-UX kernel to include LAN/9000 software.**

G. **>1 LAN card installed?** If you have installed more than one LAN card, go to Flowchart 13.

H. **Problem resolved?** If so, stop. If not, re-install the entire LAN/9000 software product. Start again with Flowchart 1, as necessary.

I. **Find correct interface name.** Using the correct interface name, start again with Flowchart 1.

J. **Reinstall LAN/9000 software.** Re-install the entire LAN/9000 software product. If necessary, start again with Flowchart 1.

Flowchart 4: Network Level Loopback Test

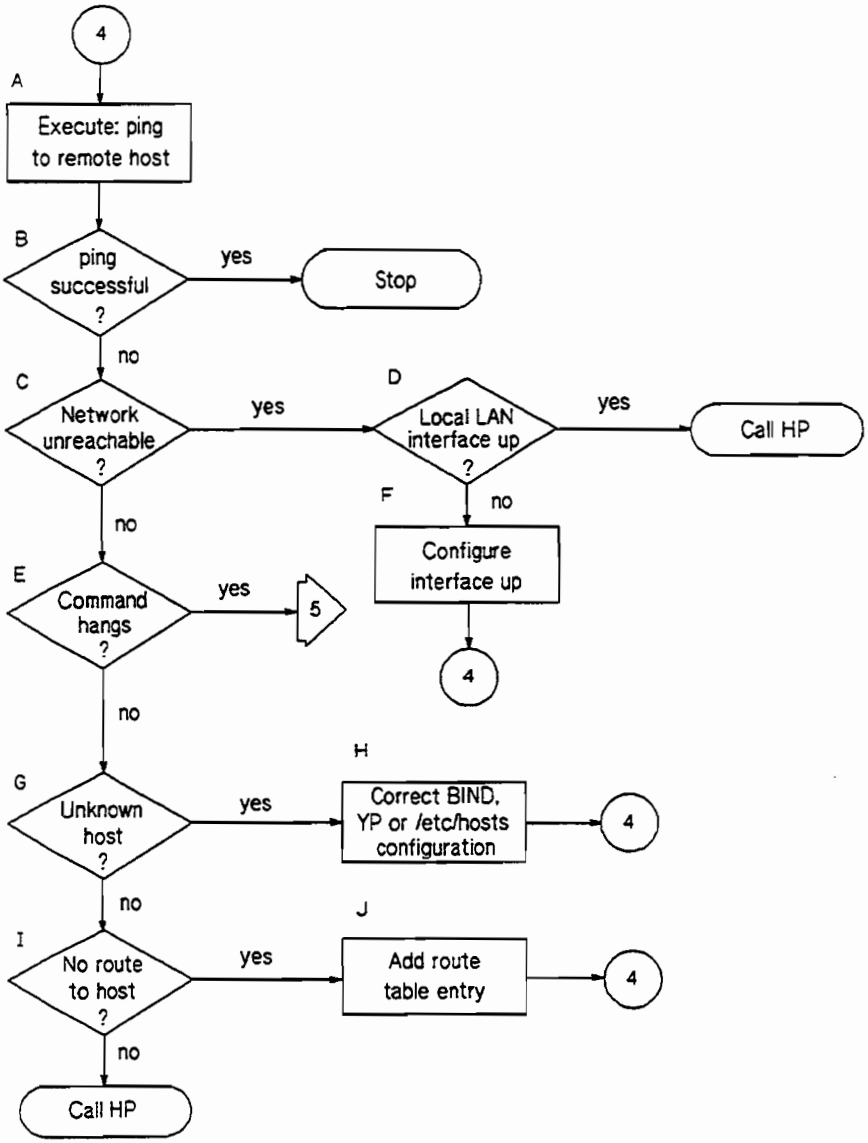

Figure 5-7. Flowchart 4

Flowchart 4 Procedures

A. **Execute: ping to remote host.** Using *ping(1M)*, send a message to the remote host you are having problems connecting to. For example, suppose this host is known to your host by the alias "bunny." Enter:

```
/etc/ping bunny
```

B. **ping successful?** A message is printed on *stdout* for each *ping* packet returned by the remote host. If packets are being returned, your system has network level connectivity to the remote host.

You may find it useful to note what percentage of the total packets are lost, if any. Losing ten percent or more may indicate the network or remote host is extremely busy. If, over a one-day period, *ping* reports a packet loss that you feel is unacceptable, yet connectivity remains, report this to your HP representative.

You may also find it useful to note the round-trip transmission times. Periodically high transmission times may indicate that the network or remote host is extremely busy. Consistently high transmission times may indicate the local host is extremely busy. Make sure that the network event logging masks are not set to values which can impair system performance (such as *DEWRP*).

C. **Network unreachable?** If so, check the status of the local LAN interface first.

D. **Local LAN interface up?** Execute *ifconfig* on the local interface to be sure it is configured up.

E. **Command hangs?** If a message is not returned after executing *ping*, go to Flowchart 5.

F. **Configure interface up.** If you find the local interface is not up, execute *ifconfig* with the appropriate flags set. Start again with Flowchart 4.

G. **Unknown host?** Error= Unknown host hostname?

H. **Correct BIND, NIS or /etc/hosts configuration.** Add the missing host name and start again with Flowchart 4.

I. **No route to host?** Error= Sendto: No route to host? If so, go to J. Otherwise, call your HP representative for help. Be prepared to discuss the problem as described in "Contacting Your HP Representative" at the end of this chapter.

J. **Add route table entry.** Using */etc/route*, add a route table entry for that host.

Flowchart 5: Network Level Loopback Test — cont.

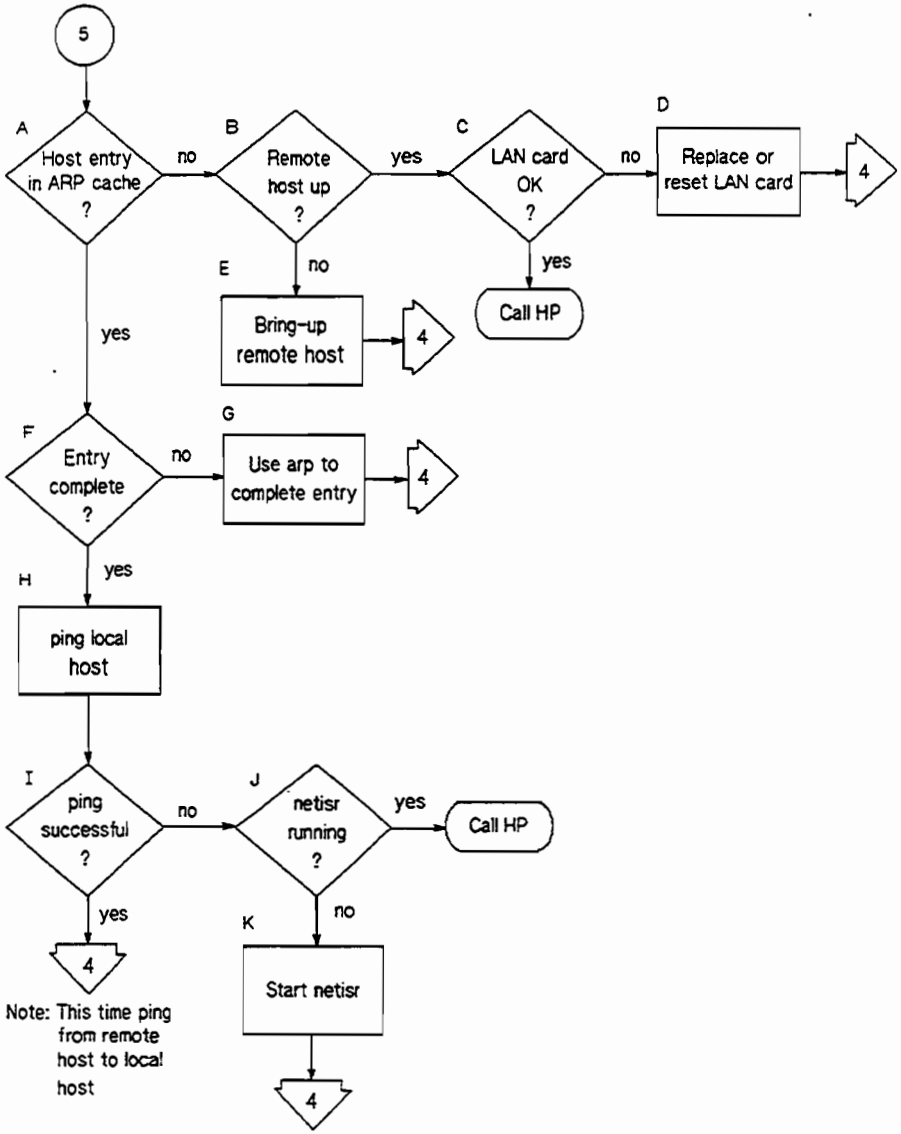

Figure 5-8. Flowchart 5

Flowchart 5 Procedures

A. **Host entry in ARP cache?** Using *arp*, check that an entry exists for the remote host in your system's ARP cache. For example, suppose the remote host is known to your system by the alias "bunny." Enter:

```
/etc/arp bunny
```

B. **Remote host up?** If there is no ARP cache entry for the remote host, first check that the remote host is up. If not, the remote host has not broadcast an ARP message, and that likely is why there is no entry in the ARP cache.

C. **LAN card O.K.?** Use *landiag* (Series 300/400, Series 600/800, or Series 700) or *LANDAD* (Series 600/800 or Series 700 only) to ensure the LAN card is operational.

D. **Replace or reset LAN card.** When the LAN card is operational, use *landiag (1M)* to reset. (Refer to Flowchart 4.)

E. **Bring-up remote host.** Have the node manager of the remote host bring that system up.

F. **Entry complete?** Perhaps there is an ARP cache entry, but it is wrong or not complete.

G. **Use arp to complete entry.** Using *arp*, enter the correct Station Address. For more information, refer to the *arp(1M)* manual page.

H. **ping local host.** Using *ping*, do an internal loopback on your own system. In other words, ping your own system. This will find if the problem is on your end.

I. **ping successful?** If the internal loopback is successful, your system is operating properly to the Network Layer (OSI Layer 3). In addition, you know an ARP cache entry for the remote host exists on your system. If this is true, the network interface or software on the remote host is suspect. Start again with Flowchart 4, but this time *ping* from the remote host to your system.

If the *ping* in Step H was not successful, go to J.

J. **netisr running?** Use the *ps* command to check that *netisr* is an active process on your host.

O. **Start netisr.** On the Series 300/400, if the *netisr* daemon is not running on your system, be sure that it is installed and execute it. On the Series 600/800 and Seris 700, if the *netisr* daemon is not running, make sure that it is configured as an interrupt. Check the *uxgen* file for the line `netisr_priority -1`.

Flowchart 6: Transport Level Loopback Test (using rlb)

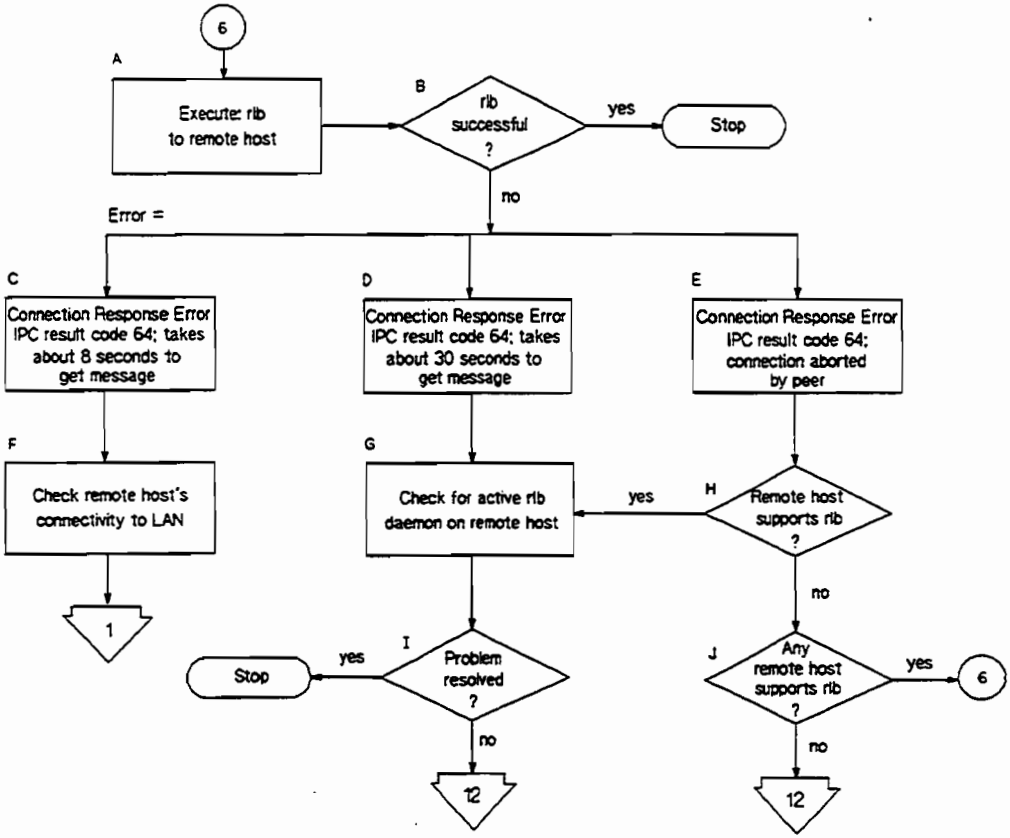

Figure 5-9. Flowchart 6

Flowchart 6 Procedures

A. **Execute: rlb to remote host.** Enter the *rlb* remote mode and use the *single* command to send a test message to the remote host you are having trouble connecting to. For more information on *rlb*, refer to Chapter 6 or the *rlb(1M)* manual page. You can only use *rlb* if the NetIPC fileset has been installed.

B. **rlb successful?** If the test was successful, stop. Network connectivity is o.k. through the Session Layer (OSI Layer 5). Your problem is not with the LAN or network interface on either host. If the test is not successful, note which error was returned and continue with this flowchart.

C. **Connection Response Error IPC result code 64; takes about 8 seconds to get message.** The remote host does not respond to the *rlb* message. It takes about 8 seconds for the error code to appear.

D. **Connection Response Error IPC result code 64; takes about 30 seconds to get message.** The remote host does not respond to the *rlb* message. It takes about 30 seconds for the error code to appear.

E. **Connection Response Error IPC result code 64; message returned within a few seconds.** The remote host does not respond to the *rlb* message. The error code appears right away.

F. **Check remote host's connectivity to LAN.** Check that the remote host is configured correctly and its network interface is up.

G. **Check for active rlb daemon on remote host.** Verify that the *rlbdaemon* is installed and active on the remote host. You can check for an active daemon by executing *ps -ef|grep rlbdaemon* on the remote host. If the only message returned is *grep rlbdaemon*, the daemon is not active.

H. **Remote host supports rlb?** Verify that the remote host has *rlbdaemon* installed. If true, check that it is active on the remote host.

I. **Problem resolved?** If so, stop. However, if the remote *rlbdaemon* is active and you still get a connection response error, go to Flowchart 12 to test LAN connections.

J. **Any remote host supports rlb?** If there are other hosts on the network which support *rlb*, restart this flowchart with one of these systems as the target host. If no other hosts on the network support *rlb*, go to Flowchart 12 to test LAN connections.

Flowchart 7: Transport Level Loopback Test (using ARPA)

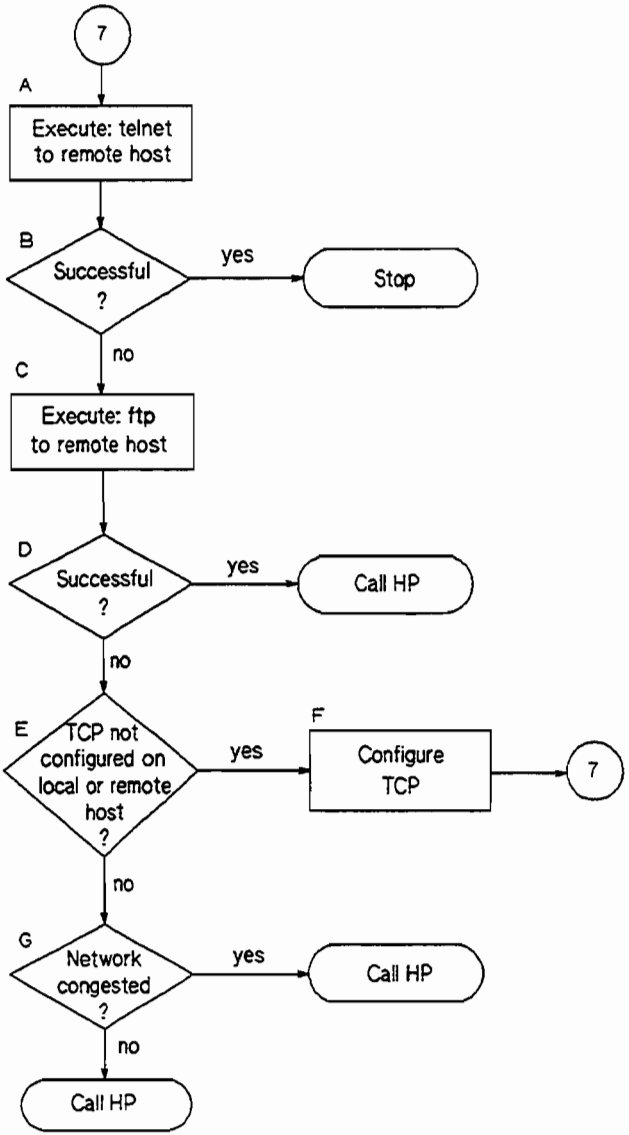

Figure 5-10. Flowchart 7

Flowchart 7 Procedures

A. **Execute: telnet to remote host.** Try to establish a *telnet* connection to the remote host.

B. **Successful?** If your *telnet* attempt was successful, stop. The connection is o.k. through the Transport Layer (OSI Layer 4).

C. **Execute: ftp to remote host.** Unlike *telnet*, *ftp* does not go through a pseudoterminal driver (pty) on your system. This step tests to see if the pty is why *telnet* failed.

D. **Successful?** If *ftp* is successful, you likely have a problem with a pty on your system. Contact your HP representative.

E. **TCP not configured on local or remote host?** Neither *telnet* or *ftp* will work if TCP is not configured on either side of the connection. Check the */etc/protocols* file on both hosts to be sure TCP is installed and configured.

F. **Configure TCP.** If necessary, install TCP on either or both hosts.

G. **Network congested?** If TCP is installed on both hosts, do a file transfer to another remote host on the network. Use *netstat* to check for lost packets.

If 10 percent or more packets are lost, the network is extremely busy. If you cannot determine the cause, contact your HP representative for help.

If network congestion is not the cause, more detailed diagnostics are required. Again, contact your HP representative.

Flowchart 8: Link Level Loopback Test

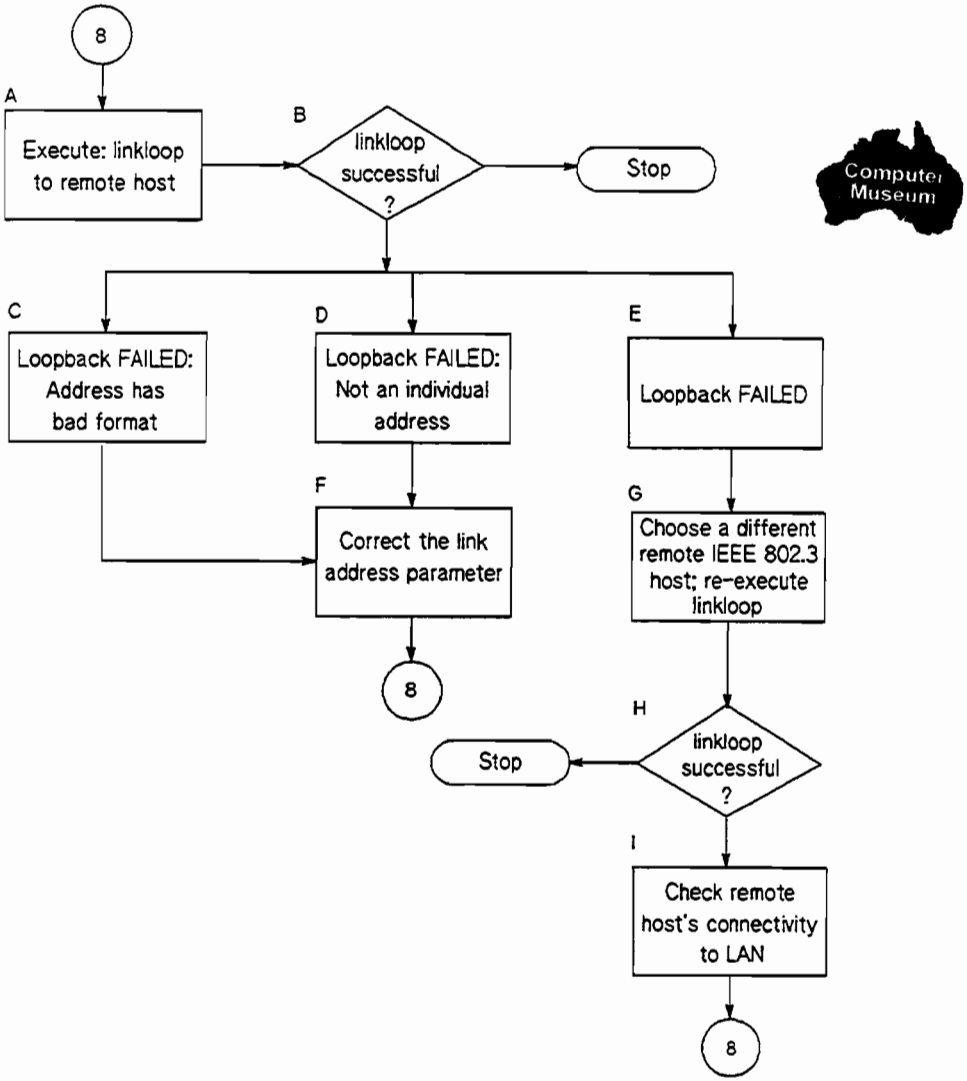

Figure 5-11. Flowchart 8

Flowchart 8 Procedures

A. **Execute: linkloop to remote host.** Enter the link level address (station address) of the remote host in hexadecimal form (preceded by "0x"). Execute *lanscan (1M)* to find the link level address (station address) on the remote host or obtain it from your network map. For more information on *linkloop*, refer to Chapter 6.

B. **linkloop successful?** If the test was successful, stop. Network connectivity is o.k. through the Link Layer (OSI Layer 2). If not successful, note which error was returned and continue with this flowchart.

C. **Loopback failed; Address has bad format.** The link level address is not correct. Go to H.

D. **Loopback failed; Not an individual address.** The link level address is not correct. The second hexadecimal digit is odd. This means it is a multicast or broadcast address, which is not allowed. The address must be unique to one remote host. Go to H.

E. **Loopback failed.** The remote host did not respond. Go to I.

F. **Correct the link address parameter.** Change the link level address to an allowed value and start again with Flowchart 8.

G. **Choose a different IEEE host; re-execute linkloop.** Restart this flowchart using a different remote host.

H. **linkloop successful?** If the test was successful, stop. Network connectivity is o.k. through the Link Layer (OSI Layer 2). If not successful, go to K.

I. **Check remote host's connectivity to LAN.** Contact the node manager of the remote host. Check that the host is configured correctly and that its network interface is up. If necessary, use Flowcharts 1 and 12 to verify configuration and connectivity of the remote host.

Flowchart 9: LAN Card Test (Series 300/400 only)

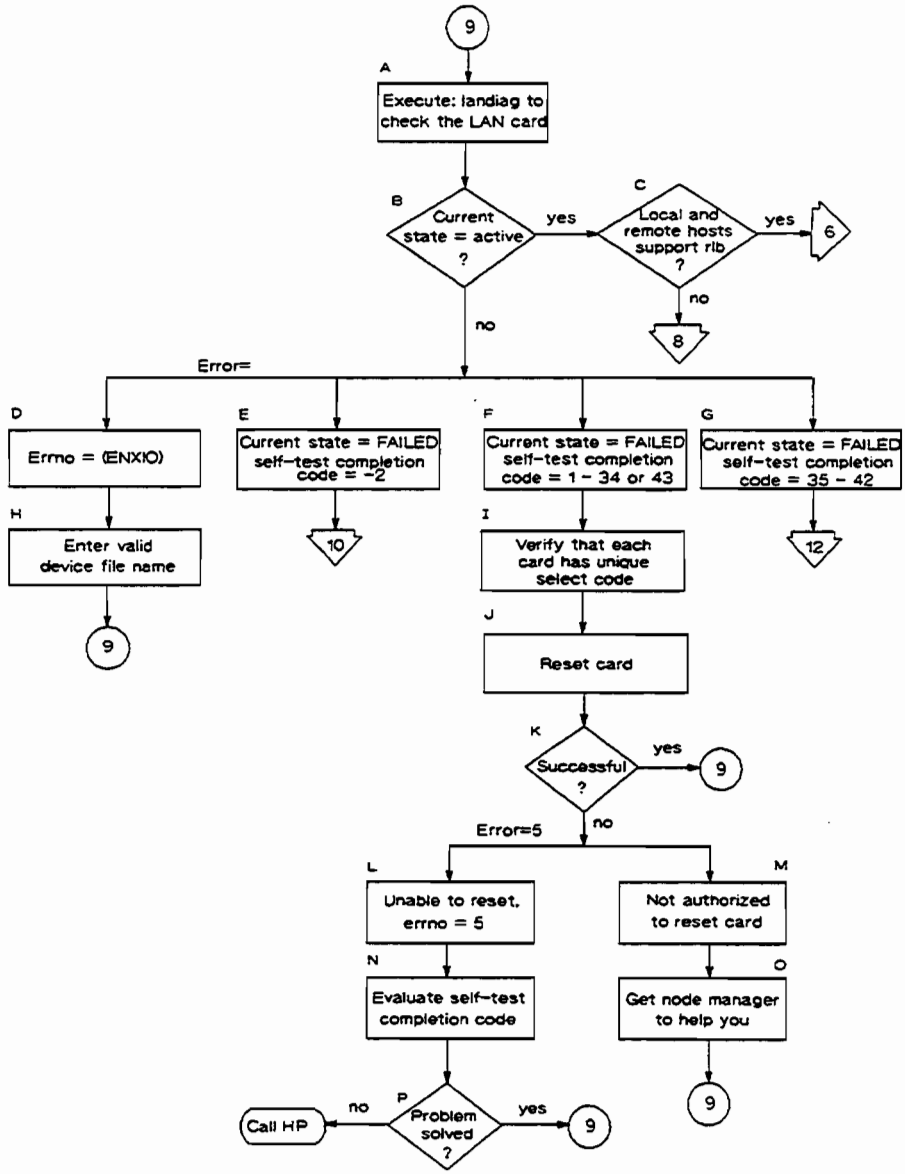

Figure 5-12. Flowchart 9

Flowchart 9 Procedures

A. **Execute landiag to check the LAN card.** Enter the *landiag* lan mode and use the *display* command to check LAN card status. For more information on *landiag*, refer to Chapter 6.

B. **Current state = active?** If the LAN card is active (o.k.), go to C. If the LAN card is not active, note which error message was returned and continue with this flowchart.

C. **Local and remote hosts support rlb?** If the *rlbdaemon* is installed on both local and remote hosts, you may use *rlb* to test connectivity through the Transport Layer (OSI Layer 4). Refer to Flowchart 6.

D. **Errno=(ENXIO).** The device file used by *landiag* does not correspond to an active LAN card. Using the *name* command, enter a valid device file name and start again with Flowchart 9.

E. **Current state = FAILED.** The LAN card is not present or is not configured correctly. Go to Flowchart 10.

F. **Current state = FAILED, selftest completion code = 1-34 or 43.** If the self-test completion code value is 1 to 34 or 43, the LAN card has a hardware failure.

G. **Current state = FAILED, selftest completion code = 35 - 42.** If the self-test completion code is 35 to 42, there is an external loopback failure. Refer to Appendix E for more information on the specific completion code you receive. Go to Flowchart 12 to check LAN connections.

H. **Enter valid device file name.** Correct the device file name and start again with this flowchart.

I. **Verify that each card has unique select code.** Verify that there are no two cards with the same select code.

J. **Reset card.** This re-executes the LAN card self-test.

K. **Successful?** If the test was successful, start again with this flowchart to display LAN card statistics.

L. **Unable to reset, errno = (EIO).** This indicates a problem in resetting the LAN card.

M. **Not authorized to reset card.** You must have super-user capability to reset the LAN card.

N. **Evaluate selftest completion code.** Look up the self-test completion code in Appendix E and try to correct the problem.

O. **Get the node manager to help you.**

P. **Problem solved?** If so, start again with Flowchart 9. If not, contact your HP representative. Be prepared to discuss the problem as described in "Contacting Your HP Representative" at the end of this chapter.

Flowchart 10: LAN Card Test (Series 300/400 only) — cont.

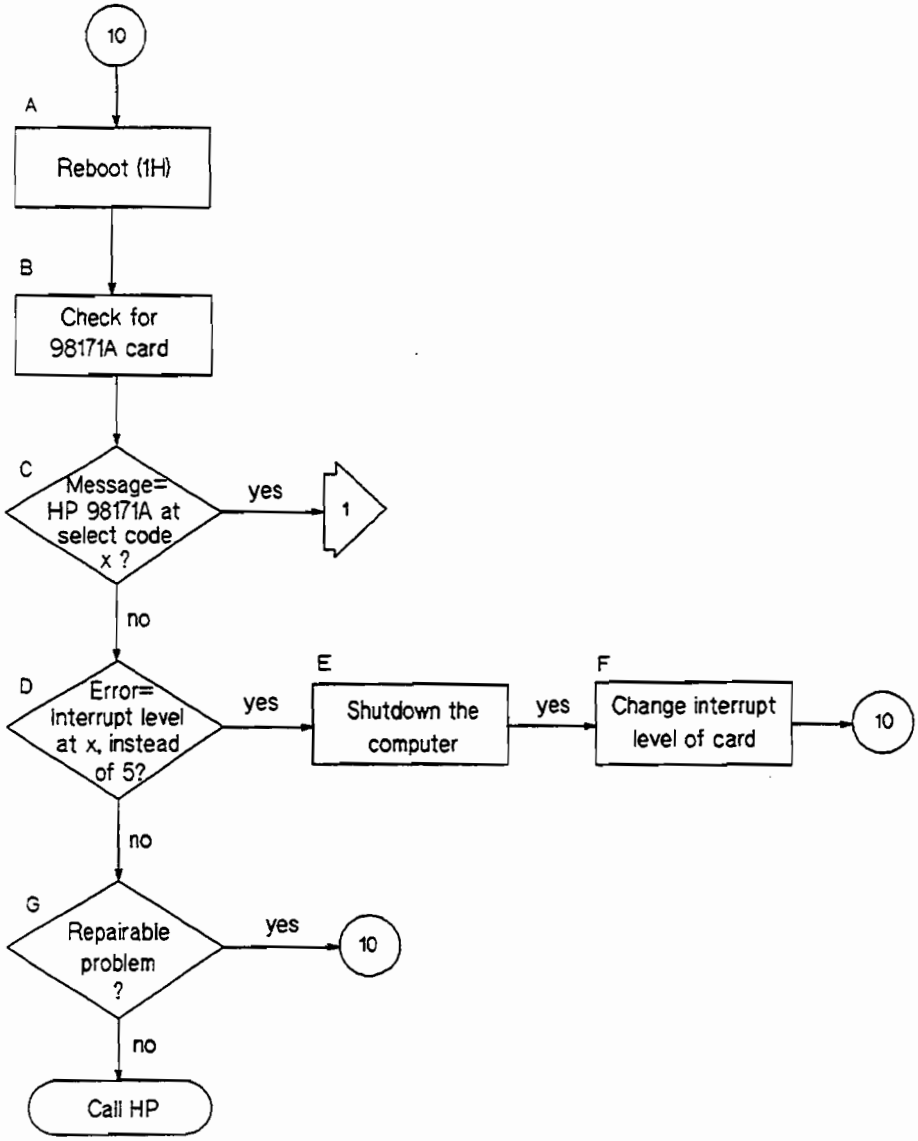

Figure 5-13. Flowchart 10

Flowchart 10 Procedures

A. **Reboot (1H).** Enter the *landiag* lan mode and use the *display* command to check LAN card status. For more information on *landiag*, refer to Chapter 6.

B. **Check for 98171A card.** When HP-UX boots up, it identifies all the interface cards. Look for the 98171A card again. Following is an example of part of an HP-UX boot display:

```
HP-IB at select code 7
P 98626 at select code 9
HP 98625A at select code 14
HP 98171A at select code 21
HP 98620B
real memory = 2086912
```

C. **Message = HP 98171 at select code x?** If this system message appears (with x indicating the actual select code), the interface cards and driver are installed correctly.

D. **Error = Interrupt level at x instead of 5?** If the system message is "HP 98171A at select code 21 ignored, Interrupt level at x instead of 5," the interrupt level of the card is not correct.

E. **Shutdown the computer.** Turn off the power so you can remove the LAN card.

F. **Change interrupt level of card.** The interrupt level switch must be changed. Refer to the installation manual for you LAN card for details. After you have made the change, start again with Flowchart 10.

G. **Repairable problem?** If you receive an error message that is not described in this flowchart, try to interpret the message. If you think you found a solution, start again with this flowchart to reboot. If the problem is not fixed, contact your HP representative for help. Be prepared to discuss the problem as described in "Contacting Your HP Representative" at the end of this chapter.

Flowchart 11: LAN Card Test (Series 600/800 and Series 700 only)

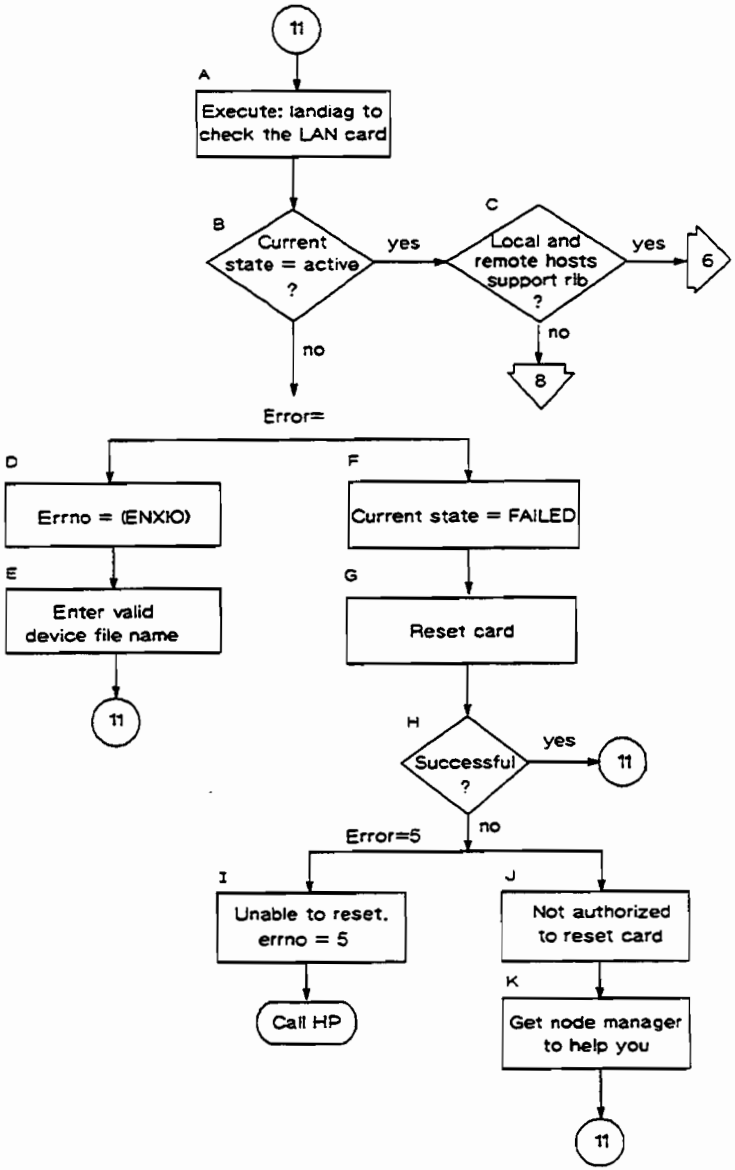

Figure 5-14. Flowchart 11

Flowchart 11 Procedures

A. **Execute landiag to check the LAN card.** Enter the *landiag* lan mode and use the *display* command to check LAN card status. For more information on *landiag*, refer to Chapter 6.

B. **Current state = active?** If the LAN card is active (o.k.), go to C. If the LAN card is not active, note which error message was returned and continue with this flowchart.

C. **Local and remote hosts support rlb?** If the *rlbdaemon* is installed on both local and remote hosts, you may use *rlb* to test connectivity through the Transport Layer (OSI Layer 4). Refer to Flowchart 6.

D. **Errno=(ENXIO).** The device file used by *landiag* does not correspond to an active LAN card. Using the *name* command, enter a valid device file name and start again with Flowchart 9.

E. **Enter valid device file name.** Correct the device file name and start again with this flowchart.

F. **Current state = FAILED.** The LAN card is not present or is not configured correctly. Go to Flowchart 10.

G. **Reset card.** This re-executes the LAN card self-test.

H. **Successful?** If the test was successful, start again with this flowchart to display LAN card statistics.

I. **Unable to reset, errno = (EIO).** This indicates a problem in resetting the LAN card.

J. **Not authorized to reset card.** You must have super-user capability to reset the LAN card.

K. **Get the node manager to help you.**

Flowchart 12: LAN Connections Test

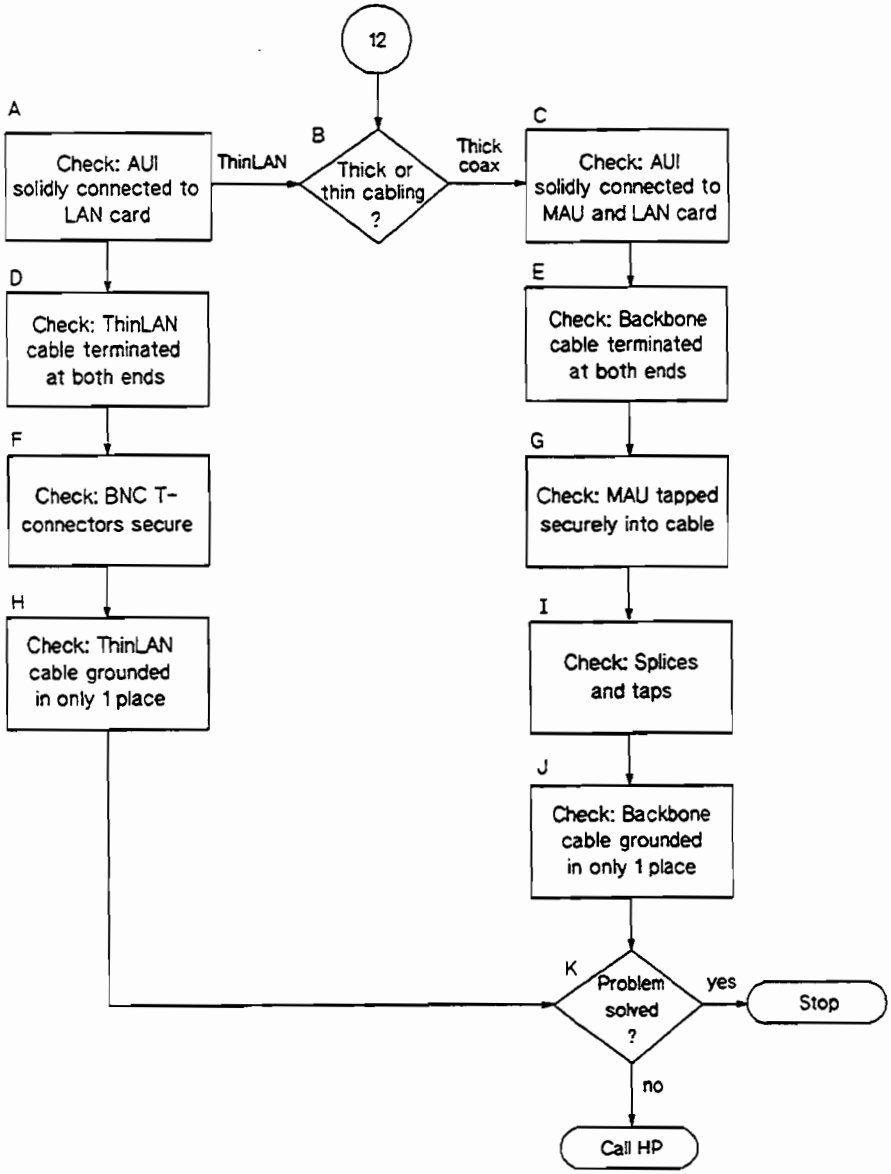

Figure 5-15. Flowchart 12

Flowchart 12 Procedures

A. **Check: AUI solidly connected to LAN card.** Make sure the AUI cable is solidly connected to the LAN card. If the AUI cable is not connected, turn off the power to the computer before you connect it.

B. **Thick or thin cabling?** If your network cabling is the thicker coaxial cabling, continue in the direction marked "thick." If your network cabling is the ThinLAN cabling, continue in the direction marked "thin."

C. **Check: AUI solidly connected to MAU and LAN card.** Make sure the AUI cable is solidly connected to the MAU and the LAN card. If the AUI cable is not connected, turn off the power to the computer before you connect it.

D. **Check: ThinLAN cable terminated at both ends.** Make sure the backbone cable is terminated at both ends.

E. **Check: Backbone cable terminated at both ends.** Make sure the backbone cable is terminated at both ends.

F. **Check: BNC T-connectors secure.** Make sure each BNC T-connector is securely attached to a BNC connector on the ThinLAN cable and that no intervening cable is between the MAU and the T-connector.

G. **Check: MAU tapped securely into cable.** Make sure the MAU is tapped securely into the backbone cable.

H. **Check: ThinLAN cable grounded in only one place.** Make sure the ThinLAN cable is grounded in only one place.

I. **Check: Splices and Taps.** Make sure all splices and taps are secure.

J. **Check: Backbone cable grounded in only one place.** Make sure the backbone cable is grounded in only one place.

K. **Problem solved?** If so, stop. If you still have a problem after working through this flowchart, you may have a failed LAN card, or a problem with the transmit or receive function of the MAU. Contact your HP representative for help. Be prepared to discuss the problem as described in

"Contacting Your HP Representative" at the end of this chapter.

Flowchart 13: Gateway Configuration Test

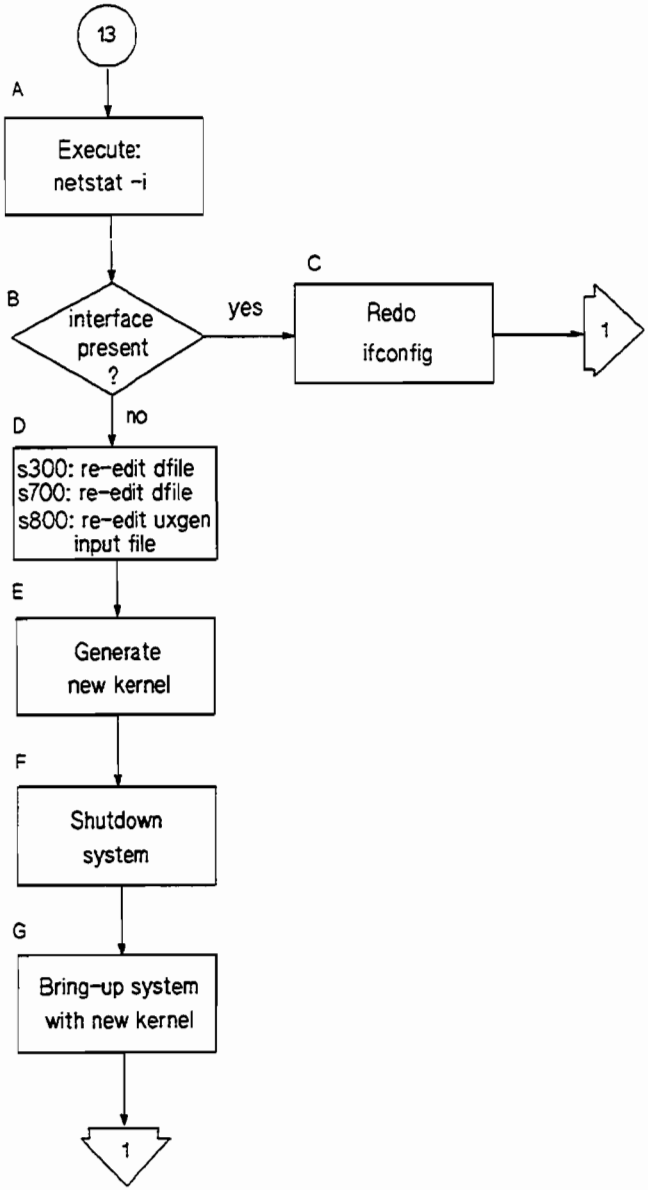

Figure 5-16. Flowchart 13

Flowchart 13 Procedures

A. **Execute netstat -i.** Check that the network interface exists. If it does, but as a logical unit you do not expect, go to C to reassign the network interface. If the network interface does not exist, proceed to step D.

C. **Redo ifconfig(1M).** Specify the network interface returned in step A. Start again with Flowchart 1.

D. **s300: re-edit dfile; s700: re-edit dfile; s800: re-edit uxgen input file.** Add entries for an extra LAN card. Refer to Chapter 3 for details.

E. **Generate new kernel.** Generate a new kernel and go to G.

F. **Shutdown system.** Shutdown the system and go to F.

G. **Bring-up system with new kernel.** Bring up the system and start again with Flowchart 1.

Flowchart 14: Gateway Loopback Test

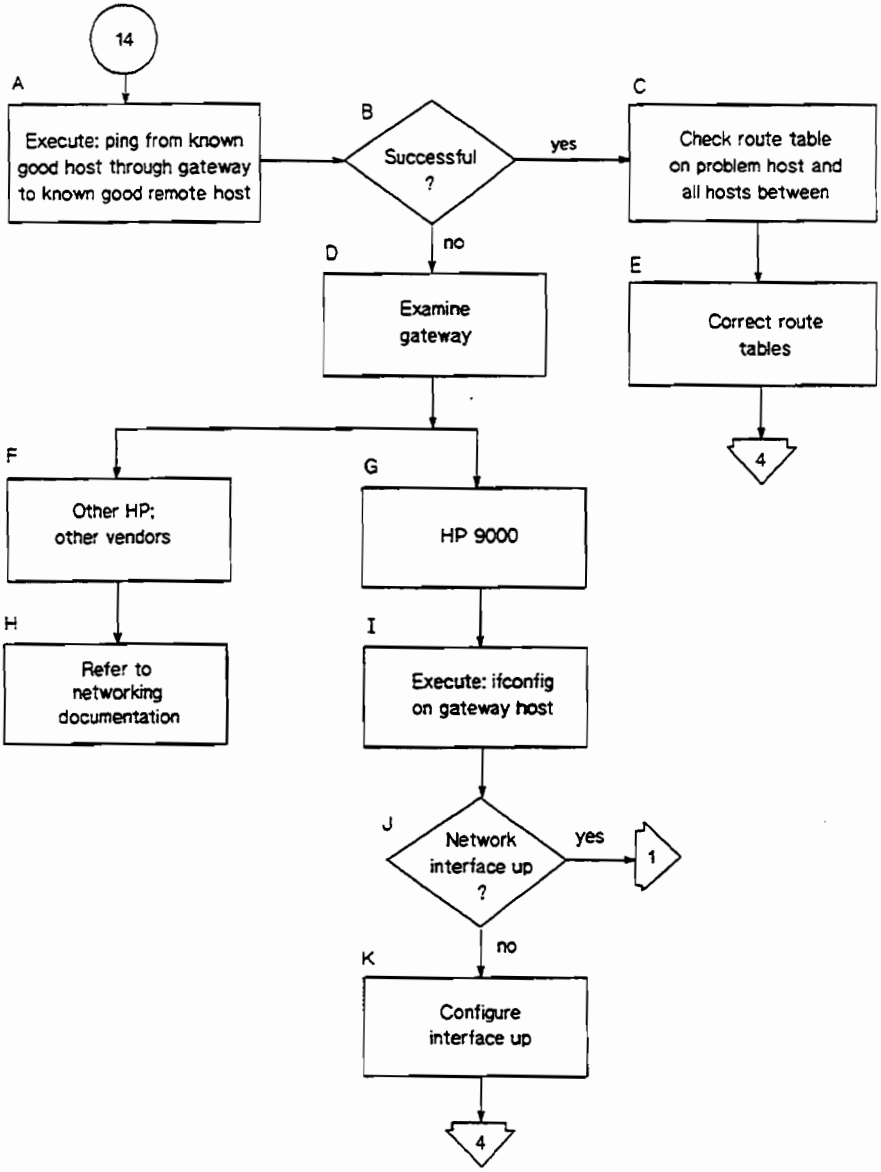

Figure 5-17. Flowchart 14

Flowchart 14 Procedures

A. **Execute: ping from known good host through gateway to known good host on remote network.** This will test gateway connectivity to the remote network. For more information on *ping(1M)*, refer to Chapter 6.

B. **Successful?** If the executing *ping* returned successfully, the problem may exist in the routing table for the problem host. Go to C.

C. **Check route table on the problem host and all hosts between.** Execute *netstat -r* to examine a route table.

D. **Examine gateway.** If the gateway is an HP 9000, go to G. If it is not, go to F.

E. **Correct route tables.** Ensure that the proper IP addresses are assigned in the *Destination* and *Gateway* fields. If you are using subnetting, make sure that the destination is what you expect: a network or a host.

F. **Other HP; other vendors.** Go to H.

G. **HP 9000.** Go to I.

H. **Refer to networking documentation.** Refer to the documentation that came with the gateway for additional diagnostics.

I. **Execute: ifconfig on gateway host.** Execute *ifconfig* for all network interfaces on the gateway.

J. **Network interface up?** If the output from *ifconfig* does not include the *UP* parameter, the network interface is down. Execute *netstat -i* to check the status of the network interfaces. An asterisk (*) next to the interface indicates that the interface is down.

If the network interface is down, go to K. If the network interfaces are *UP*, start again with Flowchart 1. Using Flowchart 1, test all network interfaces on the gateway.

Use *lanconfig* to make sure `ieee` or `ether` encapsulation is configured.

Note *Running* is always displayed. It indicates only that there is OS support for the interface.

K. **Configure interface up.** Execute *ifconfig* on each interface to bring it up. Start again with Flowchart 1. Using Flowchart 1, test all network interfaces on the gateway.

Flowchart 15: Probe Proxy Server Test

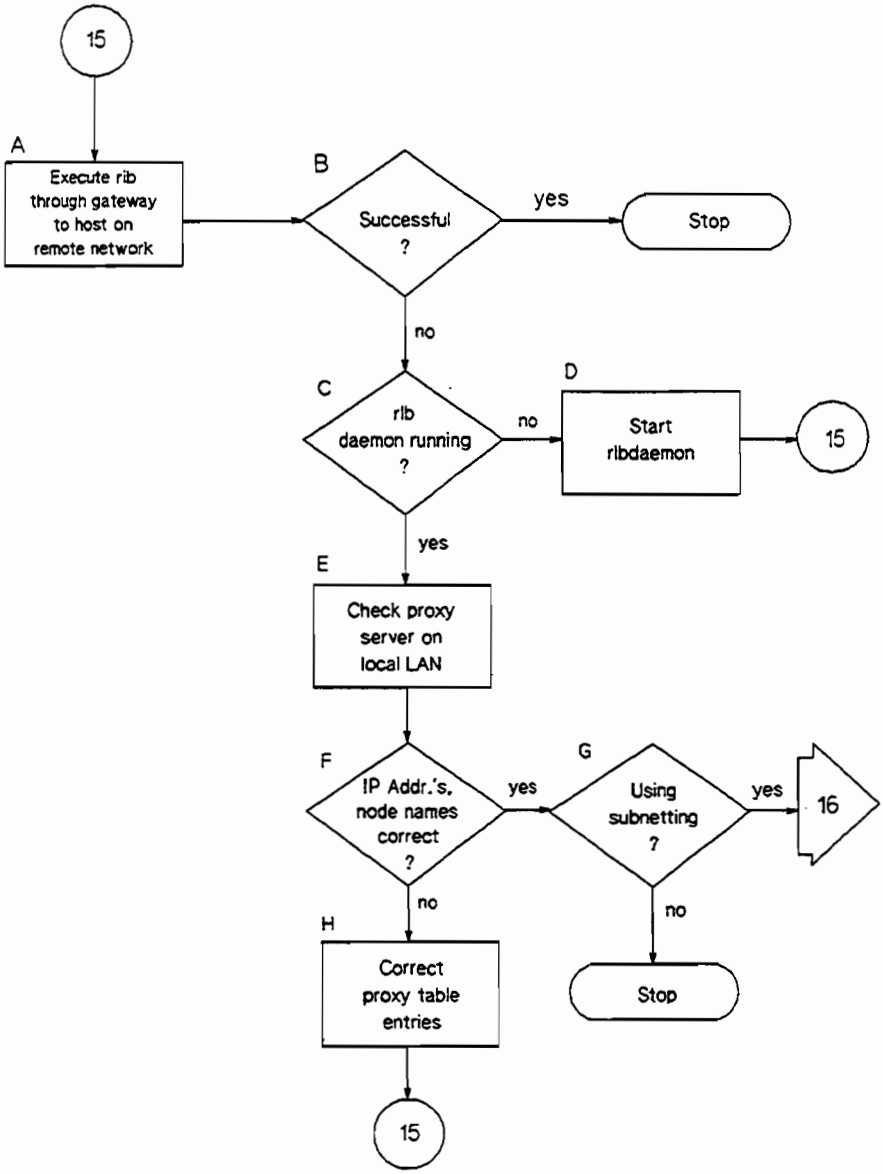

Figure 5-18. Flowchart 15

Flowchart 15 Procedures

A. **Execute rlb through gateway to host on remote network.** This tests connectivity through the gateway.

B. **Successful?** If the *rlb* test through the gateway succeeds, stop with this test. The problem is likely in the network service executing at the time of difficulty. Refer to the manual provided with the network service.

C. **rlb daemon running?** Execute the *ps -ef | grep rlb* command. If only the *grep* entry is returned, then the daemon is not running. Go to D. If an entry for the *rlbdaemon* is returned, go to E.

D. **Start rlbdaemon.** Execute */etc/rlbdaemon*. You must have super-user capability to do so. Start again with Flowchart 15.

E. **Check: proxy server on local LAN.** Execute the proxy list command on the Probe proxy server node on your LAN. Go to F.

F. **IP Addr.'s, Node names correct?** Are the IP addresses and the node names what you expect? Execute *nodename* on the problem node and check your network map to ensure the node names and IP addresses are correct. If the IP addresses and node names are not correct, go to H. If the IP addresses and node names are correct, go to G.

G. **Using subnetting?** If you are using subnetting on your network, go to Flowchart 16. If not, stop this test. You may have found an error in Probe Proxy Server software. Contact your HP representative.

H. **Correct proxy table entries.** Execute the *proxy(1M)* command. The *proxy(1M)* command is described in *Installing and Administering NS/9000*.

Flowchart 16: Subnet Test

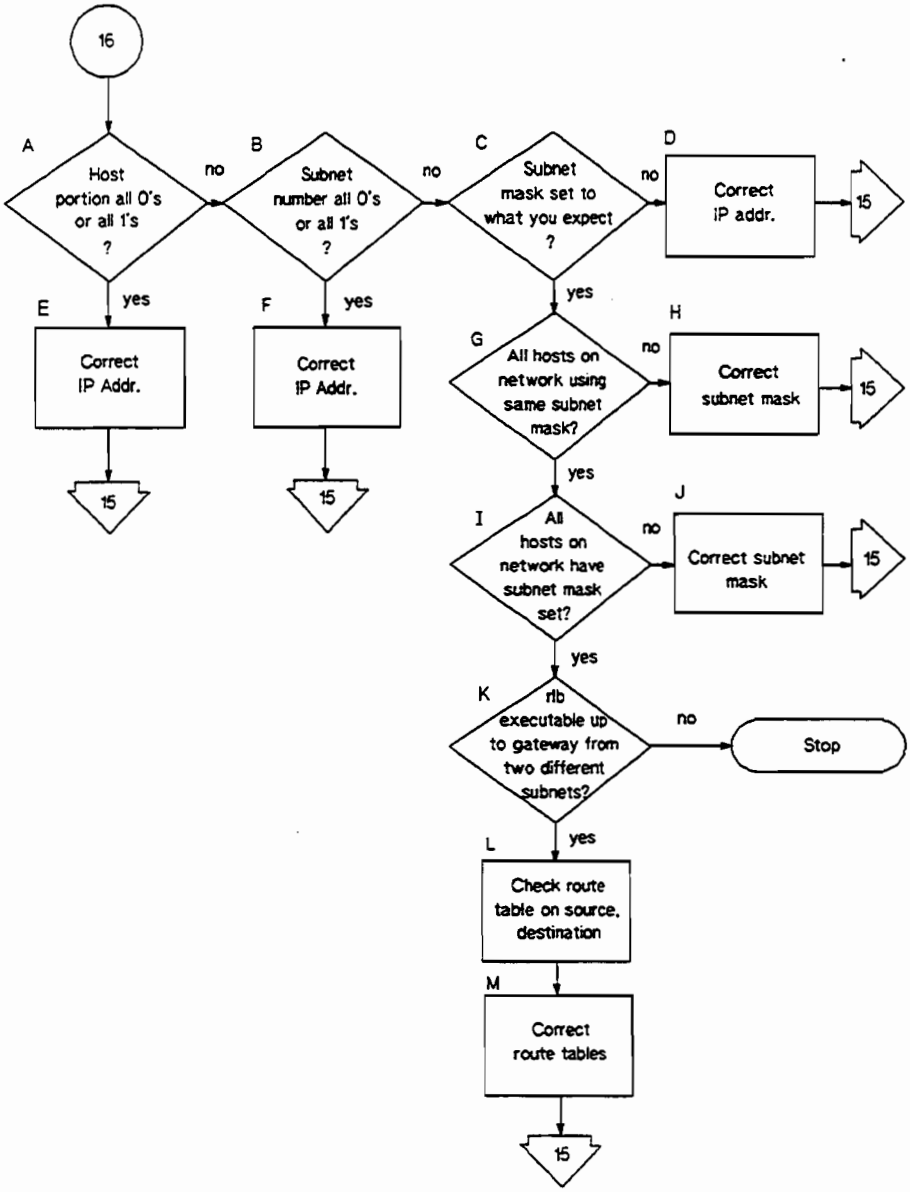

Figure 5-19. Flowchart 16

Flowchart 16 Procedures

A. **Host Portion all 0's or all 1's?** Execute *ifconfig(1M)*. Is the host portion of the IP address all 0's or all 1's? These values are reserved. Refer to Chapter 2 for details on subnets. If the host portion of the IP address is all 0's or all 1's, go to E to correct the IP address. Otherwise, go to B to examine the subnetwork number.

B. **Subnet number all 0's or all 1's?** Execute *ifconfig(1M)*. Is the subnet number portion of the IP address all 0's or all 1's? These values are reserved. Refer to Chapter 2 for details on subnets. If the subnet number portion of the IP address is all 0's or all 1's, go to F correct the IP address. Otherwise, go to C to examine the subnet mask.

C. **Subnet mask set to what you expect?** Check your network map and execute *ifconfig(1M)* to determine the subnet mask for your node. Refer to Chapter 2 for details on subnets. If the subnet mask is not what you expect, go to D. Otherwise, go to G.

D. **Correct IP addr.** Set the subnet mask to the proper value. Start again with Flowchart 15.

E. **Correct IP addr.** Correct the IP address and start again with Flowchart 15.

F. **Correct IP addr.** Correct the IP address and start again with Flowchart 15.

G. **All hosts on network using same subnet mask?** Execute *ifconfig(1M)* for every network interface on each node on the entire network. If all nodes are using the same subnet mask, go to I. Otherwise, go to H to correct the subnet masks.

H. **Correct subnet mask.** To do so, execute *ifconfig* with the proper subnet mask. Start again with Flowchart 15.

I. **All hosts on network have subnet mask set?** Execute *ifconfig* for every network interface on each node on the entire network. If all nodes have the same subnet mask set, go to K. Otherwise, go to J to set the correct subnet masks.

J. **Correct subnet mask.** To do so, execute *ifconfig* with the proper subnet mask. Start again with Flowchart 15.

K. **rlb executable up to gateway from two different subnets?** If you can communicate via *rlb(1M)* up to the gateway node from two different subnetworks, go to L to check the route tables on the non-gateway nodes. Otherwise, stop; you may have isolated an internal software error. Contact your HP representative.

L. **Check route table on source, destination.** Execute *netstat -r* on the two hosts used in the *rlb* commands executed in K above. Go to M.

M. **Correct the route tables** (if necessary). In general, specify a *net*, not a *host* when adding to the route table. Specifying a network as the destination enables you to add nodes to the remote destination subnetwork without updating the route tables on the local subnetwork every time you add a node to the remote subnetwork. Start again with Flowchart 15.

Contacting Your HP Representative

If you have no service contract with HP, you may follow the procedure described below, but you will be billed accordingly for time and materials.

If you have a service contract with HP, document the problem as an Service Request (SR) and forward it to your HP representative. Include the following information where applicable:

- A characterization of the problem. Describe the events leading up to and including the problem. Attempt to describe the source of the problem. Describe the symptoms of the problem and what led up to the problem.

  Your characterization should include: HP-UX commands; communication subsystem commands; job streams; result codes and messages; and data that can reproduce the problem.

  Illustrate as clearly as possible the context of any message(s). Prepare copies of information displayed at the system console and user terminal.

- Obtain the version, update and fix information for all software. To check your ARPA, NS or LAN/9000 version, execute the *what service_name* command, where *service_name* is a network service specific to the networking product such as *dscopy(1)* for NS and *ftp(1)* for ARPA Services/9000.

  To check the version of your kernel, execute *uname -r*.

  This allows HP to determine if the problem is already known, and if the correct software is installed at your site.

- Record all error messages and numbers that appear at the user terminal and the system console.

- Save all network log files.

  Prepare the formatted output and a copy of the log file for your HP representative to further analyze.

- Prepare a listing of the HP-UX I/O configuration you are using for your HP representative to further analyze.

- Try to determine the general area within the software where you think the problem exists. Refer to the appropriate reference manual and follow the guidelines on gathering information for that product.

- Document your interim, or "workaround" solution. The cause of the problem can sometimes be found by comparing the circumstances in which it occurs with the circumstances in which it does not occur.
- Create copies of any ARPA, NS or LAN/9000 link trace files that were active when the problem occurred for your HP representative to further analyze.
- **In the event of a system failure, a full memory dump must be taken.** Use the HP-UX utility */etc/savcore* to save a core dump. Send the output to your HP representative.

6

Using Network Diagnostics

This chapter describes LAN/9000 and HP-UX diagnostics for network troubleshooting. It contains the following sections:

- Overview of Network Diagnostics.
- *netstat(1)*.
- *ping(1M)*.
- *rlb(1M)*.
- *landiag(1M)*.
- *linkloop(1)*.
- *lanscan(1M)*.
- *LANDAD*.

Note The interrupt signal is often used to terminate diagnostic utilities. This chapter assumes you have set the **[Break]** key as your interrupt character, using the *stty* flags *brkint* and *-ignbrk*. See the *stty(1)* manual reference page for details.

Overview of Network Diagnostics

LAN/9000 provides the following diagnostics:

netstat(1)	Provides network statistics and information about network connections.
ping(1M)	Verifies network connectivity through the Network Layer (OSI Layer 3). *ping(1M)* also reports round-trip time of communications between the local and remote hosts.
rlb(1M)	Verifies network connectivity through the Transport Layer (OSI Layer 4).
landiag(1M)	Resets or reports status of the LAN card.
linkloop(1M)	Verifies network connectivity through the Data Link Layer (OSI Layer 2).
lanscan(1M)	Display information about LAN cards that are successfully bound to the system.

Note For Series 600/800 models only, the HP-UX operating system provides an additional network diagnostic called *LANDAD*. *LANDAD* is part of the HP-UX On-line Diagnostic Subsystem. *LANDAD* does the same things as *linkloop* and *landiag* as well as providing additional MAU, AUI and internal loopback tests. Whenever possible, Hewlett-Packard recommends you use *LANDAD* for Series 600/800 LAN card diagnostics.

netstat(1)

The *netstat(1)* command symbolically displays network-related statistics.

Syntax

```
        [-[A][a][n]]
        [-R[n]     ]
        [-m        ]
        [-i[n]     ]
netstat [-r [n]    ] [system]
        [-rs       ]
        [-s        ]
        [interval  ]
```

Parameters

-A	Lists the address of any protocol control blocks. Used for debugging. When used with the *-a* option, includes server processes. When used with the *-n* option, displays addresses numerically. See Example 2.
-R	Lists all socket names in the socket registry. Refer to the *NetIPC Programmer's Guide* for details. When used with the *-n* option, displays addresses numerically. Used to display NetIPC information. See Example 7.
-a	Shows the state of all sockets; normally sockets used by server processes are not shown. See Example 2.
-i	Shows the state of the network interface and its attributes. If an asterisk (*) appears next to the listing for a network interface, the interface is down. See Example 1.

`-m`	Shows statistics recorded by the network memory management routines. See Example 4.
`-n`	Displays network addresses as numbers. *-n* can be used with options *-A*, *-R*, *-a*, *-i*, and *-r*, or by itself.
`-s`	Lists statistics per protocol. The protocols listed are: *tcp*, *udp*, *pxp*, *ip*, and *icmp*. See Example 5.
`-r`	Lists gateway routing information. When used with the *-s* option, the display shows routing statistics. Refer to the *route(1M)* and *routing(7)* entries in the *LAN/X.25 Reference Pages* for details on routing and gateways. See Example 3.
interval	If *interval* is specified, packet traffic statistics are reported every *interval* seconds. That is, inbound and outbound packets are counted, along with the number of errors and the number of collisions encountered since the last line was printed. Every 24th line contains a summary of the statistics since the node was last powered up or the statistics were reset with landiag. Default: one sampling.
system	Kernel you wish to examine. Default: `/hp-ux`.

Description

netstat(1) reports network and protocol statistics regarding packet traffic and the local LAN interface. Any user can execute *netstat(1)*. Some information, such as the active connections report, is useful for the day-to-day user. Protocol statistics, however, are best understood by someone familiar with network protocols.

netstat(1) can be used to:

- Display statistics associated with a LAN interface card.
- Display protocol and routing statistics.
- List the active connections.
- List network memory statistics.
- Check the states of sockets.
- Display addresses.

Connections are either active or passive. An active connection is completed when a request is made by the client and that request is accepted by the server. Both the requestor and the server see this as an active connection.

A passive connection is viewed by the server side only. When the server is waiting to accept requests the connection is considered passive. A passive connection appears in the *LISTEN* state in *netstat(1) -a* output.

Options *-A*, *-R*, *-i*, *-m*, *interval*, and *-r* cannot be used in combination. If more than one is specified, the priority is; *-m*, *interval*, *-i*, *-r*, *-R* then *-A*.

Display formats vary, depending upon the statistics presented. For active sockets, the default display shows:

- Local and remote ("Foreign") addresses.
- Send and receive queue sizes, in bytes.
- The protocol.
- The state of the protocol.

See Example 2 for descriptions of the default fields.

Symbolic address formats follow two forms: *host.port*, if a known host address is found in the data base */etc/hosts*, or *network.port* if a socket address specifies a network but no host. The network address is found in the data base */etc/networks*.

If the *-n* option is specified, the address is printed in internet format. See Chapter 2 for a detailed description of the internet format.

After installing a new kernel or updating an old one, remove the */etc/netstat_data* file. A new version of */etc/netstat_data* will be created the next time you use *netstat(1)*. If *netstat(1)* ever returns garbled statistics, remove the */etc/netstat_data* file, then execute *netstat(1)* again.

Reporting Interface Statistics (Example 1)

```
netstat -i

Name  Mtu   Network     Address   Ipkts  Ierrs  Opkts  Oerrs  Collis
lan0  1497  192.6.142.1 hpindma   10343  0      4134   0      0
lan1  1497  192.6.143.1 hpindma   10980  0      6003   0      0
lo0   1536  loopback-n  loopback  0      0      0      0      0
```

Following is a description of each field:

Field Name	Description
Name	The network interface.
Mtu	Indicates the "maximum transmission unit." This is the maximum packet size sent by the interface card. The protocols will break down larger packets sent by user-level programs if necessary. Therefore, users do not need to be concerned about this number.
Network	The network address of the LAN to which the interface card is connected. The symbolic name is entered in */etc/networks*.
Address	The IP address associated with the LAN interface. The symbolic name is the host name entered in */etc/hosts*.

Ipkts	The number of packets received by the interface card.
Ierrs	The number of errors detected on incoming packets.
Opkts	The number of packets transmitted by the interface card.
Oerrs	The number of errors detected during the transmission of packets.
Collis	The number of collisions that resulted from packet traffic.

netstat -i shows the status of the LAN interface card, or cards, and its attributes. If an asterisk (*) appears next to the network interface entry, the LAN driver has marked the interface "down." You may need to execute *ifconfig(1M)* to bring the interface up. If *ifconfig(1M)* fails to bring the card up, refer to Chapter 5, Flowchart 1, to troubleshoot the interface.

Reporting Sockets, Active Connections, Servers, PCBs (Example 2)

```
netstat -An

Active connections
PCB      Proto  Recv-Q  Send-Q  Local Address        Foreign Address    (state)
1a66194  tcp    0       0       192.6.250.100.1023   192.6.250.101.513  ESTABLISHED

netstat-a

Active connections (including servers)
Proto   Recv-Q   Send-Q   Local Address    Foreign Address    (state)
tcp     0        0        hpindma.1023     hpindmb.login      ESTABLISHED
tcp     0        0        *.smtp           *.*                LISTEN
tcp     0        0        *.telnet         *.*                LISTEN
tcp     0        0        *.shell          *.*                LISTEN
tcp     0        0        *.login          *.*                LISTEN
tcp     0        0        *.exec           *.*                LISTEN
tcp     0        0        *.ftp            *.*                LISTEN
tcp     0        0        *.nft            *.*                LISTEN
tcp     0        0        *.rlb(1M)        *.*                LISTEN
udp     0        0        *.who            *.*
pxp     0        0        *.sockreg        *.*
pxp     0        0        *.1024           *.*

netstat -an

Active connections (including servers)
Proto   Recv-Q   Send-Q   Local Address        Foreign Address    (state)
tcp     0        0        192.6.250.100.1023   192.6.250.101.513  ESTABLISHED
tcp     0        0        *.25                 *.*                LISTEN
tcp     0        0        *.23                 *.*                LISTEN
tcp     0        0        *.514                *.*                LISTEN
tcp     0        0        *.513                *.*                LISTEN
tcp     0        0        *.512                *.*                LISTEN
tcp     0        0        *.21                 *.*                LISTEN
tcp     0        0        *.1536               *.*                LISTEN
tcp     0        0        *.1260               *.*                LISTEN
udp     0        0        *.513                *.*
pxp     0        0        *.1541               *.*
pxp     0        0        *.1024               *.*
```

Following is a description of each field:

Field Name	Description
PCB	The address of any Protocol Control Blocks. Displayed only when the *-A* option is specified, displayed in hexadecimal when *-A* is used with the *-n* option.
Proto	The Transport layer (OSI Layer 4) protocol used for the connection. Protocol possibilities are: Transmission Control Protocol (TCP), Packet Exchange Protocol (PXP) and User Datagram Protocol (UDP). For a brief description of protocols, see Chapter 1.
Recv-Q	The current length in bytes of the input queue. This is data that has been received from the network but not yet read by the user process.
Send-Q	The current length in bytes of the output queue. This is the buffered data from the user process which is ready to be sent out over the network.
Local Address	The host name/port address pair, separated by a period, that indicates the address at the local end of the connection. The host name for the local address is that which appears in the */etc/hosts* data base for the local host. Executing *netstat(1)* with the *-n* option causes the internet address to be displayed rather than the symbolic host name. The port address is shown in numeric form if no mnemonic is found in */etc/services* or if the *-n* option is specified. An asterisk (*) in either the host name field or the port address field indicates a wild-card value for sockets that are waiting to accept a connection. *netstat -a* should always list sockets in the *LISTEN* state for the NS service NFT, and for the ARPA

Services *telnet(1)*, *remsh(1)*, *rlogin(1)*, *rexec(1)*, *rwho(1)*, and *ftp(1)* if ARPA Services/9000 is installed. *netstat(1) -a* should also always list a socket in the *LISTEN* state for the *rlb(1M)* diagnostic. The "Local Address" field lists the servers in the form "**.nft*," and "**.rlb(1M)*."

Foreign Address

The host name/port address pair, separated by a period, that indicates the socket address at the remote end of the connection. The host name for the remote address is that which appears in the */etc/hosts* data base for the local host. Executing *netstat(1)* with the *-n* option causes the internet address to be displayed rather than the symbolic host name.

The port address is shown in numeric form if no mnemonic is found in */etc/services* or if the *-n* option is specified. An asterisk (*) in either the host name field or the port address field indicates an unspecified value.

(state)

The current state of the connection. However, only those connections using TCP will have state information. The possible TCP states are:

CLOSED	Socket does not exist. Returned during attempt to establish a connection.
LISTEN	Socket is listening for requests. Returned during attempt to establish a connection.
SYN_SENT	Establishing a connection.
SYN_RECVD	Establishing a connection.

ESTABLISHED	Connection exists. Data can be sent to and/or read from a socket. Returned from an active socket.
FIN_WAIT1	Cannot send any more data. Can still receive data. Returned during the graceful close of a connection.
CLOSE_WAIT	Can still send data. Will only receive queued data. Returned during the graceful close of a connection.
FIN_WAIT2	Cannot send or receive data. Getting ready to close the connection. Returned during the graceful close of a connection.
CLOSING	Cannot send or receive data. Getting ready to close the connection. Returned during the graceful close of a connection.
LAST_ACK	Cannot send or receive data. Getting ready to close the connection. Returned during the graceful close of a connection.
TIME_WAIT	Idle period just before closing a connection. This idle time guarantees that all information and signals have been received. Returned during the graceful close of a connection.

CLOSED	Connection no longer exists. Returned after the graceful close of a connection.

Reporting Routing Information (Example 3)

```
netstat -r

Routing tables
Destination     Gateway         Flags    Refcnt Use        Interface
hp-cupertino    hpindla         UG       0      163        lan0
loopback-net    loopback        U        0      0          lo0
93.0.0          hpindlm         UG       0      0          lan0
95.0.0          hpindma         U        0      2563       lan0
hpindlo         hpindda         UGH      0      163        lan1
```

Using the *-rn* option causes the numeric representation of addresses to be shown as displayed below:

```
netstat -rn

Routing tables
Destination     Gateway         Flags    Refcnt Use        Interface
98.0.0          95.0.0.18       UG       0      168        lan0
127.0.0         127.0.0.1       U        0      0          lo0
93.0.0          95.1.51.89      UG       0      0          lan0
95.0.0          95.1.51.91      U        0      2563       lan0
```

Using both the *-rs* option causes routing statistics to be shown:

```
netstat -rs

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

Following is a description of each field for the *-rn* and *-rs* options:

Field Name	Description
Destination	The host or network that can be reached through the corresponding gateway. If the *-n* option is specified, the destination address is given in numeric form. The symbolic address is entered in either the */etc/networks* or */etc/hosts* data base of the local node, depending upon whether or not the destination is a network or a host, respectively. If *default* is listed, the corresponding gateway entry will be used if no other route is found.
Gateway	The internet address of the node which serves as a gateway to the destination network or node. If the *-n* option is specified, the gateway address is given in numeric form. The symbolic address is retrieved from the */etc/hosts* data base of the local host node.
Flags	*H* signifies that the destination is a host, not a network. *G* indicates the *Gateway* entry is a gateway. *U* indicates that the *Gateway* is up and running. See the *routing(7)* entry in the *LAN/X.25 Reference Pages* for details.
Refcnt	Retained for 4.3 BSD software compatibility.
Use	Retained for 4.3 BSD software compatibility.
Interface	The device file name for the LAN interface, also called the network interface.

Note See the *route(1M)* and *routing(7)* entries in the *LAN/X.25 Reference Pages* for details.

Reporting Memory Statistics (Example 4)

```
netstat -m

116/272 mbufs in use:
13 mbufs allocated to data
1 mbufs allocated to packet headers
13 mbufs allocated to socket structures
25 mbufs allocated to protocol control blocks
6 mbufs allocated to routing table entries
58 mbufs allocated to memory account
0/203 mapped pages in use
440 Kbytes allocated to network (3% in use)
0 requests to allocate memory denied (no memory)
0 requests to allocate memory denied (no credits)
0 requests to reserve memory denied
```

Reporting Protocol Statistics (Example 5)

```
netstat -s

tcp:

0- tcp packets dropped due to bad checksum
0- tcp packets dropped due to incomplete header
0- tcp packets dropped due to bad header offset
0- tcp retransmissions

udp:
0- udp packets dropped due to bad checksum
0- udp packets dropped due to incomplete header
0- udp packets dropped due to bad data length

ip:
0- ip packets dropped due to bad checksum
0- ip packets dropped due to
actual length of data  indicated in IP header
0- ip packets dropped due to any of the following...
  - size of input datagram  min size of an IP header
  - IP version of packet did not match version in use
  - header length in IP header is too small
0- ip packets dropped because of inconsistent
header and packet lengths in IP header

icmp:
0- calls to icmp_error
0- message responses generated
0- icmp packets dropped due to bad code field
0- icmp packets dropped due to message received
minimum length allowed
0- icmp packets dropped due to bad checksum
0- icmp packets dropped due to bad length
0- ip packets of 8 bytes or less with errors
(no icmp error message generated)
0- icmp packets with errors
(no recursive icmp error message generated)

pxp:
0- pxp packets dropped due to bad checksum
0- pxp packets dropped due to bad header
0- pxp packets dropped due to bad length
```

Note Output histogram and input histogram appear only if there are non-zero values to report.

Protocol statistics accumulate since system power up and cannot be reset.

The *netstat -s* statistics show how well the protocols are handling errors in the network. Information varies depending on the protocol. Interpreting these statistics requires a keen understanding of the protocols. In general, watch for non-zero values. The *ping(1M)* diagnostic uses *ICMP* packets. The *Input histogram echo:* and *Output histogram echo reply:* fields should match.

Monitoring Packet Traffic (Example 6)

```
netstat 5

input         output  (lan0)          input         output  (Total)
packets errs  packets errs  colls     packets errs  packets errs  colls
9591    0     3841    0     0         10151   0     3842    1
2       0     2       0     0         2       0     2       0     0
0       0     0       0     0         2       0     0       0     0
0       0     0       0     0         2       0     0       0     0
0       0     0       0     0         2       0     0       0     0
3       0     0       0     0         3       0     0       0     0
0       0     0       0     0         2       0     0       0     0
0       0     0       0     0         2       0     0       0     0
0       0     0       0     0         2       0     0       0     0
```

Sending the interrupt signal, usually by pressing the **[Break]** key, terminates the output.

The first line of numeric values in the report above shows the cumulative interface statistics since the system was last powered up or the statistics were reset with landiag. Each *interval* seconds, a new line displays the number of packets that were received or sent, and any errors or collisions that occurred in that interval of time since the previous line was printed. In this example, the statistics were printed every 5 seconds.

Listing Socket Name Registry (Example 7)

```
netstat -R

Socket Name   Socket Type   Proto   (state)    Local Address
MICKEY        STREAM        TCP     LISTEN     *.1087
```

Following is a description of each field:

Field Name	Description
Socket Name	The socket name assigned by NetIPC. Refer to the *NetIPC Programmer's Guide* for information on socket naming.
Socket Type	The type of socket. Socket types include *STREAM*, *REQUEST*, and *REPLY*.
Proto	The Transport layer (OSI Layer 4) protocol used for the socket. *TCP* is the only protocol to appear here. For a brief description of protocols see Chapter 1.
(state)	The current state of the connection. Only those connections using TCP will have state information. See Example 2 in this chapter for descriptions of TCP states.
Local Address	The host name/port address pair, separated by a period, that indicates the address at the local end of the connection. An asterisk (*) will always appear in the host name field when you specify the *-R* flag, because no value is specified by NetIPC for the host name. The port address (in this case a TCP protocol address) appears to the right of the period.

Note *netstat -R* returns only NetIPC information, not BSD IPC ("Berkeley Sockets") information.

---

ping(1M)

The *ping(1M)* diagnostic sends Internet Control Message Protocol (ICMP) echo packets to a remote host.

Syntax

```
/etc/ping host_addr [packet_size ] [-n number_of_packets]
```

Parameters

host_addr

The IP address of the node that will be echoing packets. A host name from */etc/hosts* may be used in place of the IP address.

packet_size

If specified, sets the size of the ICMP packet in bytes. If *packet_size* is smaller than 16, no round-trip times are displayed.

Take care when specifying a packet size larger than 64 bytes. Some remote systems may have difficulty responding to large packets, the remote system may crash.

Range: 8 to 2048 bytes.

Default: 64 bytes.

number_of_packets

Number of packets *ping(1M)* will transmit before terminating.

Range: 1 to (231 -1) decimal.

Default: If -n is NOT specified *ping(1M)* will send packets until it is interrupted by the *SIGINT* signal (usually sent by the **[Break]** key).

Description

ping(1M) verifies the physical connection to a remote host and reports the round-trip communications time between the local and remote hosts. *ping(1M)* uses the Internet Control Message Protocol (ICMP) echo facility. The remote host must support receiving and responding to ICMP packets. A packet is sent to the remote host every second. As each echo response is received from the remote hosts, the round-trip time is reported.

Although the local host and the remote host must both be capable of ICMP, you do not need to understand this protocol in order to execute *ping(1M)*.

ping(1M) should be initiated:

- To do a preliminary connectivity check when setting up new nodes.
- When difficulties arise in connecting to a particular node or when response from a node seems unusually slow.
- To check the reliability of a route through a gateway.

After *ping(1M)* is initiated, an interrupt signal must be sent to terminate the activity. Use the **[Break]** key to do this. Following this interruption, statistics from the *ping(1M)* session are reported.

If *ping(1M)* is initiated and the remote host does not respond to the outgoing packets, no round-trip information is reported. An error message may or may not be displayed, depending on the nature of the problem. When **[Break]** is pressed, the *ping(1M)* statistics typically indicate a 100% packet loss.

ping(1M) is in the */etc* directory.

ping(1M) sends Internet Control Message Protocol (ICMP) packets to the Network Layer (OSI Layer 3) of a specific node. The network diagnostic program *rlb(1M)* sends packets to the Transport Layer (OSI Layer 4) of a specific node. The network diagnostic program *linkloop(1M)* sends a packet to the Data Link Layer (OSI Layer 2) of a specific node.

Note Only one single *ping(1M)* session may be running at any time on the system. If the initial attempt to run *ping(1M)* fails, try it again after a few minutes.

Following is a typical example of the use of *ping(1M)*. It shows normal *ping(1M)* output when hpindla is the remote host:

```
%ping hpindla 100

PING hpindla:  100 byte packets
100 bytes from x62009303: icmp_seq=1. time=21. ms
100 bytes from x62009303: icmp_seq=2. time=20. ms
100 bytes from x62009303: icmp_seq=3. time=19. ms
100 bytes from x62009303: icmp_seq=4. time=18. ms
100 bytes from x62009303: icmp_seq=5. time=20. ms
100 bytes from x62009303: icmp_seq=6. time=21. ms

  hpindla PING Statistics
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 18/19/21
```

Sending the interrupt signal, usually by pressing the **[Break]** key, terminates the output.

Following is a description of each field:

Field Name	Description
x62009303	The IP address, in hexadecimal, of the remote node *hpindla*.
icmp_seq	The sequence in which the packet was sent from the local node. A missing sequence number indicates that no response was received for that packet from the remote host node.
time	Indicates how long, in milliseconds, it took to receive an echo response from the remote node.

rlb(1M)

The *rlb(1M)* diagnostic exchanges Application Layer (OSI Layer 7) messages with other computers on the network. By using the remote communications commands of *rlb(1M)*, you can:

- Exchange messages with a particular remote computer.
- Poll all nodes known to your local computer.
- Display the time it takes a message to make a round trip.
- Alter the length and number of the messages exchanged.

Syntax

```
rlb[-e]
   [-t]
```

Parameters

`-e` The *echo* option can only be set at command execution time. Normally, input commands and any program output are printed to the display screen. Use this option if you want input commands to be written to your redirected output.

`-t` The *terse* option can be set at command execution time or in the Test Selection Mode. Terse mode means that the command menus throughout the program are not displayed. The default mode is *verbose* which can also be set in Test Selection Mode.

Description

rlb(1M) is an interactive program. It accepts commands from *stdin*, and prints its prompts on the *stderr* file. Aside from prompts and errors, output from any diagnostic command is printed on the *stdout* file. Separation of output and prompts allows you to get a hard copy of the output and still run the program interactively. To get a hard copy, you must redirect *stdout* to a printer.

rlb(1M) can be found in the */usr/bin* directory with other network commands.

rlb(1M) Command Modes

rlb(1M) is a menu-driven program that has two command modes:

- Test Selection Mode.
- Remote Communications Test Mode.

When *rlb(1M)* begins, it is in Test Selection Mode. Test Selection Mode contains six options that let you:

- Use the Remote Node Communications Diagnostic.
- Display or suppress the Test Selection Mode command menu.
- Exit *rlb(1M)*.

You can move from Test Selection Mode to Remote Communications Test Mode by selecting *remote* at the `Enter command:` prompt.

The Remote Communications Test Mode menu contains 11 options that let you:

- Display the Remote Communications Test Mode menu.
- Specify the nodes to which you want to send messages.
- Control message exchange between machines.
- Display the elapsed time for message exchange.
- Leave the Remote Communications Test Mode or *rlb(1M)*.

Each mode displays a command menu. After you select and execute a command from one of these menus, the program returns to the command menu from which you issued the command. You can then enter another command, return to Test Selection Mode, or exit.

Executing the *end* command from Remote Communications Test Mode returns you to Test Selection Mode. You can exit the diagnostic from either of the two modes by entering *quit* at the `Enter command:` prompt.

Redirection of Output

Output redirection can be specified only at execution time.

Note The following are Bourne shell commands (*/bin/sh*).

If you run *rlb(1M)* interactively, you can get a hard copy of the output, with input commands echoed to the hard copy, by using the command:

```
rlb -e 1> device_file_name
```

device_file_name is the device file (for example, a printer) to which you will send the contents of your session.

If you run *rlb(1M)* non-interactively, *stdin*, *stdout*, and *stderr* must be redirected to other files. You can redirect these files by using the command:

```
rlb -e <diag_in_file  1>  diag_out_file  2>&1
```

where:

diag_in_file	Specifies the input file containing your *rlb(1M)* commands.
diag_out_file	Specifies the output file to which *stdin* and *stdout* are redirected.

Executing rlb(1M)

When you enter the *rlb(1M)* command, the program verifies your execution time options. If the command you enter is valid, the specified parameters are set, and the following wakeup message is displayed:

```
NETWORK ONLINE DIAGNOSTIC,  Version x.x
   Fri   April 25, 1986    04:51:29

Copyright 1986 Hewlett-Packard Company
      All rights are reserved.
```

The Test Selection Mode prompt is displayed immediately following the wakeup message. If you specified the *-t* execution time option, the current mode name and the `Enter command:` prompt are displayed without the menu.

Note If you specify an invalid option when you execute *rlb(1M)*, the following message is displayed:

```
usage: -e = echo commands, -t = terse prompts
```

Entering Commands

The next few paragraphs explain how to enter commands from each of the *rlb(1M)* modes.

Abbreviating Command Names

After a menu is displayed, *rlb(1M)* prompts you with:

```
Enter command:
```

When you choose a command from one of the menus, you can enter the complete command word or abbreviate by entering only the first letter. Command names are case insensitive. After you enter the command name or abbreviation that you want, press **[Return]**.

Entering Multiple Commands

Multiple commands can be entered on one line if they are separated by tabs, blanks or commas. When you enter more than one command on a line, each command that you enter is echoed before it is executed. If *rlb(1M)* matches the characters entered to more than one command, it responds with:

```
Ambiguous command, try again.

        Enter command:
```

If *rlb(1M)* cannot match the characters entered to any command, it responds with:

```
Unrecognized command, try again.

        Enter command:
```

If *rlb(1M)* requests an input value, such as a device file name or a message length, you can keep the current value by pressing **[Return]**.

Note The interrupt signal is often used to terminate diagnostic utilities. The chapter assumes you have set the **[Break]** key as your interrupt character, setting the *stty* flags *brkint* and *-ignbrk*. See the *stty(1)* reference page for details.

You can enter the interrupt signal (usually **[Break]**) to abort the current command, at any *rlb(1M)* prompt. When you press **[Break]**, *rlb(1M)* returns to the current command menu. Any input on the line is ignored.

During communications with a remote computer, the message exchange loop can be interrupted with **[Break]**. The diagnostic returns to the Remote Communications Mode command menu and displays the following message:

```
Communications terminated by operator hitting Break.
```

Terminating rlb(1M)

You can terminate *rlb(1M)* in any three ways:

- Input an End-of-File (EOF) value. When *rlb(1M)* is terminated by EOF, it generates the following message:

  ```
  Diagnostic terminated by EOF on input.
  ```

- Enter the *quit* command. The *quit* command causes *rlb(1M)* to terminate. The following message is displayed:

  ```
  Diagnostic terminated by operator.
  ```

- Use **[CTRL]**-\ to leave *rlb(1M)* if input is coming from a file.

Test Selection Mode

If you execute *rlb(1M)* without the *-t* option, the Test Selection Mode menu and prompt are displayed immediately after the wakeup message.

```
Test Selection Mode.

    for network I/O card diagnostics, execute 'landiag or
      sysdiag (700/800)'
     menu     =  Display this menu
     quit     =  Terminate the Diagnostic
     remote   =  Remote Node Communications Diagnostic
     terse    =  Do not display command menu
     verbose  =  Display command menu

Enter command:
```

Following is a description of each Test Selection Mode command.

Menu Command

Displays the Test Selection Mode menu. This is useful if you prefer to use the terse option and do not need to reference the menu frequently.

Quit Command

Terminates the *rlb(1M)* program. Before the program terminates, it displays:

```
Diagnostic terminated by operator.
```

Remote Command

Causes *rlb(1M)* to enter the Remote Communications Test Mode. The Remote Communications Test Mode is described later in this chapter.

Terse Command

Sets the *terse/verbose* flag to terse mode. The amount of output the diagnostic produces is reduced by *terse*. This is helpful when a permanent record of the diagnostic session is kept or when the diagnostic is being used by someone

who is familiar with the commands. The setting of this flag affects the output of both command modes.

Verbose Command

Sets the *terse/verbose* flag to verbose mode. *verbose* mode causes *rlb(1M)* to display the appropriate command menu before prompting for a command. The setting of this flag affects the menus of both command modes.

Remote Communications Mode

The Remote Communications Test Mode is reached from the Test Selection Mode. Selecting the *end* command from this mode returns to the Test Selection Mode.

The Remote Communications Test Mode commands menu is displayed when you enter this mode unless you have previously specified the *terse* option. The display includes current default values.

```
Remote Communications mode.
   Message length = 100, Number of messages to exchange = 1,
   Timeout = 10 seconds, Display round trip time = off

     name     = Name the node file
     all      = Talk to all nodes specified in node file
     continue = Continue exchange if transmit/receive data
                differ
     display  = Display message round trip times
     end      = End remote mode, return to Test Selection
     length   = Set length of transmit messages
     menu     = Display this menu
     number   = Set number of messages to exchange
     quit     = Terminate the Diagnostic, return to shell
     single   = Talk to a specific remote node
     timeout  = Set no response timeout

Enter command:
```

Name Command

The *name* command informs *rlb(1M)* of the name of your node name file. This node name file is used by the *all* command. The *all* command references the file to determine which nodes to exchange messages with.

When *name* begins execution, *rlb(1M)* prompts you for the name of the file with the following message:

```
Enter node file name. Currently /etc/diagnodes:
```

You have several options. You can:

- Send the interrupt signal (usually by pressing **[Break]**), aborting the operation. In this case, the node name file is not changed.
- Press **[Return]**, retaining the previous node name file.
- Enter a complete path name for a file. *rlb(1M)* replaces the old file name with the new one. *rlb(1M)* checks the file's validity. If the file exists and can be opened, then it becomes the new node file, and *rlb(1M)* replaces the old file name with the new one. Otherwise, *rlb(1M)* displays an error message.

All Command

The *all* command causes *rlb(1M)* to exchange messages with all of the nodes on the local network which are listed in the node name file. (See the *name* command for more information on the node name file.) The results and any error messages are written to *stdout*. The results of this command are the same as if you had entered multiple single commands.

The *all* command gets the node names from the current node name file. Manually creating the node name file allows you the option of executing an *all* on a subset of the known nodes. (See the *name* command.)

rlb(1M) looks for the node name file before it tries to exchange messages with the remote nodes. If *rlb(1M)* can't find or open the node name file, an error message is displayed.

rlb(1M) goes through the nodes file, exchanging messages with each individual node. Before each exchange begins, the remote node name is displayed. The local computer exchanges messages with the remote nodes in the order that the nodes are listed in the node name file. When the exchange is complete, the following data are reported:

- If the round-trip time display (see the Remote Communications Test Mode *display* command) is turned on, the round-trip times for the message exchange are checked. If the times differ from the last displayed times by more than the trigger value, the new times are printed.
- The success or failure of the exchange is displayed.
- When the list of nodes is exhausted, *rlb(1M)* displays a summary of the successful responses.

Following is an example of the *all* command message exchange output, without the round-trip time displayed.

```
        NODES COMMUNICATION
    Fri  Dec 3,  1985   04:51:29
Local node talking to node:  my_node
Exchanged 1 messages with node:  my_node

Local node talking to node:  system2
Connection response error.
IPC result code 40 : NSR_NO_NODE

Local node talking to node:  system3
Exchanged 1 messages with node: system3

All nodes command normal completion.
2 out of 3 nodes responded correctly.
```

Once the message exchanges have begun, *all* stops if:

- An error occurs when reading the node name file.
- You send the interrupt signal (usually by pressing **[Break]**).
- The message exchanges for all nodes on the list are complete.

Upon completion of the message exchanges and summary outputs, *rlb(1M)* returns to the Remote Communications menu and prompts you for the next command. See the *single* command and *Message Exchange Sequence* sections of this section for details about the message exchange itself.

Continue Command

The *continue* command allows you to specify whether the message exchange should continue when the transmit/receive data differ. The default setting is no. When set to no, the exchange terminates if the transmit and receive messages are not identical. When set to yes, the exchange continues even though the messages differ. If the data differ, an error message is displayed regardless of the *continue* flag setting.

The continue command prompts you with:

```
Continue if transmit/receive data differ? Currently xxx:
```

where LAN card *xxx* is yes or no. Your options are to:

- Send the interrupt signal (usually by pressing **[Break]**) to abort the operation.
- Press **[Break]** to leave the flag the same.
- Enter *y* or *n*.

After setting the flag value, *rlb(1M)* returns to the Remote Communications Test Mode menu.

Display Command

The *display* command allows you to set the on/off flag for the calculation and display of round-trip message times. If you turn the flag on, you can also specify a trigger value used in comparing consecutive round-trip message times. The comparison between round-trip message times and the trigger value determines whether the current round-trip time is printed. Turning the flag on causes the diagnostic to compute and display the time it takes messages to complete the round trip between the local and remote nodes. The default flag setting is off and the default trigger value is 100 milliseconds.

When *display* begins execution, the following message prompts you for the flag setting:

```
Display round trip times? Currently xxx:
```

where *xxx* is the current value, yes or no.

You have several options. You can:

- Send the interrupt signal (usually by pressing **[Break]**), aborting the operation, without changing anything.
- Press **[Return]**, retaining the previous value.
- Enter *y* or *n* to change the flag value. If you enter *n*, *rlb(1M)* returns to the Remote Communications Test Mode menu after setting the flag to off. If you enter *y*, the diagnostic prompts you for the trigger value.

```
Enter display trigger in milliseconds.  Currently 100:
Hit RETURN to keep it, or enter a new value:
```

You can send the interrupt signal (usually by pressing **[Break]**), press **[Return]**, or enter a new value. The acceptable range for the trigger values is:

```
10 <= trigger_value >= 10000 milliseconds
```

If the value you enter is not acceptable, *rlb(1M)* prompts you again for an acceptable value.

Once you have entered an acceptable value, the trigger is set and *rlb(1M)* returns to the Remote Communications Test Mode menu.

If the *display* flag is on during a message exchange, *rlb(1M)* prints a header message, and the time required for the first message exchange. Three separate round-trip times are displayed, along with the number of messages exchanged at that point in time. The three times are the minimum, mean, and maximum round-trip times. The time values have a resolution of 1/100 second (10 milliseconds). They are displayed in seconds, with precision to three decimal digits.

Once the first message exchange times have been output, *rlb(1M)* displays the times only if the absolute value of the difference between the last two round-trip times is greater than the trigger value. The final time values are displayed when all message exchanges with that node are complete.

Following is an example of the round-trip time display.

```
Enter remote node name: system2

                    NODES COMMUNICATION
                Fri, Feb 1, 1985      14:39:20

MESSAGE ROUND TRIP TIMES IN SECONDS.  MESSAGE LENGTH = 100
BYTES

    MINIMUM    MEAN       MAXIMUM        # MESSAGES
    0.166      0.166      0.166                   1
    0.065      0.085      0.166                   4
    0.065      0.072      0.166                  10
Exchanged 10 messages with node: system2.
```

When the message exchanges are complete, *rlb(1M)* returns to the Remote Communications Test Mode menu.

End Command

The *end* command causes *rlb(1M)* to return to the Test Selection Mode. Before returning, it displays:

```
End of Remote Communications mode.
```

Length Command

The *length* command allows you to alter the length of the messages that *rlb(1M)* exchanges with remote nodes. Changing the length of the messages may have an affect on the round-trip time. The length is specified in bytes and the default value is 100 bytes. The message format is described in the "Test Message Format" section of this chapter.

The *length* command prompts you for the message length with:

```
Enter message length. Currently 100:
Hit RETURN to keep it or enter a new value:
```

Your options are to:

- Send the interrupt signal (usually by pressing **[Break]**) to abort the operation.
- Press **[Return]** to retain the current length.
- Enter an unsigned decimal number between 10 and 1450. If the value is acceptable, the new message length is set, otherwise *length* prompts you again for an acceptable value.

After setting the new value, *rlb(1M)* returns to the Remote Communications Test Mode menu.

Menu Command

The *menu* command displays the Remote Communications Test Mode menu. This is useful if you prefer to use the *terse* option and do not need to reference the menu frequently.

A node name file must be an ASCII text file, created by any HP-UX editor. This file holds a list of network node names. The default name and location of this file is */etc/diagnodes*. The format for the file is:

- One node name per line.
- Each node name is followed with an optional comment (preceded by a blank) and a newline character.
- Each node name consists of the name field, optionally followed by a domain field and organization field. These fields are separated with periods. Empty node names are not allowed. If domain and organization fields are not specified, they default to the domain and organization fields of the local node. See the *nodename* command description in the *LAN/X.25 Reference Pages* for more information.
- Each field of a node name can be up to 16 alphanumeric, case insensitive characters (including hyphens (-) and underscores (_)), beginning with an alphabetic character.

After the file is accepted, *rlb(1M)* returns to the Remote Communications Test Mode menu.

Number Command

The *number* command allows you to alter the number of messages that the diagnostic exchanges with each remote node. The default value is one message.

The number command allows any user to set the "number of messages to exchange" to any value up to 10. To set the number to a value greater than 10, you must be the super-user.

When the *number* command is entered, *rlb(1M)* prompts you with the message:

```
Enter number of messages to exchange. Currently 1:
Hit RETURN to keep it, or enter a new value:
```

Note Using the *number* command can negatively impact other network activity.

You have several options. You can:

- Send the interrupt signal (usually by pressing **[Break]**), aborting the operation without making any changes.
- Press **[Return]**, retaining the previous value.
- Enter an unsigned decimal integer (it must be less than 2**31 - 1). *rlb(1M)* checks the value of the number you enter. If it is less than or equal to 10, the new value is accepted and set by *rlb(1M)*. If the number is greater than 10, the new value is accepted only if you are the super-user. If you are not the super-user, then *rlb(1M)* substitutes 10 for the value you entered. *rlb(1M)* informs you of the substitution:

  ```
  Maximum messages you are authorized to exchange is 10.
  That value has been substituted.
  ```

After the new message exchange number is set, the program returns to the Remote Communications Test Mode menu.

Quit Command

The *quit* command causes *rlb(1M)* to terminate execution and return to the HP-UX shell. Before returning, it displays:

```
Diagnostic terminated by operator.
```

When you use this command, the program returns to the shell with a normal (0) exit status.

Single Command

The *single* command allows you to exchange messages with a single remote node. *rlb(1M)* writes the results and any error messages to *stdout*. The *single* command prompts you for the name of the node:

```
Enter remote node name:
```

You have several options. You can:

- Send the interrupt signal (usually by pressing **[Break]**), aborting the command without changing anything.
- Press **[Return]**, causing *rlb(1M)* to perform a local bounce back operation from NetIPC to the loopback (*lo*) interface (this bounce back does *not* test the network hardware).
- Enter a node name. The name you enter is checked to ensure it is a valid node name. If it is, the exchange begins. The message sequence follows the same steps as one iteration of the *all* command. (See the *all* command.)

You can terminate the exchange at any time by sending the interrupt signal (usually by pressing **[Break]**). This aborts the exchange and generates the message:

```
Communications terminated by operator hitting BREAK.
```

After completion of the message exchange, a status message is displayed.

```
          NODES COMMUNICATION
      Fri  Dec 3,  1985   04:51:29

Exchanged 10 messages with node: system1.
```

rlb(1M) then returns to the Remote Communications Test Mode menu.

Timeout Command

The *timeout* command allows you to alter the length of time that the diagnostic waits for a response from a remote node. The default value is 10 seconds.

The *timeout* command displays:

```
Enter no response timeout in seconds. Currently 10:
Hit RETURN to keep it, or enter a new value:
```

Your options are to:

- Send the interrupt signal (usually by pressing **[Break]**) to abort the operation.
- Press **[Return]** to retain the current value.
- Enter an unsigned decimal integer between 1 and 600. If the value entered is acceptable, the timeout value is set to the new number. Otherwise, *timeout* prompts you again for an acceptable value.

After the new timeout value is set, *rlb(1M)* returns to the Remote Communications Test Mode menu.

During a message exchange, if a remote node does not respond within the no-response timeout time, *rlb(1M)* generates an error message. Communications with that remote node are stopped when a timeout occurs. The diagnostic returns to the Remote Communications Test Mode menu if the command was *single*, or proceeds to the next node if the command was *all*.

The no-response timeout is not effective until a connection has been established with the remote node. If the remote node is NOT ACTIVE on the network, the timeout is not effective until and unless the node does become active. When the remote node is not active, the diagnostic is blocked until the Transport level declares a Network Timeout, or until you send the interrupt signal (usually by pressing **[Break]**). Network Timeout on the Series 600/800 computers is dynamically determined by the system.

Remote Message Exchange Sequence

The Remote Node Message Exchange Sequence is the sequence of steps *rlb(1M)* executes when it exchanges messages with a remote node. This sequence is used by *all* and *single*. The values of *number*, *length*, *timeout*, *continue*, and *display* determine the activities and the results of the message exchanges. The format of the messages is described in the "Test Message Format" section of this chapter.

Message Round Trip

The message round trip starts at the local node with an *all* or a *single* command. *rlb(1M)* gives a message to the NetIPC (OSI Layer 5) software. The message travels down through the network layers until it reaches the driver, where it is sent out on the network. The remote node receives the message from the network and passes it up the network layers to the Remote Loopback Protocol via the NetIPC software. The Remote Loopback Protocol takes the received message and sends it back down the layers to be returned to the initiating node. The local node receives the message from the network and passes it up the layers to the NetIPC software, which then gives the message to the diagnostic.

Message Round Trip Time

The message round-trip time is computed using the HP-UX system clock. The clock has a resolution of 1/60 second or 16.7 milliseconds. The round-trip time includes:

- The time it takes the local node to send the message.
- The time it takes the remote node to receive and return the message.
- The time it takes the local node to receive the message and return it to the diagnostic.

The clock is read just before giving the message to the NetIPC software and just after receiving the response back from the NetIPC software. The message round-trip time is the difference between the two times.

Errors and Interrupts

If errors occur while trying to send or receive messages, an error message is displayed. All errors that occur during the exchange sequence cause the exchange to terminate, unless the transmit/receive data differ and the *continue* flag is on. In this case, the exchange continues after the error message is displayed. (See the *continue* command.)

Sending the interrupt signal (usually **[Break]**) at any time during the message exchange sequence terminates the exchange.

Message Exchange Sequence

A message exchange begins when the local node calls the NetIPC software to build a connection with a remote node.

If the NetIPC software cannot set up a connection, *rlb(1M)* displays an error message. Otherwise, *rlb(1M)* sets up the no-response timeout.

Common reasons for connection response error messages are:

- The remote node is not on the network, or is powered off.
- The remote node has had a failure.
- The local computer is unable to access the network.

Once the node is found, a connection is established. Next the diagnostic checks to see if the round-trip time *display* flag is set. If it is, the header for the time display is output and the connection is ready to use.

To begin the message exchange with the remote node, the Session Layer software sends the message packet and waits for a response.

After receiving the response message, *rlb(1M)* calculates the round-trip time if *display* is enabled. (If *display* is enabled, *rlb(1M)* would have read the HP-UX clock when it sent the message and again when *rlb(1M)* received it. Also, the first round-trip times would have been displayed.) If the absolute value of the difference between the last two calculated times is greater than the trigger value, the new times are displayed.

Next, the diagnostic compares the data sent to the data received. If they differ in length, an example of the error message displayed is:

```
Transmit/Receive message lengths differ
Transmit length = 100, Receive length = 98.
```

If the data differ in content, the error message displayed is:

```
Transmit/Receive data differ.
```

If "continue exchange on differing data" (*continue*) is enabled, the exchange continues after printing the message. If *continue* is disabled, the connection is cleaned up and the exchange sequence terminates.

After each round trip, the diagnostic looks to see if it has exchanged the correct number of messages (set with the *number* command). If not, it repeats the sequence. If so, it exits the exchange loop, and cleans up its connections.

Once the exchange loop is complete, and if *display* is enabled, *rlb(1M)* displays the final round-trip times. The number of successfully exchanged messages is always displayed. Successful response means that the message has been received within the timeout and the sent/received data are identical. If any messages have not been successfully exchanged, the following message is displayed:

```
INCOMPLETE EXCHANGE with node: node_namme.
ONLY xx of yy messages were exchanged.
```

The number of messages with different send/receive data is also displayed.

After displaying the completion message, *rlb(1M)* terminates the remote connection. If any error occurs during termination of the connection, an appropriate message is generated.

Test Message Format

The message that is exchanged with remote nodes in the *all* and *single* commands has a fixed format. It consists of a header and data.

Message Headers

The header has a fixed value. Included in the header is the operator's real user ID. The format of the message header is:

```
UID=109, LAN REMOTE LOOPBACK PACKET:
```

Message Data

The data is a list of the displayable characters from the ASCII space character (0x20) to the ASCII tilde character (0x7E), in ascending order. The list of ASCII characters is repeated, if necessary, to achieve the desired message length.

The maximum message length is 1450 bytes.

Security

The file */usr/bin/rlb* must be owned by root and must have the "set user ID on execution" mode bit set and have 4555 (*-r-sr-xr-x*) permission.

Also, permission bits on the node name file are checked before the file is read. (See *chmod(1)* in the *HP-UX Reference Manual.*)

rlb(1M) error messages appear in Appendix B of the this manual. Appendix B lists each error message, an explanation, and a possible fix.

landiag(1M)

The *landiag(1M)* command allows you to diagnose and correct problems with the LANIC. Using *landiag(1M)*, you can:

- Reset the card.
- Check the self-test code for a failed interface test.
- Check the driver statistics for unusual and unexpected values.

Note You must have super-user capabilities to use the *clear* and *reset* functions of *landiag(1M)*.

Syntax

```
landiag [-e]
        [-t]
```

Parameters

`-e` The *echo* option can only be set at command execution time. Normally, input commands and any program output are printed to the display screen. Use this option if you want input commands to be written to your redirected output.

`-t` The *terse* option can be set at command execution time or in the Test Selection Mode. Terse mode means that the command menus throughout the program are not displayed. The default mode is *verbose* which can also be set in Test Selection Mode.

Description

landiag(1M) is an interactive program. It accepts commands from *stdin*, and prints its prompts on the *stderr* file. Aside from prompts and errors, output from any diagnostic command is printed on the *stdout* file. Separation of output and prompts allows you to get a hard copy of the output and still run the program interactively. To get a hard copy, you must redirect *stdout* to a printer.

landiag(1M) can be found in the */usr/bin* directory with other network commands.

landiag(1M) Command Modes

landiag(1M) is a menu-driven program that has two command modes:

- Test Selection Mode.
- LAN Interface Test Mode.

When *landiag(1M)* begins, it is in Test Selection Mode. Test Selection Mode contains five options that let you:

- Use the LAN Interface Diagnostic.
- Display or suppress command menus.
- Exit *landiag(1M)*.

You can move from Test Selection Mode to LAN Interface Test Mode by selecting *lan* at the `Enter command:` prompt.

The LAN Interface Test Mode menu contains seven options that let you:

- Display the LAN Interface Test Mode menu.
- Clear statistics registers.
- Display LAN Interface status and statistics registers.
- Select the LAN interface to test.
- Reset the LAN interface.
- Leave the LAN Interface Test Mode or *landiag*.

After you select and execute a command from the menu, the program returns to the command menu from which you issued the command. You can then enter another command, return to the Test Selection Mode, or exit.

Executing the end command from Remote Communications Test Mode returns you to Test Selection Mode. You can exit the diagnostic from either of the two modes by entering quit at the `Enter command:` prompt.

Test Selection Mode

If you execute *landiag* without the *-t* option, the Test Selection Mode menu and prompt are displayed immediately after the wakeup message.

```
Test Selection Mode.

     menu     = Display this menu
     quit     = Terminate the Diagnostic
     lan      = LAN Interface Diagnostic
     terse    = Do not display command menu
     verbose  = Display command menu

Enter command:
```

Note that this menu is the same as for Test Selection Mode of rlb(1M), except that there is a lan command in place of remote. The lan command invokes the LAN Interface Test Mode.

Following is a description of each Test Selection Mode command.

Menu Command

Displays the Test Selection Mode menu. This is useful if you prefer to use the terse option and do not need to reference the menu frequently.

Quit Command

Terminates the *landiag* program. Before the program terminates, it displays:

```
Diagnostic terminated by operator.
```

Lan Command

Causes *landiag(1M)* to enter the LAN Interface Test Mode. The LAN Interface Test Mode is described later in this chapter.

Terse Command

Sets the *terse/verbose* flag to terse mode. The amount of output the diagnostic produces is reduced by *terse*. This is helpful when a permanent record of the diagnostic session is kept or when the diagnostic is being used by someone who is familiar with the commands. The setting of this flag affects the output of both command modes.

Verbose Command

Sets the *terse/verbose* flag to verbose mode. *verbose* mode causes *rlb(1M)* to display the appropriate command menu before prompting for a command. The setting of this flag affects the menus of both command modes.

LAN Interface Test Mode

When you enter LAN Interface Test Mode, the commands menu and prompt are displayed. If you specified the terse option, only the mode name, current device file name and the prompt are displayed.

```
LAN Interface test mode.   LAN Interface device file =
/dev/lan

     clear    = Clear statistics registers
     display  = Display LAN Interface status and statistics
                registers
     end      = End LAN Interface Diagnostic, return to Test
                Selection
     menu     = Display this menu
     name     = Name of the LAN Interface device file
     quit     = Terminate the Diagnostic, return to shell
     reset    = Reset LAN Interface to execute its selftest

Enter command:
```

Clear Command

The *clear* command can be used by a super-user only. If you are not the super-user, the following message is displayed if you try to execute *clear*:

```
Not authorized to clear statistics.
```

Clear sets the frame (or packet) statistics registers on the LAN interface card to zero (0). These registers keep a cumulative count of local frame errors and frame traffic. The LAN Interface Status Display which results from executing the *display* command contains more information on the specific registers.

landiag begins by validating the device file. If it is not a valid LAN interface card, an error message is displayed and *landiag* returns to the LAN Interface Test Mode menu.

After the device file is opened, the status of the LAN interface card is checked. An error message is displayed if the status of the device cannot be obtained.

Once *landiag* is sure that it can clear the statistics, it displays the message:

```
Clearing LAN Interface statistics registers.
```

The registers are set to 0 and the program returns to the LAN Interface Test Mode menu.

Display Command

Executing *display* results in a display of the current status information about the local LAN interface device. The results and any error messages are written to *stdout*. After the available information is displayed, *landiag* returns to the LAN Interface Test Mode menu.

When *display* is invoked, *landiag* checks on the validity of the specified device file:

- *landiag* displays the current device file name and verifies that the file exists.
- *landiag* examines the file to ensure that it is the correct LAN interface device file.
- *landiag* opens the device file.

If any of these checks fail, an error message is displayed, and *landiag* returns to the LAN Interface Test Mode menu.

After *landiag* completes the validity checks on the device file, it begins to display the status information. For a complete description of all status information, refer to Appendix D.

As soon as *landiag* reads the select code on the device file, the display status header for the LAN interface card is printed. The following example is for the Series 300.

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986    08:51:29

Device file                      = /dev/lan
Select code                      = 21
```

Next, *landiag* checks the state of the LAN interface card. The two possible states are:

- FAILED: If the interface card is in the failed state, the diagnostic displays a message similar to that shown below. Refer to Appendix E for a list of the self-test completion code values.

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986    08:51:29

Device file                      = /dev/lan
Select Code                      = 21
Current state                    = FAILED !!!
Selftest Completion Code         = 10
*  *  *  *  LAN INTERFACE SELFTEST FAILURE *  *  *  *
```

- ACTIVE: The interface card is usually in the normal, active state. If it is, the local station address (also called the LAN interface address or the link-level address) and local packet statistics are available and displayed after the current state status.

If any errors occur in reading the station address or statistics registers, *landiag* generates error messages, terminates the *display* command, and returns to the LAN Interface Test Mode menu.

Following is an example of a display for a Series 300/400 interface card in the active state.

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986     08:51:29

Device file                       = /dev/lan
Select code                       = 21
Current state                     = active
LAN Interface address, hex        = 080009-000000
Number of multicast addresses     = 2
Frames received                   = 107983
Frames transmitted                = 113587
Undelivered received frames       = 11
Untransmitted frames              = 7
CRC errors received               = 0
Transmit collisions               = 1528
One transmit collision            = 68
More transmit collisions          = 730
Excess retries                    = 0
Deferred transmissions            = 0
Carrier lost when transmitting    = 0
No heartbeat after transmission   = 0
Frame alignment errors            = 0
Late transmit collisions          = 0
Frames lost                       = 0
Unknown protocol                  = 0
```

After the status display fills the screen, it prompts you with the message:

```
Press enter to continue
```

Press **[Return]** or **[Break]** to look at the remaining statistics.

This example shows the display for a Series 800 interface card in the active state. The `Select Code` field in the Series 300/400 display is replaced with the `lu number` in the Series 800 display. Five additional fields have also been added to the bottom of the display.

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986     08:51:29

Device file                      = /dev/lan0
Lu number                        = 0
Current state                    = active
LAN Interface address, hex       = 080009-000000
Number of multicast addresses    = 2
Frames received                  = 107983
Frames transmitted               = 113587
Undelivered received frames      = 11
Untransmitted frames             = 7
CRC errors received              = 0
Transmit collisions              = 1528
One transmit collision           = 68
More transmit collisions         = 730
Excess retries                   = 0
Deferred transmissions           = 0
Carrier lost when transmitting   = 0
No heartbeat after transmission  = 0
Frame alignment errors           = 0
Late transmit collisions         = 0
Frames lost                      = 0
Unknown protocol                 = 0
Bad control field                = 0
IEEE 802.3 XID packets           = 0
IEEE 802.3 Test packets          = 1
Unable to respond TEST/XID pkts  = 0
Illegal sized frames             = 0
Unable to find transmit buffers  = 0
One or zero receive buffers      = 0
```

After the status display fills the screen, it prompts you with the message:

```
PRESS enter to continue
```

Press **[Return]** or **[Break]** to look at the remaining statistics.

End Command

The *end* command causes *landiag* to return to the Test Selection Mode menu. Before returning, it displays the following message:

```
End of LAN Interface test mode.
```

Menu Command

The *menu* command displays the LAN Interface Test Mode menu. This is useful if you prefer to use the *terse* option but need to reference the menu occasionally.

Name Command

The *name* command allows you to tell *landiag* which LAN interface card to test. If you do not use this command, the default device file is */dev/lan*. *Name* displays the current device file name and prompts you for a new one with the following message:

```
Enter LAN Interface device file name.  Currently /dev/lan:
```

You have several options. You can:

- Press **[Break]** to abort the operation without making any changes.
- Press **[Return]** to retain the current device file.
- Enter a complete path name for a device file. The device represented by the device file name you enter becomes the current device to be tested. For example, enter */dev/lan1* to test the second LAN interface card if your system contains more than one LAN interface.

  First the device file is checked for validity by *landiag*. If the file does not exist, is not a LAN interface device file, or cannot be opened, an error message is displayed and *landiag* prompts you for the device file name again. If the file is valid, *landiag* accepts it, and then returns to the LAN Interface Test Mode menu and prompts for the next command.

Quit Command

The *quit* command causes *landiag* to terminate execution of *landiag*. Before returning, it displays the following message:

```
Diagnostic terminated by operator.
```

When you use this command, the program terminates with a normal (0) exit status.

Reset Command

Note Use *reset* with discretion, since it can disrupt the network. It is possible to lose data or abort connections when performing a reset.

The *reset* command can be used by a super-user only. If you are not the super-user, the following message is displayed if you try to execute *reset*.

```
Not authorized to reset LAN Interface.
```

Reset causes the LAN interface card to be reset and to execute its self-test.

landiag begins by validating the device file. If the current device file is not a valid LAN interface card, an error message is displayed and *landiag* returns to the LAN Interface Test Mode menu. Once *landiag* is sure that it can reset the LAN interface card, the following message is displayed:

```
Resetting LAN interface to run selftest.
```

If *landiag* encounters an error while sending reset instructions to the device, it displays an error message.

landiag is blocked during the time that it takes the interface card to reset and complete its self-test. The following connections are affected by the reset:

If you are running any of the NS/9000 services (such as Network File Transfer or Remote File Access), packets may be lost. (See the *Using Network Services* manual for more information about NS/9000 services.)

- Data will be delayed on TCP connections.
- TCP connections could be dropped.
- Data could be delayed or lost for UDP sockets.

After successful completion of the self-test, the LAN interface card state is ACTIVE. *landiag* then returns to the LAN Interface Test Mode menu.

linkloop(1M)

Another diagnostic program is *linkloop(1M)*, which allows you to run Link Layer (OSI Layer 2) loopback tests between HP 9000 computers.

Syntax

```
linkloop [-n count] [-f devfile] [-t timeout] [-s size]
[-v] linkaddr
```

Parameters

`-n count`

Sets the number of frames to transmit. If *count* is set to zero, *linkloop(1M)* transfers frames indefinitely until an interrupt signal is received. To interrupt the transfer of frames, hit the **[Break]** key. The default value for *count* is one.

`-f devfile`

Specifies which device file to use. For the Series 300/400, the device file must be an IEEE 802.3 device file, that is, the major number should be 18. In the case of the S700/S800, the encoded protocol bit #31 should be 0. If no device file is entered on the Series 300/400 or Series 600/800, *linkloop(1M)* uses */dev/lan* as the default. If */dev/lan* is not present or is the wrong type of device file, then *linkloop(1M)* uses */dev/ieee* as the default. If no device file is specified on the Series 700, */dev/lan*0 is used as the default. If */dev/lan*0 is not present or is the wrong type of device file, *linkloop(1M)* uses */dev/ieee0* as the default.

`-t timeout`

Sets the amount of time (in seconds) to wait for a reply from the remote computer before aborting the operation. If *timeout* is set to zero, *linkloop(1M)* waits indefinitely for a reply from the remote computer. The default value for *timeout* is 2 seconds.

`-s size`

Sets the size of the frame to send. The default frame size is the same as the maximum frame size, which is 1497.

`-v`

Sets the *verbose* option. In addition to the regular summary of test results, this option causes the display of more extensive error information. The verbose option supplies useful information if a response from a remote computer is received, but the reply is different than expected. For example, if the received frame is different in length or content from the transmitted frame, the verbose option causes the details of the difference to be displayed. All verbose output is preceded by the number of replies accepted before the error occurred.

`linkaddr`

linkloop(1M) tests the connectivity of the local computer and the remote computer specified by the link level (station) address. The link level address of a remote computer can be found on the network map or worksheet, or by executing the *landiag* program on the remote computer. In the "LAN Interface Status" mode of *landiag*, execute the *display* command to list the link address. This link level address is usually represented as a hexadecimal string prefixed with *0x* (but can also be represented as an octal string prefixed with *0* or as a decimal string). The least significant bit of the first byte of the link address must be set to zero. The address must not be a multicast or broadcast address.

Description

linkloop(1M) uses IEEE 802.3 link-level test frames to check connectivity within the LAN. This program is different from the remote loopback capability of *rlb* because it only tests the Link Layer connectivity, and not the Transport Layer connectivity.

To test the local computer's connectivity to a remote computer with the station address 0x008009000222, enter the following command:

```
linkloop 0x008009000222
```

If the test is successful, the following message is displayed:

```
    frames sent                    : 1
    frames received correctly      : 1
    reads that timed out           : 0
Loopback to LAN station: 0x008009000222   OK
```

If an erroneous link address is entered as the *linkloop(1M)* parameter, the following error message is displayed:

```
Loopback to LAN station: 0x0e00090000F3   FAILED
```

linkloop(1M) can be aborted with the interrupt signal. The interrupt signal is entered by hitting the **[Break]** key. If *linkloop(1M)* is aborted, the current results are displayed.

lanscan(1M)

The *lanscan (1M)* diagnostic displays the following information about LAN devices that are properly bound to system I/O services:

- Hardware Path (Series 600/800 and Series 700 only) or Select Code (Series 300/400 only)
- Station Address.
- Device *lu*.
- Hardware State.
- Network Interface Name, Unit, and State.
- Network Management ID.
- Encapsulation Methods.

Syntax

```
/etc/lanscan [system [core]]
```

Parameters

system	Specifies the system file on which the command is to be executed. The default system is */hp-ux*.
core	Specifies the core dump file. If this command is executed on the system that is currently running, the default is */dev/kmem*.

Description

This command displays information about LAN cards that are successfully bound to the system. If a LAN card physically exists in the hardware backplane but has failed to bind to the system at boot-time, no information will be displayed about it.

The first four categories of information are specific to the LAN card and the other three categories are specific to the Network Interface associated with the LAN card. The Select Code (Series 300/400 only) is the value of the dip switch setting on the LAN card. The station address (Series 600/800 and Series 700), also known as the Ethernet address, Physical address or Hardware address, is the unique 12-digit hexadecimal address stored in the NOVRAM chip on the LAN card. The hardware path for the Series 700 indicates the IO module ID, the slot number into which the card is inserted, and the functional ID of the card.

The Device Lu is the device logical unit associated with a LAN card. On the Series 800, the system assigns the Device Lu after system bootup. The system stores logical unit numbers until they are changed or removed with the *insf(1M)* or *rmsf(1M)* command. For example, if a system is booted up with one LAN card on hardware path 4.3, the system will assign *lu 0* to the card. If there is a system shutdown and the LAN card is moved from hardware path 4.3 to 4.5, the system will assign *lu 1* to this LAN card when the system is rebooted. *lu 0*, which was assigned during a previous system bootup, is still reserved for hardware path 4.3 although there is no longer a LAN card on this path. If this system is shutdown again and a second LAN card is installed on hardware path 4.3, the system will assign *lu 0* to the second LAN card when the system is booted up the third time. The first LAN card remains on hardware path 4.5 and its *lu* is still *1*.

On the Series 800 two special device files are created for each Device Lu when the system is booted up. For a LAN card with *lu x*, device files */dev/lanx* and */dev/etherx* are created. You can use these device files to send and receive packets, and to control the device via Link Level Access (LLA). When the Device Lu is displayed as a dash instead of a numerical value, the system core dump occurred before the boot-up process was complete and no Device Lu was assigned.

On the Series 300/400, the Device Lu is determined by the increasing order of the Select Code. For example, if a system has three LAN cards with select codes 21, 23, and 29, the Device Lu numbers for these LAN cards will be 0, 1, and 2 respectively.

On the Series 700, the Logical Unit is identical to the Network Interface Unit. Note that for the Series 700, the network interface unit is assigned in the order in which the LAN cards are detected by the IO subsystem.

The Hardware State is the LAN card state. If the hardware state is up, the card is functioning properly. If the hardware state is down, the card cannot send or receive packets due to a hardware, firmware, or driver state problem.

To correct the problem, check the LAN card hardware, the MAU, and the cable connection. You should also test the device with diagnostic tools. If the MAU is disconnected, reconnect the MAU and reset the LAN card.

The Network Interface Name, Unit, and State fields provide information about the Network Interface associated with the LAN card. A Network Interface associated with a LAN card has the name *lan*.

On the Series 800, the Network Interface Unit is determined by the physical location of the LAN card relative to the other LAN cards in the hardware backplane. The LAN cards located in a lower hardware module will be assigned Interface Unit numbers before cards located in a higher hardware module. If there is more than one LAN card in the same hardware module, the card in the lower slot will be assigned an Interface Unit before the card located in the higher slot.

For example, a system might have a CIO Channel Adapter in hardware module 4 and another in hardware module 8. There are two LAN cards in slots 4 and 5 in the Adapter in module 4 and three LAN cards in slots 3, 9, and 10 in the other Adapter. In this case, the hardware paths of these cards are 4.4, 4.5, 8.3, 8.9, and 8.10 and their Network Interface Unit numbers are 0, 1, 2, 3, and 4 respectively.

In the next example the system has three NIO LAN cards in hardware modules 4, 6, and 8. The hardware paths of these LAN cards are 16, 24, and 32, and the Network Interface Unit numbers are 0, 1, and 2 respectively.

Note The Network Interface Unit of a LAN card does not necessarily match its Device *lu*.

On the Series 300/400, the Network Interface Unit is determined by the relative order of the Select Code. For example, if a system has three LAN cards with Select Codes 21, 23, and 29, then the Network Interface Unit numbers are 0, 1, and 2 respectively.

On the Series 700, the network interface unit is assigned according to the order in which the LAN cards are detected by the IO subsystem. The Lan Interface Controller on the Core IO card is detected first. An interface unit value of 0 is assigned to the Core IO Lan card. EISA cards have interface unit numbers ranging from 1 to 4.

You can configure the Network Interface State with the *ifconfig(1M)* command. If the current Network Interface State is down and the Hardware State is up, then you can use the *ifconfig(1M)* command to bring up the Network Interface. If the hardware state is down, the Network Interface State will be brought down by the system. If the Hardware State changes to up again while the system is running, the Network Interface State will remain down until you use the *ifconfig(1M)* command to bring it up again. When the Hardware State is up, you can use the *ifconfig(1M)* command to change the Network Interface State to either up or down.

The Network Management ID specifies a unique ID assigned by the system for the Network Management of each Network Interface.

The Encapsulation Method specifies the configured encapsulation method for a Network Interface. It can be ETHER (Ethernet encapsulation) only, IEEE (IEEE802.3 encapsulation) only, or both. You can use the *lanconfig(1M)* command to configure and deconfigure the encapsulation method of each Network Interface associated with a LAN card.

Example 1

The following command executes the *lanscan* command on the system */hp-ux.test* file.

```
/etc/lanscan /hp-ux.test
```

Example 2

The following command executes the *lanscan* command on the */tmp/hp-ux.*1 file with */tmp/hp-core.1* as the core dump file.

```
/etc/lanscan /tmp/hp-ux.1 /tmp/hp-core.1
```

Example 3

The following list gives examples of possible hardware paths:

`4.3` Specifies the CIO Channel Adapter in hardware module 4 and the CIO LAN card in slot 3 of this module.

`4.4` Specifies the CIO Channel Adapter in hardware module 4 and the CIO LAN card in slot 4 of this module.

`4.7` Specifies the CIO Channel Adapter in hardware module 4 and the CIO LAN card in slot 7 of this module.

2/4.6 Specifies the bus converter in hardware module 2, the CIO channel adapter in hardware module 4 on this converter, and the CIO LAN card in slot 6 of the CIO module.

32 Specifies the NIO LAN card in hardware module 8.

Example 4

The following list gives examples of select code:

21 Specifies the DIO LAN card on the mother board.

29 Specifies a second DIO LAN card.

Example 5

The following examples give the active LAN Station Address stored in RAM (Random Access Memory) on the LAN card.

```
0x08000903C657
0x08000902094C
```

Example 6

This example shows a display from the *lanscan(1M)* command:

```
Select    Station          Dev  Hardware   Net-Interface     NetMgt  Encapsulation
Code      Address          lu   State      NameUnit State    ID      Methods
29        0x08000903C567   0    UP         lan0     UP       2       ETHER
30        0x08000902094C   1    UP         lan1     DOWN     3

Hardware  Station          Dev  Hardware   Net-Interface     NetMgt  Encapsulation
Path      Address          lu   State      NameUnit State    ID      Methods
4.3       0x0800090171E9   2    UP         lan0     DOWN     1       ETHER
4.4       0x080009021DF4   0    DOWN       lan1     DOWN     3       IEEE802.3

Hardware  Station          Dev  Hardware   Net-Interface     NetMgt  Link
Path      Address          lu   State      NameUnit State    ID      Type
2/4.5     0x0800090190E8   5    UP         lan0     UP       3       ETHER IEEE802.3
```

LANDAD

In addition to the previous network diagnostics, *LANDAD* is available for Series 600/800 and Series 700 computers. *LANDAD* stands for Local Area Network Device Adapter Diagnostic. It is part of the On-Line Diagnostic Subsystem *sysdiag*, supplied with the HP-UX operating system. You can use *LANDAD* to do the following on HP 9000 Series 800 computers:

- Identify the product type and station address of the LAN card.
- Report the status of the LAN card.
- Report the link statistics of the LAN card.
- Reset the LAN card.
- Perform self-test on the LAN card.
- Execute a local or external loopback test.
- Send TEST or XID (exchange identification) packets to a remote node and interpret the results.
- Perform AUI cable and MAU fault tests.

Caution HP recommends that the On-Line Diagnostic Subsystem be used by HP Customer Engineers and trained customers only.

Operation of *LANDAD* is beyond the scope of this manual. For detailed information refer to:

On-Line Diagnostic Subsystem Manual
LAN Link Hardware Troubleshooting Manual

The following *LANDAD* example reads LAN interface card statistics. The user has super-user capability. (*pdev* is the physical device number.)

```
#/usr/diag/bin/sysdiag[Return]

DUI > run landad pdev=8.2 section=7
```

7

Using the Logging and Tracing Facility

This chapter describes the common logging and tracing tool. It contains the following sections:

- Overview of Logging and Tracing.
- Using the *nettl* Logging Facility.
- Using the *nettl* Tracing Facility.
- *nettl(1M)*.
- *netfmt 1M)*.
- Examples of *nettl* and *netfmt* Operation.
- Filter Command Lines.

Overview of Logging and Tracing

The *nettl* command controls logging and tracing for LAN/9000. *nettl* is the common logging and tracing tool for most HP-UX networking products, including X.25/9000, NS/9000 and MAP.

The *nettl* logging and tracing facility uses background daemon processes to receive log and trace data from network subsystems and direct that data to the proper files and, if a disaster log message is encountered, to the system console.

The *nettl -start* command starts the *nettl* daemons. If you execute the *ps* command after starting the logging and tracing facility, these daemons will be displayed as *ntl_reader* and *nktl_daemon*. You will also see the *ntfmt* process displayed. It formats disaster messages that are to be sent to the system console. The *nettl -start* command should be in the */etc/netlinkrc* file before any networking subsystem is started. The *update* command executes this command automatically when you reboot the system and starts the logging facility. You must execute the *nettl -traceon* command after the system is rebooted to start the tracing facility.

When the *nettl* daemons are started, they read the */etc/conf/nettlgen.conf* file. This file contains network configuration information used by the *nettl* daemons to log and trace network activities. This file also identifies the subsystems and the level of detailed information that is to be logged. Entries are added to this file automatically when you install your software.

The default setting of the logging facility of the *nettl* daemon specifies that error and disaster messages from all subsystems are to be logged. The default setting is defined in *nettlgen.conf*.

The *nettl -stop* command stops the *nettl* daemons and terminates logging and tracing for all network subsystems. You should not stop the *nettl* facility unless all network activities have been halted.

Using the nettl Logging Facility

Log messages record unusual or exceptional events such as errors, warnings, and state transitions. Logging is part of standard network operation and is started automatically when the system is booted up.

Starting Logging

The following command should be placed into the */etc/netlinkrc* file to enable logging for all subsystems:

```
nettl -start
```

You can modify the default logging options for a subsystem by placing *nettl* commands in the */etc/netlinkrc* file following the *nettl -start* command. For example:

```
nettl -log warning -entity ns_ls_driver
```

If there is some area of network activity that is of particular concern, such as a subsystem that was recently installed, had its configuration modified, or has been subject to performance degradation, you may elect to start a *netfmt* process that sends all log messages for that particular subsystem to a file or to the system console.

Log Files and Logging Operations

When the logging and tracing facility is started, the nettl daemons open the log files specified in */etc/conf/nettlgen.conf*. By default, the log files are */usr/adm/nettl.LOG00* or */usr/adm/nettl.LOG01*. You can change the default files with the *nettlconf* command. This command is described in the *HP-UX Manual Reference Pages*.

The *nettl* daemons always write to the */usr/adm/nettl.LOG00* file. When that file is full, the daemons copy the file contents to */usr/adm/nettl.LOG01* and purge the contents of */usr/adm/nettl.LOG00*. If */usr/adm/nettl.LOG01* already exists, the contents are overwritten. If that file does not exist, it is created.

This process writes to the *.LOG00* file continously and copies to the *.LOG01* file while the *nettl* daemons are running. This process insures that the oldest log data on the system is always in the *.LOG01* file, and the latest log data is

always in the *.LOG00* file. This technique allows log files to correctly sequence log entries during system shutdowns and reboots. By default, the maximum size for these files is 500 Kbytes.

Each log entry is written in an internal binary format. It contains a header information field and a data field. The header information includes a time stamp, a subsystem ID, a log class, and other miscellaneous fields. The data field contains subsystem specific information describing the log event and subsystem error number. The log class is one of 4 values: Disaster, Error, Warning, or Information.

Disaster Log Class Messages

Disaster log class messages indicate events that may jeopardize system or network integrity. When a disaster log class message occurs, the node should be taken offline and all networking operations aborted or suspended until the problem is corrected.

Error Log Class Messages

Error log class messages indicate events that will not affect overall system or network operation, but will cause application program calls to fail or complete with an error. An error event requires special action on the part of the user or application, such as repeating a transmission request or reestablishing an SVC.

Warning Log Class Messages

Warning log class messages indicate events that may be recoverable by the network. They may result from an incorrectly specified parameter or the misuse of a command. Most subsystems can recover from a warning event without further action on the part of the user or application.

Information Log Class Messages

Informational log class messages describe significant events that cause state changes within the LAN/9000 subsystem. These events do not require any exceptional action on the part of the LAN/9000 subsystem and are part of normal operation. These events include the establishment and termination of SVCs.

You should use the *netfmt* command to view log data. This command formats the data in a readable fashion that is suitable for viewing at a terminal screen or printing. The log files contain log messages for all network subsystems

running on the machine. The *netfmt* command allows you to filter out messages in which you are not interested.

For example, consider the following command

```
netfmt -c lanlog_filters -F -f /usr/adm/nettl.LOG00
```

This command causes *netfmt* to use the format filters specified in *lanlog_filters* in the */usr/adm/nettl.LOG00* file. The -F option causes *netfmt* to remain active and follow the */usr/adm/nettl.LOG00* file as the *nettl* daemons write new records. This process continues even if the */usr/adm/nettl.LOG00* file is written to */usr/adm/nettl.LOG01* and purged of data.

The *netfmt* command only formats information that has been logged by the *nettl* daemons. To change the information being logged, issue the *nettl* command with the *-log* option set as shown in the example below:

```
nettl -log disaster error warning -entity ns_ls_ip
ns_ls_driver
```

This command causes warning messages, in addition to the error and disaster messages, to be logged. The syntax and semantics of the *nettl* and *netfmt* commands are described later in this chapter. Since this tool is part of a larger subsystem used by other HP-UX products as well, only options applicable to LAN/9000 are described here. Refer to the appropriate documentation to use the common logging and tracing tool with other networking products.

Using the nettl Tracing Facility

Tracing is a detailed examination of operations performed by a subsystem. Trace messages record normal operational events including the reception and transmission of data. Unlike logging, which is part of standard network operation, tracing is used only as a debugging and troubleshooting tool and is not part of standard operation of a subsystem.

Starting Tracing

To start tracing, the *nettl* daemons and network logging must be active. This is usually done automatically when the logging and tracing facility is started during system startup by commands in the */etc/netlinkrc* file. When network logging is in progress, the *nettl -traceon* command initiates tracing on a subsystem.

When tracing begins, two additional *nettl* daemons begin executing. If you subsequently issue a *ps* command, you will see four processes: two shown as *ntl_reader* and two shown as *nktl_daemon*. One pair of *nettl* daemons is dedicated to network logging, and one pair is concerned with tracing. There is also a *netfmt* process running to send disaster messages to the system console.

Only one trace of a LAN/9000 card can be active at any given time.

Trace Files and Tracing Operations

The *nettl -traceon* command allows you to specify the files used in the trace, the size of the files, and the maximum length of trace records. When tracing begins, the *nettl* daemons use the same circular file method as used by the logging facility. The pathname that you specify in the command is used with a suffix added and the filenames will have the following format: *filename.TR0* and *filename.TR1*.

The *nettl* daemons write to the *filename.TR0* file. When that file is full, the daemons copy the file contents to the *filename.TR1* file and purge the contents of the *filename.TR0* file. If the *filename.TR1* file already exists, the contents are destroyed. If that file does not exist, it is created.

The process of writing to the *filename.TR0* and copying to the *filename.TR1* continues for the duration of the trace. This process insures that the oldest

log data on the system is always in the *.TR1* file, and the latest log data is always in the *.TR0* file.

If no trace file is specified, trace records are written to the standard output file, usually the terminal.

The *nettl* daemons capture trace records in a trace buffer as they are received from a network subsystem. The daemons store the records there until they can write them to the trace file. In some cases, when large trace records are being produced very quickly or even when the system and the disks are heavily loaded, it is possible to lose trace records. To prevent this, you can increase the size of the trace buffer with the *-size* option.

Each trace entry is written in an internal binary format. It contains a header information field and a data field. The header information field includes a time stamp, a subsystem ID, a trace kind ID, and other miscellaneous fields. The data field contains the actual data that was transmitted or received.

You can use the *netfmt* command to view the trace data. This command formats the data in a readable formn that is suitable for viewing at a terminal screen or printing. The trace files contain messages for all network subsystems running on the machine. The *netfmt* command allows you to filter out messages in which you are not interested.

Because tracing is primarily used in a troubleshooting or debugging situation, users typically want to see trace data as it is created and act on it immediately. For this reason, trace data is often piped immediately to the *netfmt* command.

For example, consider the following command.

```
nettl -traceon pduin pduout -entity ns_ls_driver | netfmt -c
lantrace_filters
```

The command above causes the *nettl* daemons to collect *pduin* traces and *pduout* traces from the LAN card and to write the data to the standard output file. The *netfmt* command receives the trace data from the standard input file and writes the filtered and formatted record to the terminal. The filters are specified in the *lantrace_filters* file.

Because of the special relationship between the *nettl* daemons and the *netfmt* command, the shell is active when the *nettl* command is piped to *netfmt*. This is caused by the fact that trace data is not produced by the *nettl* command, but rather the *nettl* daemons. The *nettl* command starts the *nettl* daemons and exits. When the *nettl* command exits, the shell begins operation.

To turn off tracing, use the *nettl -traceoff* command.

The syntax and semantics of the *nettl* and *netfmt* commands are described later in this chapter. Since this tool is part of a larger subsystem used by other HP-UX products as well, only options applicable to LAN/9000 are described here.

nettl(1M)

Controls the network tracing and logging facility. (Requires super-user capability.) Only options applicable to LAN/9000 are shown here.

Syntax

```
          [-start]
          [-stop]
          [-status info]
nettl     [-traceon {kind...}} [-card dev_name] [-entity {subsystem...}]
          [-file filename] [-size limit] [-tracemax maxsize] [-m length]
          [-traceoff] [-entity {subsystem...}]
          [-log {class...}] [-entity {subsystem...}]
```

Options

`-start` Starts the *nettl* daemon, initializes the tracing and logging facility, and enables logging for all subsystems. The *nettl* daemon runs in the background and maintains the network tracing and logging system. Log messages are sent to the file named */usr/adm/nettl.LOGxx*. The suffix *xx*, 00 or 01, will be appended onto the filename. The default logging class for LAN/9000 is `error disaster` or 12. See the *-log* option.

This option may be abbreviated as `-st`.

Note We strongly recommended that the tracing and logging facility be started before any other networking. Otherwise, log data may be lost. The */etc/nettl -st* command should be placed before other networking commands in */etc/netlinkrc*. This is done automatically if you have configured your system with SAM.

`-stop`

Stops the *nettl* daemon, terminates the tracing and logging facility, and disables logging for all subsystems. The network should not be operated without the *nettl* daemon running.

This option may be abbreviated as `-sp`.

`-status` *info*

Reports tracing and logging status for all subsystems known to the *nettl* daemons. The *nettl* daemons must be running when you issue this command. *info* specifies the type of information that is to be displayed. The supported values are:

`log`	for logging status information
`trace`	for tracing status information
`ALL`	for logging and tracing status information

This option may be abbreviated as `-ss`.

`-traceon` *kind*

Starts tracing on the specified subsystem. The *nettl* daemon must be running when you issue this command. *kind* defines the trace masks used by the tracing facility before recording a message. You may enter either the keyword or mask as the *kind* value.

The supported values are:

KEYWORD	MASK	Meaning
`hdrin`	`0x80000000`	Header Received
`hdrout`	`0x40000000`	Header Transmitted
`pduin`	`0x20000000`	Packet Received
`pduout`	`0x10000000`	Packet Transmitted

This option may be abbreviated as `-tn`.

These values specify the incoming or outgoing packets or frames (depending on which level is being traced). You can combine masks as a single number. For example, to trace both `pduin` and `pduout`, you would specify `0x30000000` (the logical OR of `0x20000000` and `0x10000000`).

`-traceoff`

Disables tracing of subsystems specified with the *-entity* option. The trace file remains and you can format it to view the tracing messages.

This option may be abbreviated as `-tf`.

`-log` *`class`*

Controls the class of log messages that are enabled for the subsystems specified with the *-entity* option. The *nettl* daemon must be running when you issue this command.

This option may be abbreviated as `-l`.

class specifies the logging class. Available classes are:

Full	Abbrev	Mask
INFORMATIVE	I	1
WARNING	W	2
ERROR	E	4
DISASTER	D	8

You may specify *class* as a keyword or a numeric mask. The default logging classes are ERROR and DISASTER. The meanings of all of the possible *class* values are shown below.

`INFORMATIVE` Describes significant operations and activities of LAN/9000.

`WARNING` Indicates abnormal events or conditions which have no

permanent degradation of the integrity of LAN/9000.

ERROR	Indicates abnormal events or conditions which have no permanent degradation of the integrity of LAN/9000, but will cause a system call to fail and possibly an application program to terminate.
DISASTER	Signals an event or condition which WILL affect the overall subsystem or network operation and may cause several programs to fail or the entire card to shut down.

`-card` *dev_name*

Specifies the LAN/9000 interface card on which logging and tracing are to be initiated. This is the programmatic access name of the card. For example, */dev/lan_0*.

This option may be abbreviated as `-c`.

Note Only one LAN/9000 card may be traced at a time.

`-file` *name*

Initializes tracing and creates a file with the *name* specified and a suffix of *.TRCx* (where x = 0 or 1). All subsystems whose tracing is enabled with this command use this file. If *-file* is omitted, trace output will go to *stdout*. If the *-file* option is issued for a subsystem already being traced, the option is ignored unless that file is *stdout*.

When tracing is enabled, every operation through that entity is recorded if it conforms to the *kind* mask. This option may only be used with the *-traceon* options.

This option may be abbreviated as `-f`.

`-size` *`limit`*

Sets the trace buffer size (in kbytes). Trace messages will be held in the buffer until they are written to the file. Default value: 32 kbytes. Possible range: 1 to 512 kbytes. This option may only be used with the *-traceon* option.

This option may be abbreviated as `-s`.

`-m` *`length`*

Specifies the maximum number of bytes to trace. You may not need to capture the entire packet or frame. A number between 50 and 100 bytes is enough to capture the frame or packet header. The default is the entire packet or frame. This option may only be used with the *-traceon* option.

`-tracemax` *`maxsize`*

Specifies the maximum size of both trace files (*TRC0* and *TRC1*) combined. *maxsize* stands for the number of mbytes the combined size may be. The default size is 10. The range is from 1 to 999. If the trace buffer is not large enough to handle all incoming trace records, trace records can be lost. This option may only be used with the *-traceon* option.

`-entity` *`subsystem [subsystem]`*

Limits trace status information to the specified protocol layers. Some of the subsystems for LAN/9000 are:

```
NS_LS_COUNT   NS_LS_TCP
NS_LS_IPC     NS_LS_UDP
NS_LS_LAN0    NS_LS_RLBD
```

This option may only be used with the *-traceon* or *-log* option. This option may be abbreviated as `-e`. To obtain a complete list, run *nettl -ss ALL*.

netfmt(1M)

Formats common tracing and logging binary files. (Requires super-user capability.)

Syntax

```
netfmt [-c config_file] [-f input file] [-p]
       [-t records][-F] [-l] [-v]
```

Options

`-c config_file` — Specifies the file that contains formatter filter configuration commands. The *config_file* must be in the search path, or be a complete pathname. If you omit `-c` and the file contains trace data, *netfmt* uses the *$HOME/.nettr* file. If the file being formatted contains log data, *netfmt* uses the *$HOME/.netlog* file.

`-f input_file` — Specifies the file containing trace or log records recorded by *nettl*. If you don't specify `-f`, *netfmt* uses *stdin*. The file suffixes, *.LOG0X* or *.TRCX*, must be included in the *input_file* specification.

`-p` — Indicates *config_file* input is to be parsed. This allows you to perform a syntax check on the *config_file* specified with the `-c` option. The `-f` option is ignored. If the syntax is correct, *netfmt* terminates with no output or warnings.

`-t records` — Specifies the number of records to format from the end of the file. This allows you to quickly access the most recent information.

`-F` — Specifies that the input file is to be followed and not to be closed when end of file is encountered. *netfmt*

keeps it open and continues to read from it as new data arrives. This is helpful when you want to watch events as they occur while troubleshooting a problem, or to record events to a hard copy device for auditing.

`-l` Removes inverse video functions from the output stream. This option is useful if you are piping the output from *netfmt* to a non-video display device, such as, a line printer.

`-v` Causes verbose output for log messages. That is, CAUSE:, EFFECT:, and ACTION: messages to be displayed along with the standard log message.

The Formatting Filter Configuration File

This section describes the syntax and use of the *config_file* specified in the *netfmt* command with the `-c` option or the default file, *$HOME/.nettr* or *$HOME/.netlog*, used when log data is in the input file.

When *netfmt* begins operation, it reads and interprets the *config_file* specified with the `-c` option or the default file *.nettr* or *.netlog*. The *config_file* specifies filters that will serve to reduce the number of trace or log records that will be formatted and written to *netfmt's stdout* file. If no *config_file* can be found by *netfmt*, all records are formatted.

The *netfmt* reads the *config_file* from beginning to end. A filter enabled in the beginning of the file can be disabled in subsequent lines in the *config_file*. The filter types supported for LAN/9000 are `class`, `kind`, `subsystem`, `time_from`, and `time_through`.

When a trace or log record is read by *netfmt*, it compares the fields in the record to the filter settings specified in the *config_file*. If the record matches the filter settings, then the packet is formatted and written to *netfmt's stdout* file. Otherwise, the packet is discarded. If the record is not filtered out, then it is formatted and written to the output file.

Examples of nettl and netfmt Operation

Following are some examples of the *nettl* and *netfmt* commands.

Example 1

This example initializes the tracing/logging facility.

```
nettl -st
```

Example 2

This example changes log class to WARNING for all subsystems.

```
nettl -l WARNING -e ALL
```

Example 3

This example turns on tracing for the subsystems, `ns_ls_ip`, and `ns_ls_driver` (all types of tracing are enabled by OR'ny bit masks), and sends binary trace messages to file */usr/adm/trace.file*.

```
nettl -tn 0xffffffff -e ns_ls_ip ns_ls_driver -f
/usr/adm/trace.file
```

Example 4

This example determines trace status.

```
nettl -ss TRACE
```

The resulting information should resemble the following:

```
Date started:          Fri Apr 22 13:47:28 PDT 1988
 Trace Information:
 Trace File name:        /usr/adm/trace.file
 Uid:                    0 (root)  Buffer Size:     32
 KBytes
 Dropped Messages 0                Messages Queued:   0
```

```
Subsystem                    Trace Kind:
ns_ls_ip                     0xffffffff
ns_ls_driver                 0xffffffff.
```

Example 5

This example stops tracing/logging:

```
nettl  -sp
```

Example 6

The following command reads the file */usr/adm/trace_file.TRC1* for the binary data and uses the *conf.file* as the filter configuration file.

```
netfmt -f /usr/adm/trace_file.TRC1 -c conf.file
```

Example 7

The following command formats the last 50 records in the file */usr/adm/log.fileLOG00* (the default log file).

```
netfmt -f /usr/adm/log.file.LOG00 -t 50
```

Example 8

The following command uses the follow option (-F) and the configuration file to send disaster log messages to the console.

```
netfmt -F -c DISASTER.ONLY < /usr/adm/log.file.LOG00 > /dev/console
```

DISASTER.ONLY contains

```
-------------------------------------------------------------
  SUBSYSTEM      REQUEST_TYPE     ARGUMENT1      ARGUMENT2
-------------------------------------------------------------

FORMATTER        filter           class           !*
FORMATTER        filter           class           DISASTER
```

Filter Command Lines

Each command line specifies a criterion for selecting trace and log records from the input file.

General Format of the Filter Configuration File

netfmt interprets the configuration file according to the following rules:

- Data in the configuration file is interpreted a line at a time.
- A line beginning with a pound sign (#) is a comment. Comments are terminated by a newline (end-of-line characters). All other characters appearing in a comment are ignored.
- Each filter command must appear on a separate line.
- White space such as spaces and tabs may be used freely to format filter command lines. A blank line is a valid construction.
- Keywords within a filter command line are case independent. For example, "error" is not distinguished from "ERROR".

Syntax

```
filter  type [!] {value | *}
```

Filter Types

type	is one of the following keywords: `class`, `kind`, `time_from`, `time_through`, and `subsystem`.
!	Negates the following value. Instead of selecting all records whose *value* matches the *value* specified in the filter command, the exclamation point matches all records whose *value* does not match the specified *value*.

value — is entirely dependent on the keyword specified for *type*.

* — is always interpreted to mean all possible values. You can use it along with an exclamation point, "!*" to mean "not all" This *value* is not valid for `time_from`, and `time_through`.

type Keyword Descriptions

`class` — By default all log classes are formatted. To select only records of a single `class`, turn off all log classes with the *filter class !* filter* command, and then specify one of the single classes listed below.

The possible *values* for the `class` type and their meanings are:

INFORMATIVE	Describes significant operations and activities of LAN/9000.
WARNING	Indicates abnormal events or conditions which have no permanent degradation of the integrity of LAN/9000.
ERROR	Indicates abnormal events or conditions which have no permanent degradation of the integrity of LAN/9000, but will cause a system call to fail and possibly an application program to terminate.
DISASTER	Signals an event or condition which WILL affect the overall subsystem or network operation and may cause several programs to fail or the entire card to shut down.

`kind`

Trace type or mask type. A mask is a hexadecimal representation of a (set of) trace kind(s). You may enter either a keyword or mark as the *kind* value. The trace kinds and their corresponding masks are:

`hdrin`	`0x80000000`
`hdrout`	`0x40000000`
`pduin`	`0x20000000`
`pduout`	`0x10000000`

`subsystem`

A complete list of subsystem names is available in the *nettl(1M)* man page. You can also list them out with the *nettl -status all* command. By using combinations of the operators below, it is possible to specify only a few subsystems to format out of a file containing many possible subsystems.

No more than one subsystem name may be given per line; multiple lines will "OR" the request. You can turn off the subsystem name with the ! operator, given all subsystem except the one(s) indicated. Also, all subsystems may be specified with the * operator (default), or all subsystems turned off with the !* operator.

`time_from`

Starting time of the trace and log records to be formatted. Records whose time stamp comes before the specified time and date are not formatted. *value* has the following format: `hh:mm:ss.dddddd MM/DD/YY`, where the following meaning applies:

hh	stands for hours from 00 to 24.
mm	stands for minutes from 00 to 59.
hh	stands for seconds from 00 to 59.

	hh	stands for microseconds from 000000 to 999999.
	MM	stands for months from 00 to 12.
	DD	stands for days from 00 to 31.
	YY	stands for years from 00 to 99.
`time_through`	Ending time of the trace and log records to be formatted. Records whose time stamp comes later than the specified time and date are not formatted. *value* has the following format: `hh:mm:ss.dddddd MM/DD/YY`. The syntax and semantics for this construction is described above.	

Examples

The following examples show formatting filter commands in the configuration file.

Example 1

This example formatting file instructs *netfmt* to format only INFORMATIVE messages coming from the `ns_ls_ip` subsystem that occurred from 10:31:58 to 10:41:00 on November 23, 1988.

```
REQUEST_TYPE      ARGUMENT1      ARGUMENT2

filter            time_from      10:31:53   11/23/88
filter            time_through   10:41:00   11/23/88
filter            class          !*
filter            class          INFORMATIVE
filter            subsystem      !*
filter            subsystem      ns_ls_ip
```

Example 2

This example formatting command file instructs *netfmt* to format only `pduin kind` coming from the `ns_ls_driver` subsystem for the process 10289.

```
 ..............  REQUEST_TYPE  ARGUMENT1
filter           kind          pduin
filter           subsystem     !*
filter           subsystem     ns_ls_driver
filter           process_ID    10289
```

A

Installation Error Messages

This appendix lists and describes error messages that can be produced during installation and configuration of LAN/9000. It contains the following sections:

- Installation Messages.
- Configuration Messages.

Installation Messages

The following ASCII messages may be returned by the *update* utility program as you attempt to load network software.

MESSAGE **Could not change into the new directory *uxgenname*. You will have to perform the kernel generation manually as outlined in the installation guide.**

CAUSE The *update* program could not change into the new *uxgen* directory *uxgenname*.

ACTION Continue the installation manually.

MESSAGE **No file named *uxgenname*, consult installation guide.**

CAUSE The *update* program could not locate the new *uxgen* file *uxgenname*.

ACTION Continue the installation manually.

MESSAGE **Parsing of input file failed. You will have to perform the kernel generation manually as outlined in the installation guide.**

CAUSE The *update* program could not remove comment delimiters from your *uxgen* input file.

ACTION Continue the installation manually.

MESSAGE	**Storage of new kernel failed. You will need to make enough room in the root partition then restart the update process.**
CAUSE	The root directory (/) did not contain enough room for the new kernel created with NS/9000 and/or NS/9000 libraries.
ACTION	Move or remove unneeded files from the root directory. Retry the *update* program.

MESSAGE	**Storage of old kernel failed. You will need to make enough room in the root partition then restart the update process.**
CAUSE	The root directory (/) did not contain enough room for the backup kernel.
ACTION	Move or remove unneeded files from the root directory and retry the *update* program.

MESSAGE	**Storage of new uxgen input file failed. You will need to make enough room in the root partition then restart the update process.**
CAUSE	The root directory (/) did not contain enough room for the new *uxgen* input file.
ACTION	Move or remove unneeded files from the root directory and retry the *update* program.

MESSAGE	**Storage of old uxgen input file failed. You will need to make enough room in the root partition then restart the update process.**
CAUSE	The root directory (/) did not contain enough room for the old *uxgen* input file.
ACTION	Move or remove unneeded files from the root directory and retry the *update* program.

MESSAGE	**The core kernel has not been updated. Consult the installation guide. Then update the kernel.**
CAUSE	The core kernel has not been updated to the current HP-UX release.
ACTION	Update the core kernel to the current HP-UX release and try again.

MESSAGE	**The core kernel must be updated first. Consult the installation guide. Then update the kernel.**
CAUSE	The *uxgen* input file has not been updated to the current HP-UX software release.
ACTION	Make sure you have the current HP-UX release software. Update the core kernel to the current HP-UX release and try again. If you are unsuccessful, contact your HP representative.

MESSAGE **The library libprot.a is not present. This update will not work.**

CAUSE The library file */etc/conf/libprot.a* was not installed with the other LAN/9000 files.

ACTION Make sure you have the current LAN/9000 software. Retry the installation. If you are unsuccessful, contact your HP representative.

MESSAGE **The link library libns.a is not present. This update will not work.**

CAUSE The library file */etc/conf/libns.a* was not installed with the other LAN/9000 files.

ACTION Make sure you have the current LAN/9000 software. Retry the installation. If you are unsuccessful, contact your HP representative.

MESSAGE **The link tape has not been updated yet. You must do that first before a new kernel can be created.**

CAUSE You installed ARPA Services/9000 or NS/9000 before installing LAN/9000. No harm has been done; the tapes have just been installed out of order.

ACTION Install LAN/9000 before you install NS/9000 or ARPA Services/9000.

MESSAGE **Uxgen could not complete. You will have to perform the kernel generation manually as outlined in the installation guide.**

CAUSE The *update* program could not generate a new kernel.

ACTION Continue the installation manually.

MESSAGE **You do not have the required lan0 line. You will have to update manually. Consult the installation guide.**

CAUSE The *update* program could not find the *lan0* line in the *uxgen* input file.

ACTION Add the following line to your *uxgen* input file:

```
lan0 lu 0 address 4;
```

Continue the installation manually.

MESSAGE **You do not have the required nsdiag0 line. You will have to update manually. Consult the installation guide.**

CAUSE The *update* program could not find the *nsdiag0* line in the *uxgen* input file.

ACTION Add the following line to your *uxgen* input file:

```
include nsdiag0;
```

Continue installation manually.

MESSAGE **You do not have the required nsnsipc0 line. You will have to update manually. Consult the installation guide.**

CAUSE The *update* program could not find the *nsnsipc0* line in the *uxgen* input file.

ACTION Add the following line to your *uxgen* input file:

```
include nsnsipc0;
```

Continue the installation manually.

MESSAGE **Symbolic link of /etc/yp to /usr/etc/yp failed.**

CAUSE */etc/yp* is already present.

ACTION Remove */etc/yp*.

Configuration Messages

The following error messages may be returned by the nodal management commands *nodename(1)*, *route(1M)*, *netstat(1)*, and *ifconfig(1M)*.

MESSAGE **permission denied**

CAUSE Permission to execute either the *nodename(1)* or *ifconfig(1M)* commands was denied.

ACTION You must be a super-user to use the *nodename(1)* command to configure a node name or to set flags; you must also be a super-user to use the *ifconfig(1M)* command to configure an IP address or set flags.

MESSAGE **invalid node name syntax**

CAUSE The syntax specified for the node name was invalid.

ACTION Check the syntax and try again.

MESSAGE **nodename not yet configured**

CAUSE The *nodename(1)* command was used to print the node name before the *nodename(1)* command was used to configure the system node name.

ACTION Use *nodename(1)* to configure the system node name.

MESSAGE **unexpected error returned from IPC:** *errno*

CAUSE A node management command invoked a NetIPC call that returned an error. A NetIPC error code is returned in *errno*.

ACTION Refer to the error codes listed in the following appendix for the meaning of *errno*.

MESSAGE	**no such interface**
CAUSE	The interface name passed to *ifconfig(1M)* does not exist on the system.
ACTION	Check the spelling and names of interfaces on the system.

MESSAGE	**invalid internet address**
CAUSE	The internet address specified was not in the proper form.
ACTION	Check the syntax and try again.

MESSAGE	**IPCCREATE returned error:** *errno*
CAUSE	The NetIPC call *ipccreate()* returned an error. The error code is returned in *errno*.
ACTION	Refer to the error codes listed in the following appendix for the meaning of *errno*.

MESSAGE	**message catalog can't be opened/accessed for language** *lang*. **Language n-computer will be used instead.**
CAUSE	This error can be returned from the *ifconfig(1M)*, *netstat(1)*, *nodename(1)*, *route(1M)*, and *rlb(1M)* commands. The message catalog for language *lang* isn't in */usr/lib/nls/lang*.
ACTION	Verify that the $LANG variable is set to the correct language. If so, you need to install the desired message catalog.

MESSAGE	**ipaddr must be set also**
CAUSE	The super-user attempted to set the subnet mask with *ifconfig(1M)* without specifying an IP address.
ACTION	Execute the *ifconfig(1M)* command again, specifying both the IP address and the subnet mask.

MESSAGE	**ifconfig option *bad_opt* is not supported**
CAUSE	Option *bad_opt* is invalid.
ACTION	Check spelling and names of LAN interfaces on the system and try again.

MESSAGE	**route: socket: permission denied**
CAUSE	A non-super-user attempted to alter the route table.
ACTION	Gain super-user access rights or contact the node manager to alter the route table.

MESSAGE	**not in table**
CAUSE	The super-user tried to delete entry in the route table that does not exist.
ACTION	Check destination and gateway addresses or symbolic names and execute the *route delete* command again.

MESSAGE	**entry in use**
CAUSE	The super-user tried to add an entry to the route table that already exists.
ACTION	Delete the existing route and add a new one.

MESSAGE	**routing table overflow**
CAUSE	You have the maximum number of routes in your routing table.
ACTION	Delete a route entry no longer used and then add the new entry. Execute the *route delete* command again.

B

Diagnostics Error Messages

This appendix lists and describes error messages that are returned by network diagnostics. It contains the following sections:

- *ping(1M)* Messages.
- *rlb(1M)* Messages.

ping(1M) Messages

MESSAGE **packet size too big, maximum size is 2048 bytes**

CAUSE The value entered for *packet_size* in the *ping(1M)* command exceeded the limit for that argument. The size limit is 2048. *ping(1M)* has terminated.

ACTION Execute *ping(1M)* again with a smaller *packet_size*.

MESSAGE **packet size too small, minimum is 8 bytes**

CAUSE The value entered for *packet_size* is less than the minimum value allowed for that argument. The minimum value allowed is 8 bytes. *ping(1M)* has terminated.

ACTION Execute *ping(1M)* again with a larger *packet_size*.

MESSAGE **unknown host *hostname***

CAUSE The host name was not found in the */etc/hosts* file. *ping(1M)* has terminated.

ACTION Check the spelling of the *host* parameter. If it is correct, ask the node manager to add it to the */etc/hosts* file. You can also use the IP address for the remote host.

MESSAGE **socket: File table overflow**

CAUSE There are too many open files and sockets in the system at this time.

ACTION This error causes *ping(1M)* to pause for 5 seconds before trying again. This is a temporary situation. You can either let *ping(1M)* continue to try, or terminate and try later.

MESSAGE	**socket: Host is down**
CAUSE	The local host does not have the network powered up.
ACTION	This error causes *ping(1M)* to pause for 5 seconds before trying again. Ask the node manager to power up the network.

MESSAGE	**socket: No buffer space available**
CAUSE	Currently, there is not enough networking memory available for *ping(1M)* to execute.
ACTION	This error causes *ping(1M)* to pause for 5 seconds before trying again. This is probably a temporary situation. If you allow it to continue, *ping(1M)* may find enough memory. Alternatively, you may terminate *ping(1M)* and try again later.

MESSAGE	**socket: Permission denied**
CAUSE	*ping(1M)* has not been set up for execution by users other than the super-user.
ACTION	This error causes *ping(1M)* to pause for 5 seconds before trying again. Terminate *ping(1M)*. Log in as a super-user or ask the node manager to help you extend your super-user privileges for *ping(1M)*.

MESSAGE	**recvfrom:** *errmessage*
CAUSE	An error, described by *errmessage*, occurred while the local host was receiving data.
ACTION	This error requires HP notification.

MESSAGE **sendto: Interrupted system call ping: wrote *hostname n* chars, ret=-1**

CAUSE *ping(1M)* was interrupted by a signal while trying to send an *n*-byte packet to host *hostname*.

ACTION This can only occur if someone is sending *SIGALARM* signals to *ping(1M)*. It is not a fatal problem, but it may result in showing lost packets in the final statistics.

MESSAGE **sendto: No buffer space available ping: wrote *hostname n* chars, ret=-1**

CAUSE There is not enough networking memory available for *ping(1M)* to send the *n*-byte packet to host *hostname*. It could result in *ping(1M)* reporting lost packets.

ACTION This is probably a temporary situation. You can either let *ping(1M)* continue or terminate and execute *ping(1M)* again later.

MESSAGE **sendto: No route to hostping: wrote *hostname n* chars, ret=-1**

CAUSE There was no response from the remote host. This could occur if the remote host does not have the network powered up, the remote host computer is turned off, or the remote host does not support ARP.

ACTION Terminate *ping(1M)*. Resolve the problem given the suggestions above and try again, or try a different remote host.

MESSAGE	**sendto:** ***errormessage*** **ping: wrote** ***hostname n*** **chars, ret=-1**
CAUSE	*ping(1M)* received the error indicated by *errormessage* while trying to send an *n*-byte packet to host *hostname*.
ACTION	This error requires HP notification.

MESSAGE	**wrote** ***hostname n*** **chars, ret=***m*
CAUSE	*ping(1M)* tried to send a packet of *n* bytes to host *hostname*. Only *m* bytes were sent.
ACTION	This error requires HP notification.

MESSAGE	**network is unreachable**
CAUSE	Incorrect IP address.
ACTION	Correct the IP address. Nodes should have the same network number.

rlb(1M) Messages

The following error messages are generated in the Remote Communications Mode of *rlb(1M)*.

MESSAGE **All nodes interrupted by operator.**

CAUSE The operator interrupted the Remote Communications Mode *all* command during its execution. The interruption is usually caused by the operator hitting the **[Break]** key. A summary of the exchanges up to that time is displayed.

ACTION This is an informational message only. No action is necessary.

MESSAGE **Communications terminated by operator hitting BREAK.**

CAUSE The operator terminated a message exchange with a remote node before the exchange sequence was complete. A summary of the exchange up to that point is displayed.

ACTION This is an informational message only. No action is necessary.

MESSAGE **Connection response error.**

CAUSE An error occurred while waiting for a connection response from a remote node. The system generated error code follows this message. Possible causes are: (1) The remote node may not be powered up on the network, or may not have the LAN/9000 software powered up; (2) The Remote Loopback Protocol daemon may not be powered up on the remote node; (3) The LAN Interface may have failed on the remote or local node; (4) A cabling problem may have occurred; (5) The remote node may be unable to accept connections due to congestion or lack of memory.

ACTION Possible actions are: (1) Power up the remote node on the network or power up the LAN/9000 software on the remote node; (2) Power up the Remote Loopback Protocol daemon on the remote node; (3) Check the LAN Interface on the remote and local node; (4) Check the cable; (5) Try again later.

MESSAGE **Error reading node name file:** ***nodefilename.***

CAUSE An error occurred while attempting to read a node name from the node name file *nodefilename.*

ACTION Check the node name file. Refer to the system generated error code follows this message for more information.

MESSAGE	**Error trying to receive data.**
CAUSE	An error occurred while attempting to read the response message from the remote node. Possible causes are: (1) The no-response timeout may be too small; (2) The network may be busy or congested; (3) The remote node may have been powered down; (4) The Remote Loopback Protocol server may have been killed; (5) The LAN Interface may have failed; (6) A cabling problem may have occurred.
ACTION	The system generated error message or code follows this message. Fix the problem according to the returned message.

MESSAGE	**Error trying to send data.**
CAUSE	An error occurred while attempting to send a message to a remote node. Possible causes are: (1) The no-response timeout may be too small; (2) The network may be busy or congested; (3) The remote node may have been powered down; (4) The Remote Loopback Protocol server may have been killed; (5) The LAN Interface may have failed; (6) A cabling problem may have occurred.
ACTION	The system generated error message or code follows this message. Fix the problem according to the returned message.

MESSAGE	**Error trying to shutdown the connection.**
CAUSE	An error occurred while attempting to shut down a connection to a remote node.
ACTION	The system generated error message or code follows this message. Fix the problem according to the returned error code's message.

MESSAGE	**INCOMPLETE EXCHANGE with node *nodename*.**
CAUSE	*rlb(1M)* was unable to exchange all of the requested messages with the remote node *nodename*. The operator may have hit the Break key, an error may have occurred while trying to send/receive data or the response data may differ from the transmitted data.
ACTION	This message is followed by a display of how many of the total number of messages were exchanged and if there were any messages with transmit/receive data that differed. Refer to these messages for more information.

MESSAGE	**Length must be integer between 10 and 1450. The operator specified an invalid value for the message length.**
CAUSE	The specified value is not within limits.
ACTION	Specify a new value.

MESSAGE	**Maximum messages you are authorized to exchange is 10. That value has been substituted.**
CAUSE	An operator who is not super-user attempted to set the number of messages to exchange to a value greater than 10. The value has been set to 10.
ACTION	Talk to the node manager if you need super-user capabilities.

MESSAGE	**Name is too long, it cannot exceed 50 characters.**
CAUSE	A remote node name is longer than 50 characters.
ACTION	Check the node name.

MESSAGE	**Number of messages must be integer 0.**
CAUSE	The operator specified an invalid value for the number of messages to exchange with remote nodes.
ACTION	The number must be an unsigned integer greater than 0 and less than or equal to 2**31 -1.

MESSAGE	**Received message exceeded input buffer size.**
CAUSE	The response message from the remote node was larger than the maximum buffer used by *rlb(1M)* to send messages. *rlb(1M)* will attempt to resynchronize with the remote node by repeatedly reading input data until the end-of-message designator is read.
ACTION	If the *continue when transmit/receive data differ* option is enabled, *rlb(1M)* then continues the message exchange. If *rlb(1M)* cannot resynchronize with the remote node, try again with different message lengths. This error requires HP notification.

MESSAGE	**Timeout must be integer between 1 and 600. The operator specified an invalid value for the timeout period.**
CAUSE	The specified value is not within limits.
ACTION	Specify a new value.

MESSAGE	**Transmit/Receive data differ.**
CAUSE	The data portion of the message sent does not match the data portion of the message received back from the remote node.
ACTION	If the *continue when transmit/receive data differ* option is enabled, the message exchange continues after this error message is reported. Try again with different message lengths. This error requires HP notification.

MESSAGE	**Transmit/Receive message lengths differ.**
CAUSE	The length of the message sent does not match the length of the message received back from the remote node. This message is followed by a display of the length of the transmitted and received messages.
ACTION	If the *continue when transmit/receive data differ* option is enabled, the message exchange continues after this error message is reported. Try again with different message lengths. This error requires HP notification.

MESSAGE	**Trigger must be integer between 10 and 10000.**
CAUSE	The operator specified an invalid trigger value.
ACTION	The value must be between 10 and 10000 milliseconds.

MESSAGE **Unable to find node name file: *nodenamefile*.**

CAUSE An attempt was made to open a node name file which did not exist. If any other errors occur while attempting to open a node name file, this error message is displayed.

ACTION Check the node name file.

MESSAGE **Unable to open node name file: *nodefilename*.**

CAUSE *rlb(1M)* was unable to open an existing node name file with the name *nodefilename*. This file name is passed using the *name* command in Remote Communications mode. It is supposed to hold a list of node names for the diagnostic to attempt to exchange messages with.

ACTION Check the file according to the returned error code's message. If the problem persists, this error requires HP notification.

MESSAGE **Unable to send complete message.**

CAUSE An error occurred while sending a message to a remote node. This message is followed by a display of how many of the total bytes in the message were sent.

ACTION If the *continue when transmit/receive data differ* option is enabled, the message exchange continues after this error message is reported. Try again with different message lengths. This error requires HP notification.

MESSAGE	**Node does not exist**
CAUSE	The destination node name may have been typed incorrectly, the destination node may be down, or *ifconfig* was not used correctly on the remote node.
ACTION	Retry, "up" the destination node, or re-execute *nodename* on the remote node.

MESSAGE	**System feature not installed.**
CAUSE	The NetIPC fileset is not installed.
ACTION	*rlb* cannot be used without the NetIPC fileset. Either choose another diagnostic or install the NetIPC fileset.

C

Network Event Logging Messages

This appendix lists the log messages that are returned by subsystems of the LAN/9000 products. If these products are installed on your node, and network logging is enabled, the messages may be written to the console or a file.

Subsystem: IP

5001	MESSAGE	**ICMP error message generated: type** *type* **code** *code* **destination address** *address*.
	CAUSE	Normal protocol operation.
	ACTION	None. This message is for diagnostic purposes only.
5005	MESSAGE	**ICMP packet input: type** *type* **code** *code* **destination address** *address*.
	CAUSE	Normal protocol operation.
	ACTION	None. This message is for diagnostic purposes only.

Subsystem: LAN

1000	MESSAGE	**LAN driver failed to BIND to its associated HW. The LAN manager index is *mgr_index*. Check LAN HW and system I/O configuration.**
	CAUSE	(Disaster) This is probably due to a hardware problem or too many LAN cards in the backplane. The *mgr_index* field is for HP internal use only.
	ACTION	Use the *ioscan(1M)* command to find the bind error code. Make sure the number of LAN cards does not exceed the maximum allowed. Refer to explanation of the error codes in the *ioscan(1M)* man page to resolve the problem.

1001	MESSAGE	**LAN CTRL message reply HW_ERROR on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) Reply to CIO CTRL request indicates hardware error.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.

1002	MESSAGE	**LAN DMA message reply HW_ERROR on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) Reply to CIO DMA request indicates hardware error.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1005	MESSAGE	**LAN driver software TIMEOUT_ERROR on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) DMA or CTRL timer has expired. The LAN card is not responding or processing requests.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1008	MESSAGE	**LAN card status of DEAD_OR_DYING on interface unit *if_unit*; reset or reboot. The card error code is *status*.**
	CAUSE	(Disaster) The LAN driver has received DEAD_OR_DYING status from the LAN card. This is an unrecoverable error. The *status* field is for HP internal use only.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.

1009	MESSAGE	**LAN card status of PROTOCOL ERROR on interface unit *if_unit*; reset or reboot. The card error code is *status*.**
	CAUSE	(Disaster) The LAN driver has received *PROTOCOL_ERROR* status from the LAN card. Though usually recoverable, protocol error status stops all card backplane/frontplane communication. The *status* field is for HP internal use.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1011	MESSAGE	**LAN LLIO hardware problem on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) The LAN driver has received a CIO DMA reply from a Low Level I/O indicating a hardware problem.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.

1014	MESSAGE	**LAN driver INTERNAL software error on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) The LAN driver has detected an internal state error.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1015	MESSAGE	**LAN card status SELF_TEST_FAIL; reset or reboot interface unit *if_unit*.**
	CAUSE	(Disaster) LAN card self-test failed.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1016	MESSAGE	**LAN card status WRITE_TEST_FAILURE; reset or reboot interface unit *if_unit*.**
	CAUSE	(Disaster) LAN card write-test failed.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.

1017	MESSAGE	**(Disaster) LAN card status DRIVER_TIMEOUT or BAD_CONTROL (APR); reset or reboot interface unit** *if_unit*.
	CAUSE	The LAN driver has written a bad control request to the card or has failed to handshake with the card for over one minute.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1018	MESSAGE	**LAN card status UNKNOWN_HARDWARE_ERROR; reset or reboot interface unit** *if_unit*.
	CAUSE	(Disaster) The LAN card has set the error status to an unknown state.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1020	MESSAGE	**LAN driver could not log HP RESERVED SAP, Type or Canonical Address of** *value*; **reboot.**
	CAUSE	(Disaster) Internal software error.
	ACTION	Reboot. If rebooting does not solve the problem, notify your HP representative.

1021	MESSAGE	**LAN card is OFFLINE on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) Hardware problem or LAN driver state inconsistency.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1022	MESSAGE	**LAN card ACTIVE STATION ADDRESS CHANGE filed on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) The LAN driver failed to change the active station address of the LAN card. This is probably due to a hardware problem.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Reset the card. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.

1023	MESSAGE	**LAN card has an invalid HARDWARE ID; check HW and system I/O configuration. The interface unit is *if_unit*. The expected hardware ID was *exp_id*; *bad_id* was returned instead.**
	CAUSE	(Disaster) Wrong LAN card or no LAN card in backplane.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Use the *ioscan(1M)* command to check for bind errors. Make sure the correct LAN card is being used. If the problem persists, notify your HP representative.
1024	MESSAGE	**LAN driver failed to read PERMANENT STATION ADDRESS on interface unit *if_unit*; reset or reboot.**
	CAUSE	(Disaster) LAN card failed to read address on interface unit.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Use the *ioscan(1M)* command to check for bind errors. Make sure the correct LAN card is being used. If the problem persists, notify your HP representative.
2001	MESSAGE	**LAN driver received a packet TOO SHORT or TOO LONG on interface unit *if_unit*. The length is *pkt_leng* and the mbuf address is *m_addr*.**
	CAUSE	(Error) This is probably due to a LAN card problem. The *m_addr* field is for HP internal use only.
	ACTION	If this problem persists, notify your HP representative.

2002	MESSAGE	**LAN card status PROTOCOL_EROR; the interface unit *if_unit* is being reset. The error code is *status*.**
	CAUSE	(Error) The LAN driver has received a Write Data Protocol Error status from the LAN card. A powerfail may have occurred on a remote bus in the backplane. The LAN card is being reset by the driver. The *status* field is for HP internal use only.
	ACTION	If the problem persists, notify your HP representative.
2003	MESSAGE	**LAN DMA or CTRL request TIMER popped; the interface unit *if_unit* is being reset. The timer event counter is *count*.**
	CAUSE	(Error) The DMA or CTRL timer has expired. The LAN driver is resetting the LAN card. The *count* field is for HP internal use only.
	ACTION	If the problem occurs again, the LAN driver will try repeatedly to reset the card before logging a disaster (1005). No action is necessary.
3007	MESSAGE	**LAN power failure; CONTROL REQUEST was ABORTED on interface unit *if_unit*.**
	CAUSE	(Warning) A power failure caused a control request to abort.
	ACTION	Repeat the control request.

3016	MESSAGE	**LAN card status LINE_ERROR (LAW); check link attached to interface unit *if_unit*.**
	CAUSE	(Warning) The LAN card has reported a non-fatal line error. The LAN driver will continue to attempt normal operations.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card. Check the cable and the MAU connection. If the problem persists, notify your HP representative.
4001	MESSAGE	**LAN driver cannot allocate MEMORY to post read buffer to LAN card.**
	CAUSE	(Resource Limitation) No memory available at this time.
	ACTION	This is an informational message only. No action is necessary.
4002	MESSAGE	**LAN driver has dropped an outbound packet due to insufficient memory.**
	CAUSE	(Resource Limitation) The LAN driver had not reserved sufficient memory for an extra long outbound packet. The packet has been dropped.
	ACTION	If the problem persists, notify your HP representative.

4007	MESSAGE	**LAN driver dropped LLA outbound packet. No room on outbound queue. The interface unit is *if_unit*.**
	CAUSE	(Resource Limitation) A LLA outbound packet was dropped because the outbound queue was full.
	ACTION	If the problem persists, you may be overloading your network. Reduce network overhead or notify your HP representative.
5000	MESSAGE	**LAN driver has a pending write request on interface unit *if_unit*, but the network interface output queue is EMPTY.**
	CAUSE	(Protocol Log) This is probably due to a protocol or LAN driver state inconsistency or software timing problem.
	ACTION	If the problem persists, notify your HP representative.
5008	MESSAGE	**LAN driver dropped inbound 802.3 packet due to unsupported or invalid CONTROL field. The interface unit is *if_unit*.**
	CAUSE	(Protocol Log) An inbound IEEE 802.3 packet was dropped because of an invalid CTRL field in the packet header.
	ACTION	This is an informational message only. No action is necessary.

5035	MESSAGE	**LAN driver logged 802.3 Destination Service Access Point (DSAP)** *d_sap* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An IEEE SAP *d_sap* was successfully logged by a protocol wishing to receive packets at that DSAP on this node.
	ACTION	This is an informational message only. No action is necessary.
5036	MESSAGE	**LAN driver logged Ethernet Type** *type* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An Ethernet Type *type* was successfully logged by a protocol wishing to receive packets at that TYPE on this node.
	ACTION	This is an informational message only. No action is necessary.
5037	MESSAGE	**LAN driver logged HP Canonical Address** *c_addr* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An HP Canonical Address *c_addr* was successfully logged by a protocol wishing to receive packets at that address on this node.
	ACTION	This is an informational message only. No action is necessary.

5038	MESSAGE	**LAN driver unlogged IEEE 802.3 Destination Service Access Point (DSAP)** *d_sap* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An IEEE DSAP *d_sap* was successfully dropped by a protocol no longer wishing to receive packets at that DSAP on this node.
	ACTION	This is an informational message only. No action is necessary.
5039	MESSAGE	**LAN driver unlogged Ethernet Type** *type* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An Ethernet Type *type* was successfully dropped by a protocol no longer wishing to receive packets at that Type on this node.
	ACTION	This is an informational message only. No action is necessary.
5040	MESSAGE	**LAN driver unlogged HP Canonical Address** *c_addr* **on interface unit** *if_unit*.
	CAUSE	(Protocol Log) An HP Canonical Address *c_addr* was successfully dropped by a protocol no longer wishing to receive packets at that address on this node.
	ACTION	This is an informational message only. No action is necessary.

5041	MESSAGE	**LAN driver DROPPED packet destined for unlogged DSAP *d_sap* on interface unit *if_unit*.**
	CAUSE	(Protocol Log) An inbound IEEE 802.3 packet was discarded by the LAN driver. This is probably because the DSAP *d_sap* had not been previously logged, or because the network interface was down or was not configured to receive IEEE packets.
	ACTION	This is an informational message only. No action is necessary.

5042	MESSAGE	**LAN driver DROPPED packet destined for unlogged Type *type* on interface unit *if_unit*.**
	CAUSE	(Protocol Log) An inbound Ethernet packet was discarded by the LAN driver. This is probably because the Type *type* had not been previously logged, or because the network interface was down or was not configured to receive Ethernet packets.
	ACTION	This is an informational message only. No action is required.

5043	MESSAGE	**LAN driver DROPPED packet destined for unlogged Canonical Address *c_addr* on interface unit *if_unit*.**
	CAUSE	(Protocol Log) An inbound packet in the HP Canonical Addressing format (HP Extended SAP or Extended Type) was discarded by the LAN driver. This is probably because the address *c_addr* had not been previously logged, or because the network interface was down or was not configured to receive IEEE or Ethernet packets.
	ACTION	This is an informational message only. No action is necessary.
5047	MESSAGE	**LAN driver DROPPED packet encoded in Trailing Header format on interface unit *if_unit*.**
	CAUSE	(Protocol Log) An inbound packet in Berkeley Trailer Format was discarded by the LAN driver. This is probably due to the packet having an incorrect format.
	ACTION	This is an informational message only. No action is necessary.

Subsystem: PROBE

2012	MESSAGE	**ARP packet duplicate IP address &A from %04.4hx-%04.4hx-%04.4hx.**
	CAUSE	(Error) Two machines on the same network are using the same internet address. The internet address is returned in "dot" format in the IP_addr field. The station address (also called the link-level address) is returned in hexadecimal in the station_addr field.
	ACTION	Identify which machine is in error and alter its internet address.

Subsystem: TCP

5019	MESSAGE	**TCP connection change of state: old** *old*, **new** *new*, **lport** *local port*, **fport** *remote port*, **and faddr** *remote address*.
	CAUSE	Normal protocol operation.
	ACTION	None. This message is for diagnostic purposes only.
5020	MESSAGE	**TCP sent a RST: lport** *local port*, **fport** *remote port*, **faddr** *remote address*
	CAUSE	Normal protocol operation.
	ACTION	None. This message is for diagnostic purposes only.

LAN Interface Card Statistics

This appendix contains descriptions of the status values and statistics for the Series 300/400, Series 600/800, and Series 700 LAN interface cards. The statistics kept by the local LAN card can be listed with the *display* command in the *landiag* LAN Interface Test Mode.

The display for the Series 300/400 LAN interface status is shown in Figure D-1, the display for the Series 600/800 is shown in Figure D-2, and the display for the Series 700 is shown in Figure D-3. Following is a description of each field.

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986     08:51:29

Device file                        = /dev/lan
Select code                        = 21
Current state                      = active
LAN Interface address, hex         = 0x008009000636
Number of multicast addresses      = 5
Frames received                    = 107983
Frames transmitted                 = 113587
Undelivered received frames        = 11
Untransmitted frames               = 7
CRC errors received                = 0
Transmit collisions                = 1528
One transmit collision             = 68
More transmit collisions           = 730
Excess retries                     = 0
Deferred transmissions             = 0
Carrier lost when transmitting     = 0
No heartbeat after transmission    = 0
Frame alignment errors             = 0
Late transmit collisions           = 0
Frames lost                        = 0
Unknown protocol                   = 0
Bad control field                  = 0
```

Figure D-1. Series 300/400 LAN Interface Status Display

```
                    LAN INTERFACE STATUS DISPLAY
                     Fri,Mar 21,1986     08:51:29

Device file                       = /dev/lan0
Lu number                         = 0
Current state                     = active
LAN Interface address, hex        = 0x080009000000
Number of multicast addresses     = 2
Frames received                   = 107983
Frames transmitted                = 113587
Undelivered received frames       = 11
Untransmitted frames              = 7
CRC errors received               = 0
Transmit collisions               = 1528
One transmit collision            = 68
More transmit collisions          = 730
Excess retries                    = 0
Deferred transmissions            = 0
Carrier lost when transmitting    = 0
No heartbeat after transmission   = 0
Frame alignment errors            = 0
Late transmit collisions          = 0
Frames lost                       = 0
Unknown protocol                  = 0
Bad control field                 = 0
Illegal sized frames              = 0
Unable to find transmit buffers   = 0
One of zero receive buffers       = 0
IEEE 802.3 XID packets            = 0
IEEEU 802.3 TEST packets          = 1
Unable to respond TESt/XID pkts   = 0
```

Figure D-2. Series 600/800 LAN Interface Status Display

```
                LAN INTERFACE STATUS DISPLAY
                 Fri,Mar 21,1986     08:51:29

Device file                      = /dev/lan0
Lu number                        = 0
Current state                    = active
LAN Interface address, hex       = 0x080009000000
Number of multicast addresses    = 2
Frames received                  = 107983
Frames transmitted               = 113587
Undelivered received frames      = 11
Untransmitted frames             = 7
CRC errors received              = 0
Transmit collisions              = 1528
One transmit collision           = 68
More transmit collisions         = 730
Excess retries                   = 0
Deferred transmissions           = 0
Carrier lost when transmitting   = 0
No heartbeat after transmission  = 0
Frame alignment errors           = 0
Late transmit collisions         = 0
Frames lost                      = 0
Unknown protocol                 = 0
Bad control field                = 0
IEEE 802.3 XID packets           = 0
IEEEU 802.3 TEST packets         = 1
Unable to respond TESt/XID pkts  = 0
```

Figure D-3. Series 700 LAN Interface Status Display

Description of Status Fields

Field	Description
Device file	The name of the LAN interface device file from which the display information is taken. (The device file can be set with the LAN Interface Test Mode *name* command.)

Select code	Series 300/400 only. The location of the LAN interface card, as specified by the minor number field of the device file. (Refer to *mknod(1M)* for further information about minor numbers.)
lu number	Series 600/800 and Series 700 only. The number of the device logical unit associated with a LAN card. The system assigns this number after system bootup.
Current state	The state of the LAN interface card upon the execution of the *display* command. The state indicates the availability of the device for network traffic. The possible states are ACTIVE and FAILED.
Self-test completion code	The result of the device's last self-test. A non-zero code indicates an error. **This value is displayed only if the card has FAILED.** Refer to Appendix E for a list of the self-test completion code values.
LAN Interface Address	The six-byte Ethernet or IEEE 802.3 address of the LAN interface card. (Also called link-level address or network station address.) The address can be found on the NOVRAM chip of the LAN interface card. The value is printed in hexadecimal form.
Number of multicast addresses	The number of accepted multicast addresses.

Description of Statistics Fields

The count values for the following statistics accumulate until the statistics registers are cleared.

Field	Description
Frames received	The number of frames received by the LAN interface card.
Frames transmitted	The number of frames transmitted by the LAN interface card. If you know the date that the statistics registers were last cleared, the number of frames received and the number of frames transmitted since that date, you can estimate the traffic on the network involving your node.
Undelivered received frames	The number of undeliverable frames that the card received. The frames could not be delivered because the software buffer was overrun when frames were sent faster than they could be received.
Untransmitted frames	The total number of frames that the card was unable to transmit due to errors. Errors specific to other statistics are also tallied here.
CRC errors received	The number of frames with a bad CRC code received by the LAN interface card. The CRC, or Cyclic Redundancy Check, is a link-level data integrity check for the entire packet. The normal value is 0. If the value is high in relation to the *Frames Received* statistic, or if you cannot communicate with a particular node, you may have a hardware failure. The failure could be on the receiving or the transmitting computer. To determine which computer has the failure, run the *ping* diagnostic program on one of the computers for approximately 10 seconds. Check the *ping* statistics for packet loss. Recheck the CRC

errors. For further information about hardware troubleshooting, refer to Chapter 6.

Transmit collisions — The number of collisions detected by the LAN interface card during a transmission. This is a general indication of how heavily the network is being used.

One transmit collision — The number of times one retry was needed to transmit a frame. Because a single collision is not a serious occurrence, the normal range is not limited to 0.

More transmit collisions — The number of times the transmission of a frame was completed after 2 to 15 retries. The normal range is not limited to 0, but if it is large, the LAN was heavily used during the time since the statistics were last cleared. If a large value persists for this statistic, try to determine which individual computers are creating heaviest use of the network and whether the use is due to applications running on the computer or due to LAN hardware or software problems.

Excess retries — The number of times the transmission of a frame failed after 15 retries. The normal range is not limited to 0, but if it is large, the LAN was heavily used during the time since the statistics were last cleared. If a large value persists for this statistic, try to determine which individual computers are creating heaviest use of the network and whether the use is due to applications running on the computer or due to LAN hardware or software problems.

Deferred transmissions — The number of times the network was busy when the LAN interface card attempted to transmit. Indicates the amount of traffic on the network.

Carrier lost when transmitting — The number of times the carrier was lost when transmitting a frame. The normal value is 0. If the value is not 0, the LAN interface card can

no longer find the network. Run the *display* function of the *landiag* diagnostic program on another HP 9000 computer. If the remote computer has the same problem, check the LAN cable for possible faults. If the remote computer does not have the same problem, make sure that the AUI cable is correctly plugged into the local computer's LAN interface card and MAU. Make sure that the MAU connection to the LAN cable is correctly installed. This may mean reinstalling the MAU.

No heartbeat after transmission

The number of times no heart beat was indicated after a transmission. The heartbeat is transmitted from the MAU to the LAN interface card to inform the interface card that the MAU is functioning correctly. If you are using an Ethernet compatible MAU and you are receiving this error, it indicates that you are using the wrong card connector cable. If you are using an IEEE 802.3 compatible MAU, it indicates a failure. You may need to replace the MAU, the LAN interface card or the AUI cable.

Frame alignment errors

The number of frames received with both CRC error(s) and alignment error(s). See the discussion on *CRC errors received.* An alignment error means that extra bits have been transmitted with a packet. This is only significant if there is also a CRC error.

Late transmit collisions

The number of transmissions aborted because a collision occurred after the allotted channel time had elapsed. If this value is not 0, you may have too large a network or a repeater that is not working, or you may need to replace your LAN interface card.

Frames lost

The number of times that a frame was missed due to a lack of resources on the interface card. Frames were not received by the hardware because the sender transmitted too fast.

Unknown protocol	The number of frames received with a *sap* field or *type* field that had no associated protocol. The normal value is 0. If the value is not 0, find the address of the computer that sent the packet and determine why it is sending packets to the local computer. You may need a LAN Analyzer to figure out who the remote computer was.
Bad control field	The number of IEEE 802.3 frames received with an illegal control field. The normal value is 0. If the value is not 0, a control field value of other than XID, TEST or UI has been received, or an Ethernet type field in the restricted range was received.
Illegal sized frames	Series 600/700/800 only. The number of time the card received and discarded packets that were illegal in size (greater than 1514 bytes).
Unable to find transmit buffers	Series 600/700/800 only. The number of times that the card exhausted its transmit buffer space.
One or zero receive buffers	Series 600/700/800 only. The number of times the card had one or no buffers to accept incoming packets.
IEEE 802.3 XID packets	Series 600/700/800 only. The number of IEEE 802.3 XID packets that were received.
IEEE 802.3 TEST packets	Series 600/700/800 only. The number of IEEE 802.3 TEST packets that were received.
Unable to respond TEST/XID pkts	Series 600/700/800 only. The number of IEEE 802.3 XID or TEST that were received but not responded to due to lack of resources.

E

LAN Interface Card Self-test Codes

The self-test completion code for the LAN interface card on the Series 300/400 is displayed in decimal form. The completion code is displayed by *landiag* when the LAN interface state is "FAILED."

The list of code meanings is below.

Decimal Value	Meaning
-2	Check the priority on the interface card. It should be set to 5 or 6. If it is set below 5, HP-UX ignores the interface card.
1 - 34	LAN interface card failure.
35	Cable is unterminated at one end or MAU is not securely tapped into the backbone.
36	Cable is unterminated at both ends.
37	AUI cable is not connected to the MAU or the backbone cable is grounded and should not be.
38	A remote computer is trying to transmit to the local computer while the local computer is performing its loopback test.
39 - 42	Link failure.
43	Hardware failure.
44	Hardware failure.

LAN Filesets

To obtain the specific functionalities desired for your HP-UX system, you must select the related include statements (S800) or keywords (S300/S700) and be sure that they are present in the *S800* (S800) or *dfile* (S300/S700) prior to generating the kernel. The table in this appendix shows the correspondence between fileset names and required include statement/keywords to facilitate your selection process when a new kernel is to be generated.

In some cases, a set of include statements or *dfile* keywords are required to link filesets in the kernel; in other cases, filesets that are configurable may require additional include statements or keywords to configure them into the kernel. The fileset, NET, does not require any include statements or keywords.

You can use the table below to help verify the correctness of your *S800* file or your *dfile* prior to generating the kernel. This table does not depict the dependencies between filesets or explain fileset selection procedures. Refer to Chapter 3 for additional information on filesets and procedures to generate a new kernel.

Table F-1. Correspondence Between LAN/9000 Filesets, S800 File Include Statements, and S300/S700 dfile Keywords

Fileset Name	S800 file Include Statement	S300 dfile Keyword	Additional Information
BSDIPC-SOCKET	include uipc;	uipc	required for fileset
NETINET	include inet;	inet	required for fileset
	include nm;	netman	required for Network Management
	include ni;	ni	optional for PPL
NET			
NETIPC	include nipc;	nipc	required for fileset
NETTRACELOG	include netdiag1;	netdiag1	required for fileset
APPLTALK	include atalk;	atalk	required for fileset
LAN	include lan;		required for fileset
	include lan0;		required for CIO cards
	include lan1;		required for NIO cards
		lan01	required for fileset
		lla	required for fileset
		num_lan_cards x	required for 3, 4 or 5 LAN cards; x = number of LAN cards; default is 2; valid range is 1 to 5

Index

G

H

I

K

L

M

N

O

P

R

S

T

U

V