HP 9000 Networking

Installing and Administering LAN/9000 Software

Edition 3
E0792

98194-60530
Printed in U.S.A. 07/92

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 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 "Installing LAN/9000" describes how to install LAN/9000 software.

Chapter 2 "Configuring LAN/9000 Using SAM" describes the steps to configure LAN/9000 software automatically using the System Administration Manager (SAM).

Chapter 3 "Troubleshooting LAN/9000" provides flowcharts to help diagnose LAN/9000 software and hardware problems.

Chapter 4 "Manually Installing and Configuring LAN/9000" describes the steps to build the kernel, create device files, and manually configure LAN/9000 using the vi editor.

Chapter 5 "Configuration Commands" provides description of the *eisa_config(1M)*, *ifconfig(1M)*, *lanconfig(1M)* and *route(1M)* commands.

Chapter 6 "Network Diagnostic Commands" provides descriptions of useful diagnostic utilities. Use these command descriptions as reference information when you are following the troubleshooting procedures in chapter 3.

Chapter 7 "Logging and Tracing Commands" describes the commands that activate the common logging and tracing tool, *nettl*.

Chapter 8	"Product Description" describes product structure and relates software components to the Open Systems Interconnect (OSI) Reference Model.
Chapter 9	"Network Addressing" defines networking terms and explains network interface name and unit, network addresses, and names and subnets.
Chapter 10	"LAN Device and Interface Terminology" defines terms used by the I/O system to identify LAN adapters and device files.
Appendix A	"Installation Error Messages" lists error messages related to loading and configuring LAN/9000 software.
Appendix B	"Diagnostic 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	"LAN Interface Adapter Statistics" describes LAN adapter status values and statistics returned by *landiag(1M).*
Appendix E	"LAN Interface Adapter Self-test Codes" lists error codes returned by a "failed" *landiag(1M)* interface test.
Appendix F	"LAN Filesets" describes the contents of each LAN fileset and the S800 include statements and S300/S700 keywords required for generating a new kernel.
Appendix G	"Network Daemons and Library Routines" provides a quick reference list of the daemons and library routines provided and used by the LAN/9000 product.
Appendix H	"Reconfiguring the S700 Kernel (Standalone)" provides a complete list of the steps required to reconfigure the S700 kernel (standalone).
Appendix I	"Reconfiguring the S300/S400 Kernel (Standalone)" provides a complete list of the steps required to reconfigure the S300/400 kernel (standalone).

Appendix J	"Reconfiguring the S800 Kernel (Standalone)" provides a complete list of the steps required to reconfigure the S800 kernel (standalone).
Appendix K	"Logging and Tracing Subsystems" lists the LAN/9000 logging and tracing subsystems used with the *nettl* utility.
Appendix L	"Sample netlinkrc File" provides an example */etc/netlinkrc* file.
Appendix M	"Related Manuals, Protocols, and Standards" provides lists of related networking and system documentation along with lists of the protocols and standards on which the LAN/ARPA products are based.

Contents

Chapter 4 Manually Configuring LAN/9000

Chapter 8 Product Description

Chapter 9 Network Addressing

Chapter 10 LAN Device and Interface Terminology

Appendix A Installation Error Messages

Figures

Tables

1

Installing LAN/9000

Note If you have a system with LAN/9000 preinstalled on it, you may skip this chapter and go directly to chapter 2. Execute the command, `/etc/lanscan`, to determine if the LAN/9000 software and hardware have been pre-installed.

If, in addition, you have a preconfigured system, you may also skip chapter 2, "Configuring LAN Using SAM." The Series 700 Core I/O 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.

This chapter describes the procedures to load LAN/9000 software and to install LAN hardware onto your system. It contains the following sections:

- Overview of LAN Installation.
- Step 1: Checking LAN Installation Prerequisites.
- Step 2: Loading LAN Software.
- Step 3: Setting the EISA Card Configuration (Series 700 only).
- Step 4: Installing LAN Hardware (Series 300/400 and Series 700 only).

Note If you are unfamiliar with HP LAN products or networking concepts, HP recommends that you read chapter 8, "Product Description," chapter 9, "Network Addressing," and chapter 10, "LAN Device and Interface Terminology" before beginning LAN/9000 installation.

Overview of LAN Installation

Installation of LAN/9000 includes checking installation prerequisites, loading the LAN/9000 filesets using the *update* utility, setting the EISA card configuration (Series 700 only) and installing LAN hardware (Series 300/400 and Series 700 only).

If you have Series 600/800 hardware, this chapter assumes that LAN/9000 hardware has been installed and verified.

On Series 300/400 and Series 700 systems, after adding the LAN hardware card into your system, proceed to chapter 2, "Configuring LAN Using SAM."

Note Prior to installing LAN/9000, HP recommends that you create a network map or update the existing map of your LAN network. Refer to chapter 9 for a sample LAN network map and sample worksheet.

Step 1: Checking LAN Installation Prerequisites

Prior to loading the LAN/9000 product onto your system, check that you have met the following hardware and software prerequisites:

1. The operating system should have been upgraded to HP-UX 9.0 software.

   To obtain this information, execute the command:

   ```
   uname -a
   ```

2. You have a thinLAN cable with a BNC T-connector or an AUI cable to connect to a backbone MAU, thinMAU or ethertwist MAU.

3. **Series 300/400:** The workstation backplane contains an available empty slot for the LAN card.

   Series 700: The system backplane contains an EISA system interface. If you have a S720 model, be sure that you ordered and installed the EISA system interface; if you have a S730 model or S750 model, the EISA system interface is preinstalled. The EISA interface should also contain an empty EISA slot for the LAN card.

   Series 600/800: Not applicable. The hardware should already be installed in the system.

4. You have an IP address, subnet mask (optional), and host name alias for your new LAN card.

5. You have super-user capability.

6. You have the installation and service manual for your network card.

Step 2: Loading LAN Software

Follow the steps below to load LAN/9000 software using the HP-UX *update* program.

1. Insert the software media (tape or disk) into the appropriate drive.

2. Run the *update(1M)* program in interactive mode using the command:

   ```
   /etc/update
   ```

3. Select and load the LAN/9000 filesets.

   If you want to verify the LAN/9000 fileset prior to installation, select `Select/View Partitions and Filesets` from the Main Menu. Refer to appendix F for a description of the contents of each fileset, a listing of the filesets required for each type of networking software, and a table showing the correspondence between the filesets and the include statements/keywords used for kernel generation.

   To load the LAN/9000 filesets, return to the Main Menu and select `Select All Filesets on the Source Media` from the Main Menu and `Start Loading Now...`. Respond `y` to the prompt to rebuild the kernel.

   update (1M) loads the filesets, runs the customized scripts for the filesets, and builds the kernel. Estimated time for processing: 8 to 10 minutes.

   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*.

4. 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.

5. Wait until the system reboots.

   When the kernel build is successful, *update (1M)* automatically reboots the system.

   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.

6. Log in as `root`.

7. **Series 300/400:** Skip the next section on setting the EISA card configuration and go to the following section, "Step 4: Installing LAN Hardware."

   Series 700: Go to the next section, "Step 3: Setting the EISA Card Configuration," and then go to "Step 4: Installing LAN Hardware."

   Series 600/800: As your hardware should already be installed, you may skip the next two sections and go to chapter 2, "Configuring LAN Using SAM."

For additional information on the HP-UX *update* program, refer to *Installing and Updating HP-UX*, which is available in separate volumes for Series 300/400 and Series 700 systems.

Step 3: Setting the EISA Card Configuration

The information in this section applies only to Series 700 EISA LAN/9000. If you are installing Series 300/400 LAN/9000, go to the next section, "Step 4: Installing LAN Hardware."

Follow the steps below to add the LAN card configuration file using the `eisa_config` system configuration tool. (The instructions for installing the card are in Step 4 in the next section.) For more detailed information on the `eisa_config` utility, refer to *Installing Peripherals*.

1. Run the `eisa_config` utility in interactive mode using the command:

   ```
   /etc/eisa_config
   ```

   The screen display shows a slot number, configuration (CFG) file, and contents description for each EISA card.

2. Verify that `eisa_config` displays either !HWPC000 at slot 0 for the S720/730 EISA system board or !HWPC010 at slot 0 for the S750 EISA system board.

Note If you inserted the hardware card into the EISA interface prior to installing the LAN/9000 software using the `update` utility, `eisa_config` may automatically recognize the card. If so, you should proceed to Step 4 below.

---

3. Add the LAN card configuration file to the EEPROM memory chip on the EISA interface using the *add* command, followed by the number of an empty slot for the card, after the EISA prompt. HP recommends that you use the lowest numbered empty slot.

   In the example below, 3 is the slot number of the EISA LAN interface:

   ```
   add !HWP1850.CFG 3
   ```

   The addition of this card can cause previous card configurations to change. This situation should be handled as described in the E/ISA configuration documentation and in conjunction with any specific product installation manuals corresponding to those cards.

4. Execute the *show board* command followed by the slot number of the card to display the board's basic attributes. In the example below, 3 indicates the slot number of the EISA LAN interface.

   ```
   show board 3
   ```

5. To save the new card configuration, execute the command:

   ```
   save
   ```

   No switches or jumpers have to be changed.

6. To leave the `eisa_config` utility, execute the command:

   ```
   quit
   ```

   Upon exiting `eisa_config`, a list of required steps will appear on the screen.

   For the S700 networking cards, steps 1 and 2 on the list do not apply. As the necessary device file will be created later by SAM or, if you decide to configure the product manually, when you follow the instructions in chapter 4, step 1 is not necessary. As the LAN device drivers were added to the kernel following the update process, step 2 is also not necessary. Detailed hardware installation instructions for steps 3 through 6 are included in the next section.

7. Go to the next section, "Step 4: Installing LAN Hardware."

Step 4: Installing LAN Hardware

Note If you installed the card before you installed the software, execute `shutdown` (no `-h` option) and proceed to chapter 2.

Follow the steps below to prepare the system for installation of LAN hardware.

1. At the HP-UX prompt, execute the command:

   ```
   /etc/shutdown  -h
   ```

   Wait for the system to respond with a message indicating that the system has been halted.

2. Observe antistatic precautions by following the guidelines as described in the installation instructions in the hardware manual or the *Antistatic Precautions Note.*

3. **Series 300/400:** Install the card following the instructions in the hardware installation and service manual.

   Series 700: If you have a S720/S730 model, remove the EISA system interface and insert the EISA LAN card into the card guide on the EISA interface. If you have a S750 model, the EISA interface is not removable. Insert the card directly into the lowest empty slot in the EISA interface.

4. Attach the thinLAN cable to the LAN using a BNC connector or, if you are using an AUI cable, to the LAN using a backbone MAU, thinMAU, or ethertwist MAU. Refer to figure 1-1 for details..

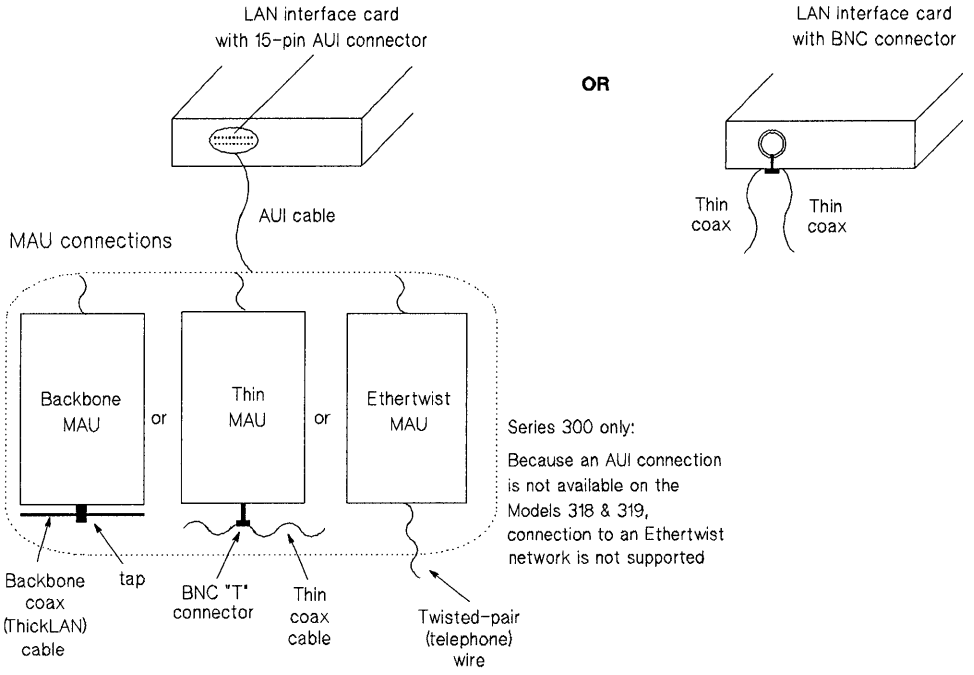


Figure 1-1. LAN/9000 Connections

5. Power up the system to complete the process. The LAN card will run a self-test automatically. Any error messages will appear on the terminal display or system console.

6. When the system is up and running, log in as `root` and execute the */etc/dmesg* command to verify the status of the LAN card. The following messages should appear:

   ```
   "LAN(X) STATUS: Opening Card....Opened Successfully"
   ```

7. Proceed to chapter 2, "Configuring LAN Using SAM."

2

Configuring LAN/9000 Using SAM

This chapter describes how to configure LAN using SAM. It contains the following sections:

- Overview of Configuration Using SAM
- Step 1: Configuring the LAN Link
- Step 2: Configuring Network Connectivity
- Step 3: Verifying the Installation

Overview of Configuration Using SAM

Once you have installed LAN hardware and 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. SAM has two user interfaces, an X-Windows system interface and a text terminal interface. The primary components and functionality of SAM are the same for both interfaces. The differences are the screen appearance and the navigation methods.

You can access the SAM online help system using the following methods:

- Choose an item from the "Help" menu (located in the menubar) for information about the current SAM screen, keyboard navigation within SAM, and the version of SAM on your system.
- Activate the **HELP** button from a dialog or message box for information about the attributes and tasks you can perform from the currently displayed window.
- Press the **F1** key for context-sensitive information for the object at the location of the cursor.

Using SAM, configuring LAN/9000 can be divided into two procedures:

- Configuring the LAN Link.
- Configuring Network Connectivity.

Follow Step 1 to add the IP address, any alias names, and, if the LAN card is on a subnetwork, the subnet mask for your LAN card. This procedure will automatically initialize the LAN link and attach your node to the local area network (LAN). Follow Step 2 to add remote system names and remote system IP addresses for network connectivity, and also to specify default gateway information.

Note Using SAM is the preferred method for LAN/9000 configuration. However, SAM currently does not support the domain name format. The domain name format 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 section, "Configuring LAN Software Manually" in chapter 4.

Step 1: Configuring the LAN Link

Note Make sure the LAN card and driver are installed in the system before you use SAM to configure the software.

Log in as `root` and do the following:

1. At the HP-UX prompt, type: `sam`
2. Highlight *Networking/Communications* at the SAM main window as shown in figure 2-1.
3. Activate the **OPEN** button.

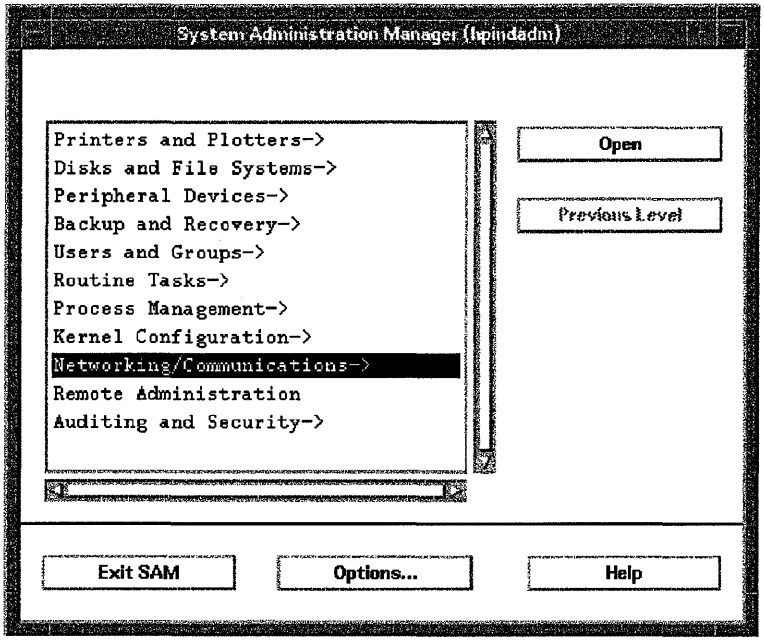


Figure 2-1. SAM Main Window (Series 800)

4. Highlight *Network Card Configuration* at the Networking/Communications window as shown in figure 2-2.

5. Activate the **OPEN** button.

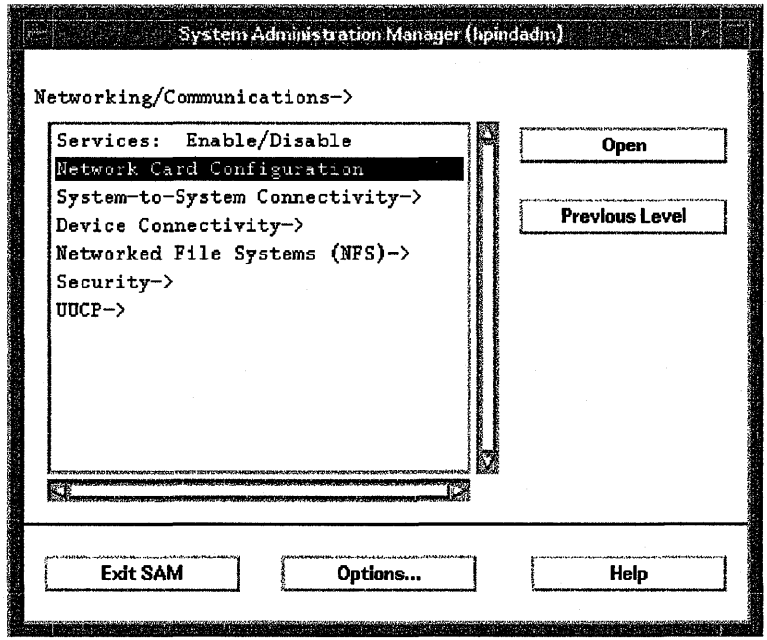


Figure 2-2. Networking/Communications Window

6. Highlight the LAN interface card that you want to configure from the object list as shown in figure 2-3. If the card is not displayed, then go to chapter 1 and check that your hardware has been properly installed.

   Series 300/400 and 700: The SAM object list *always* shows the name of the built-in LAN (IEEE802.3/Ethernet) as *lan0*, and the first LAN card in an EISA slot as *lan1*, whether it is Ethernet, Token Ring, or FDDI. The LAN cards installed in other slots are named sequentially (*lan2, lan3*, etc.), according to the order of the occupied slots.

   Series 600/800: If the system includes the built-in LAN option, it is *lan0*. If the system does not contain a built-in LAN and your LAN card is the first LAN card added to the system, it is *lan0*.

   In this example, Ethernet LAN card in slot 1 is *lan0* (this card is already configured). The slot number does not necessarily correspond to the LAN card's name.

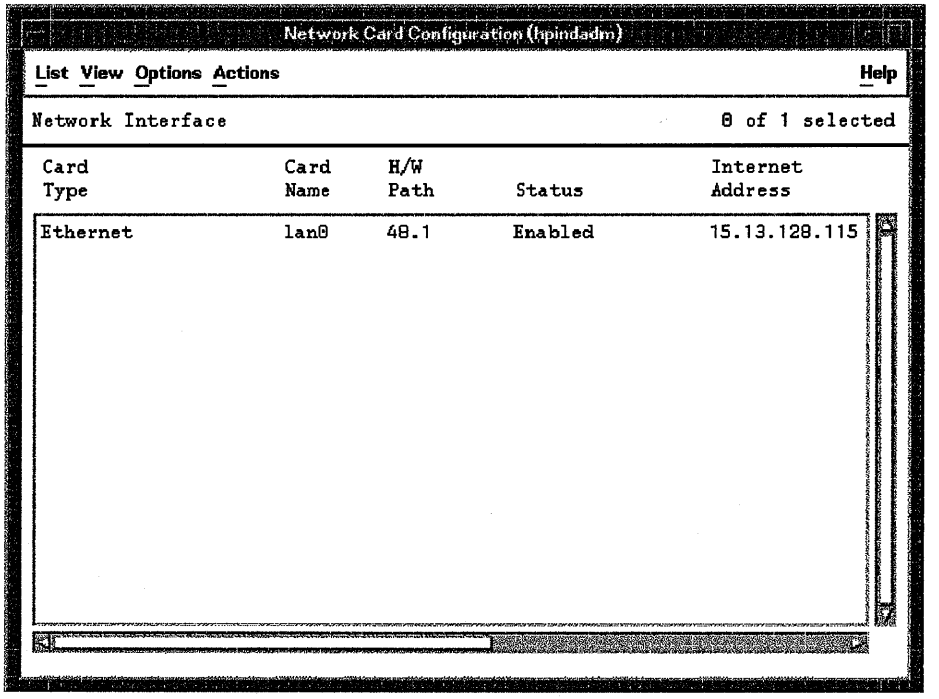


Figure 2-3. Network Card Configuration Object List

7. Verify that the hardware path is correct for your LAN card.

   Series 700: The middle number of the hardware path should match the backplane slot number of the LAN card. For example, if the hardware path is 4.3.0, then the LAN card should be in slot 3.

   Series 600/800: The second number of the hardware path should match the CIO LAN card module slot number of a CIO LAN card. For example, if the hardware path is 4.3, then the CIO LAN card should be in slot 3.

   The hardware path of an HP-PB LAN card should equal 4 times the hardware module number in which the card has been installed. For example, if the hardware path is 32, then the HP-PB LAN card should be in hardware module 8.

8. Choose `Configure` from the "Actions" menu to open the Configure LAN Card window as shown in figure 2-4.

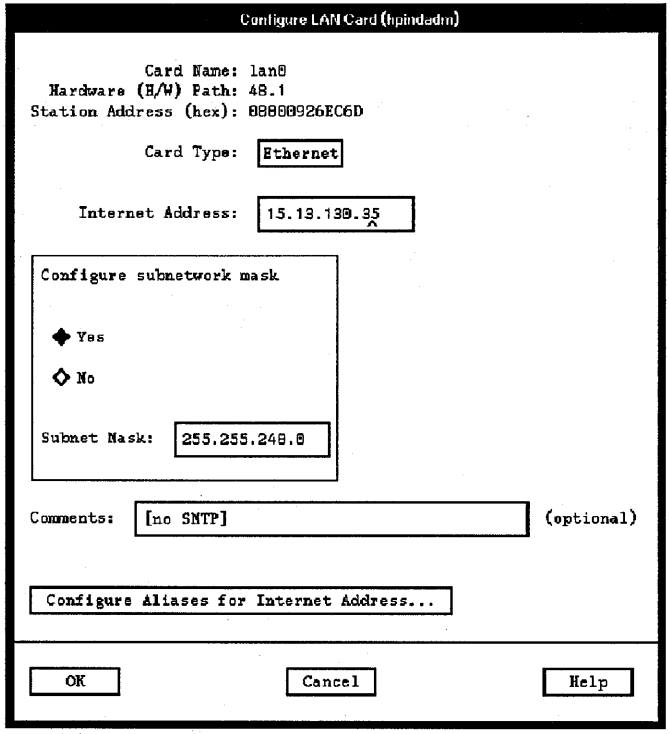


Figure 2-4. Configure LAN Card Window

a. Enter the information about the LAN card. To do so, press the Tab key to move through the data entry fields.

Note SAM displays the Card Name, Hardware (H/W) Path, and Station Address fields with the appropriate values. These fields cannot be modified.

---

b. Choose the type of your LAN card. The default is IEEE802.3/Ethernet.

c. Enter the Internet address for your LAN card.

   Upon exiting the Internet Address field, SAM checks to make sure that the IP address you entered is correctly formatted and is not currently in use.

d. Specify whether your LAN card will be on a subnetwork.

   If you choose **YES**, enter the subnet mask for your subnetwork.

e. Optionally, enter comments about your LAN card.

f. Choose `Configure Aliases for Internet Address` to open the Configure Aliases window as shown in figure 2-5. You must complete this step if you have more than one LAN card installed in your system.

g. Add, modify, or remove alias names for your LAN card.

h. Activate the **OK** button to perform the task and return to the Configure LAN Card window.

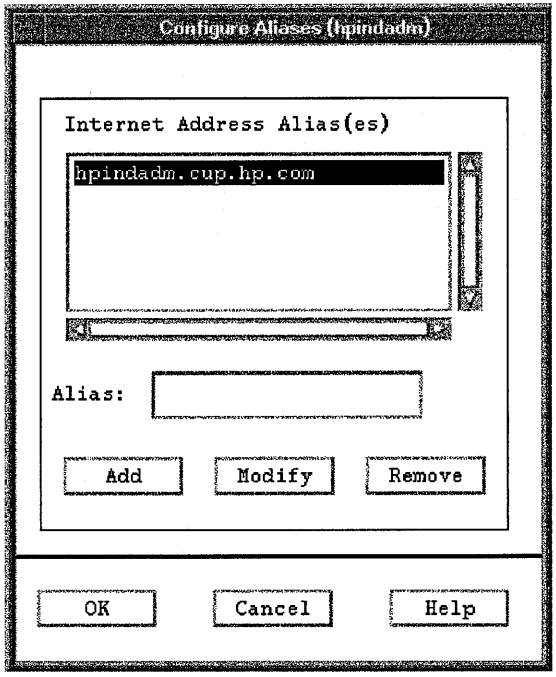


Figure 2-5. Configure Aliases Window

9. Activate the **OK** button at the Configure LAN Card window to enable your LAN card.

   If the software is correctly configured, SAM displays the Network Card Configuration object list with the status **Enabled** for your LAN card; otherwise, SAM displays an error message.

10. Choose `Exit` from the "List" menu. SAM will display the Create a New Kernel window.

    If you have completed your configuration, choose the option to create a new kernel now.

    If you want to configure network connectivity, choose the option to defer kernel generation until later.

11. At the Networking/Communications window, activate the **Exit SAM** button.

    If you have moved or removed any LAN cards from the system, HP recommends that you verify the IP address of every card in the backplane before leaving SAM.

12. Verify that the LAN/9000 device files have been created correctly by executing the HP-UX commands:

```
cd /dev
ls -l lan* ieee* ether*
```

    Series 300/400: After running SAM to configure LAN software, you should verify that SAM has found the select code and device logical unit number (lu) of each LAN card and has created three device files: */dev/lan*X, */dev/ieee*X, and */dev/ether*X (where X is the lu of the card).

    The fifth column in the display is the major number. This number should be 18 for DIO LAN drivers using the IEEE 802.3 protocol and 19 for DIO LAN drivers using the Ethernet protocol. The sixth column is the minor number. The first two-digit field in the minor number is the select code and should be 21 or 15 hex. The other two, two-digit fields should always be 0.

    Series 700: After running SAM to configure LAN software, you should verify that SAM has found the lu of each add-on LAN card and has created three device files: */dev/lan*X, */dev/ieee*X, and */dev/ether*X (where X is the lu of the interface). The device files for the built-in LAN are created during system boot-up.

    The major number for EISA LAN cards should be 52. The slot number of the card should correspond to the fifth most significant digit in the minor number. For example, if your card is installed in slot three of an S750 system, the minor number is 0x430000. The sixth most significant digit (4 in this example) means EISA.

Series 600/800: Series 800 device files are created and bound to the I/O subsystem during system boot-up.

The major number for CIO LAN cards should be 50, the major number for 815 and 8x2 HP-PB LAN cards should be 51, and the major number for 8x7 HP-PB LAN cards should be 32. The minor number is 0x00*nn*00 where *nn* is the byte for the manager class index. The first card has a manager class index of 0. If additional cards are added to the system, index values will be assigned in numerical sequence as the cards are installed on the system.

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.

If you want to configure your system for network connectivity, continue with the next section, "Step 2: Configuring Network Connectivity." If not, continue to "Step 3: Verifying the Installation."

Step 2: Configuring Network Connectivity

Your system may not be able to communicate with other systems (for example, PCs, workstations, servers, etc.) until you configure system-to-system connections. You can use SAM to do this automatically by completing the following steps:

1. At the HP-UX prompt, type: `sam`
2. Highlight *Networking/Communications* at the SAM main window as shown in figure 2-6.
3. Activate the **OPEN** button.

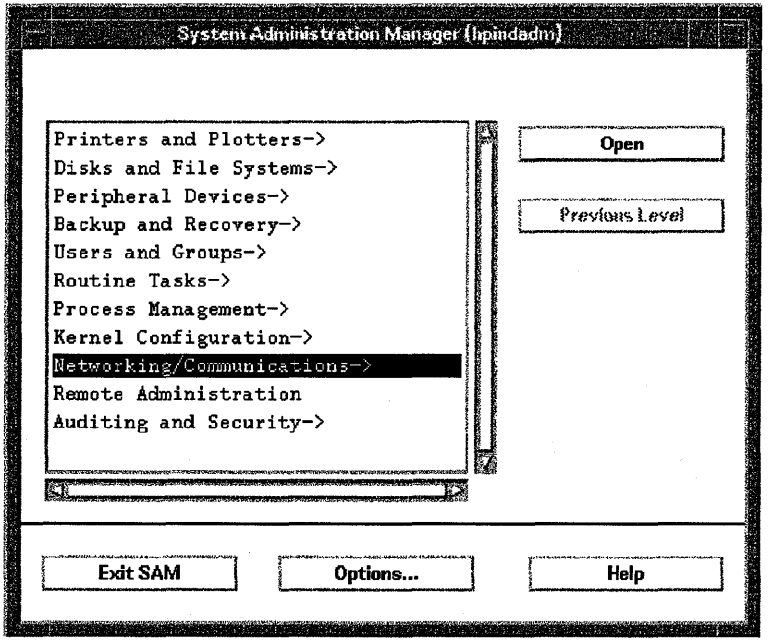


Figure 2-6. SAM Main Window (Series 800)

4. Highlight *System-to-System Connectivity* at the Networking/Communications window as shown in figure 2-7.

5. Activate the **OPEN** button.

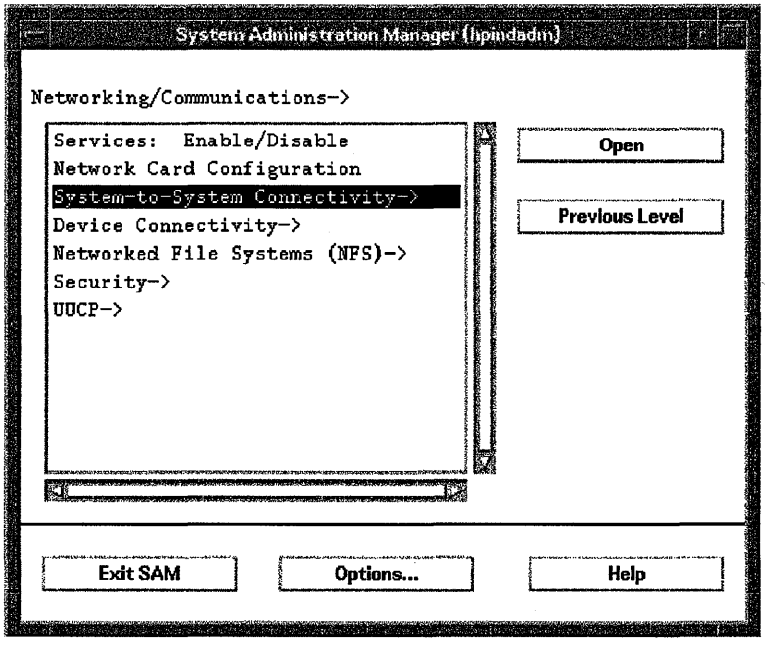


Figure 2-7. Networking/Communications Window

6. Highlight *Internet Connectivity* to enable your system to communicate with other systems using the TCP/IP protocol.

7. Activate the **OPEN** button.

   SAM displays the remote system names and Internet addresses that are already configured as shown in figure 2-8.

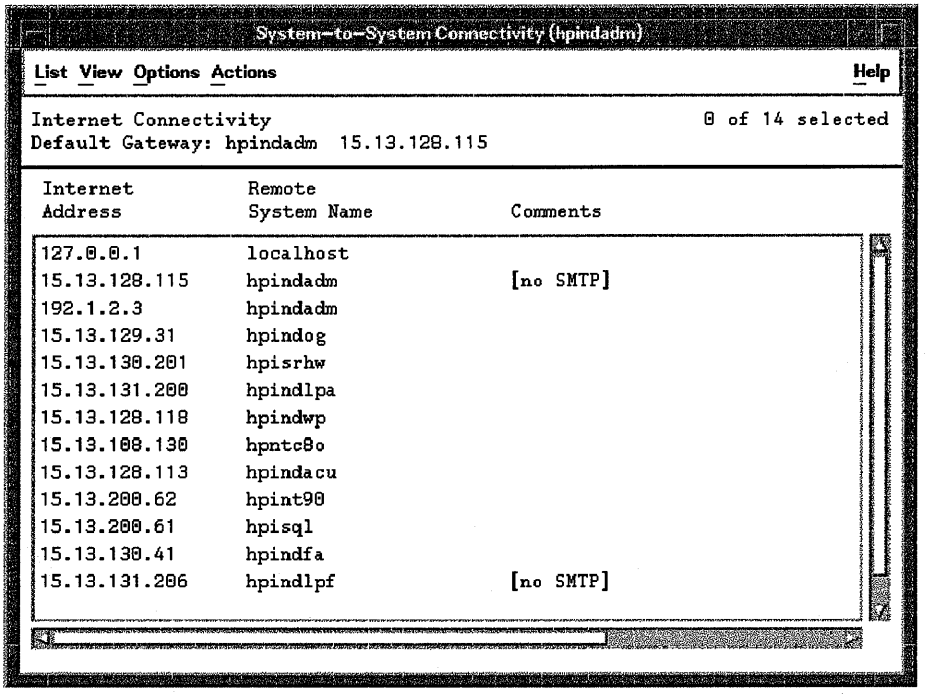


Figure 2-8. System-to-System Connectivity Object List

8. Choose Add from the "Actions" menu to open the Add Internet Connectivity window as shown in figure 2-9.

   Use the SAM online help system for information about adding remote system connections.

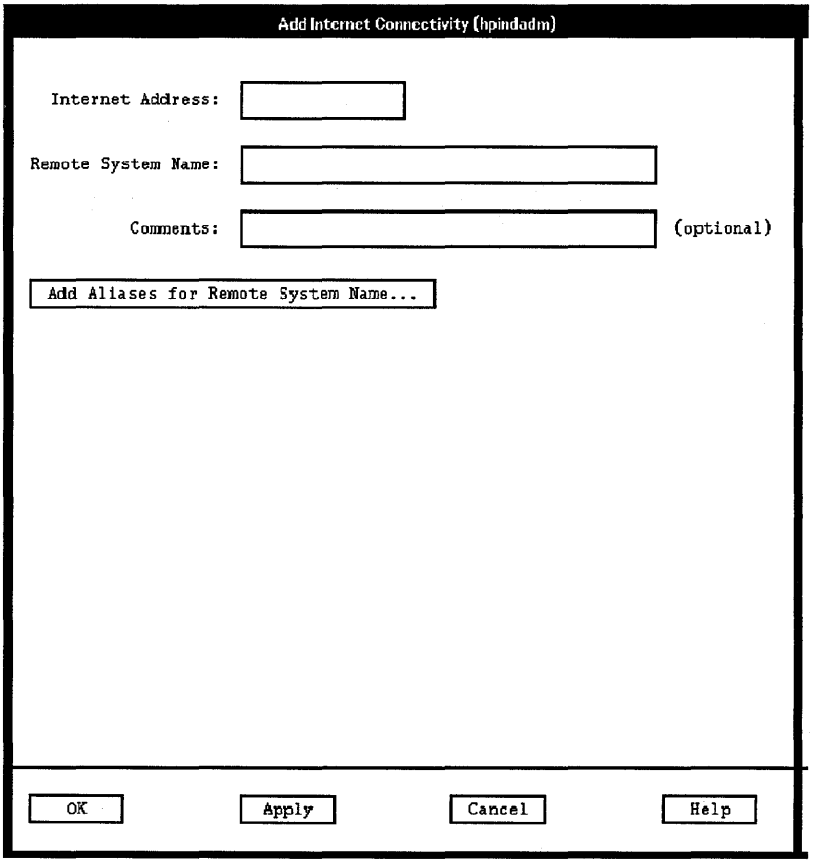


Figure 2-9. Add Internet Connectivity Window

a. Enter the Internet address for the remote system.

Upon exiting the `Internet Address` field, SAM checks to make sure you have entered a valid IP address. SAM also determines if a gateway is required for the connection (see step 8f).

b. Enter the remote system name.

Upon exiting the `Remote System Name` field, SAM checks to make sure that connectivity has not already been configured for this system. If it has, SAM displays an error message.

c. Optionally, choose `Add Aliases for Remote System Name` to open the Aliases for Remote System Name window as shown in figure 2-10.

d. Add, modify, or remove alias names for the remote system.

e. Activate the **OK** button to perform the task and return to the Add Internet Connectivity window.

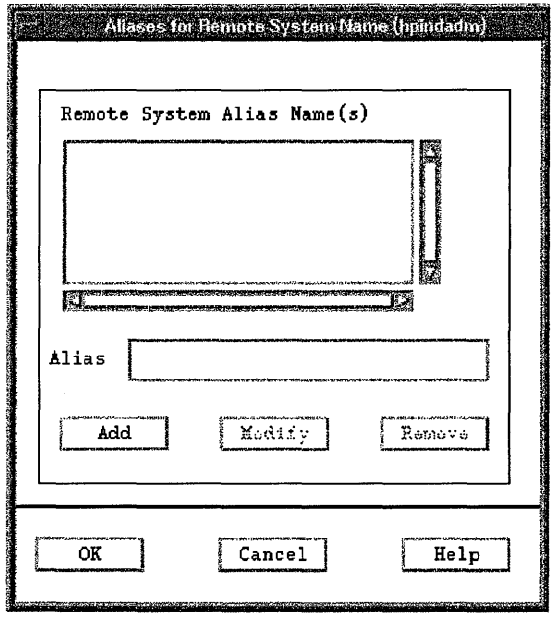


Figure 2-10. Aliases for Remote System Name Window

f. Proceed to step 9 if a gateway is not required for this remote connection.

   SAM displays fields for entering gateway information if a gateway is required for this remote system connection. Use the SAM online help system for information about gateways.

9. Activate the **OK** button to enable your system to communicate with this system and return to the System-to-System Connectivity object list.

   SAM updates the object list to include the remote system you configured.

Note You can modify or remove remote systems and modify default gateways by highlighting the Remote System Name from the object list and choosing `Modify`, `Remove`, or `Modify Default Gateway` from the "Actions" menu.

---

10. Choose `Exit` from the "List" menu. SAM will display the Create a New Kernel window.

    Now that you have completed your configuration, choose the option to create a new kernel now.

11. At the System-to-System Connectivity window, activate the **Exit SAM** button.

12. Verify remote system configuration.

    a. View the list of remote systems you can communicate with using a symbolic name by typing the following command at the HP-UX prompt:

       ```
       more /etc/hosts
       ```

    b. View the configured destinations reached through gateways and the gateways used to reach those destinations by typing the following command at the HP-UX prompt:

       ```
       netstat -r
       ```

To verify that you can communicate with a remote system via the LAN/9000 product, continue to "Step 3: Verifying the Installation."

Step 3: Verifying the Installation

Once your LAN/9000 software is installed, fully configured and running, you should execute the *ver_link* script and run the following commands to verify LAN hardware and software installation. Refer to chapter 6 for complete descriptions of the commands listed below.

1. To verify the installation, run the LAN/9000 verification script:

   ```
   /usr/nettest/ver_link
   ```

   For information about the checks and tests run by this script, refer to the next section, "Running the LAN Verification Script."

2. To check that the your system can communicate with other systems, enter the *ping* command at the HP-UX prompt. In this example, *191.2.1.2* is the IP address of the remote system. Enter **[CNTRL]-C** to stop *ping*.

   ```
   ping 191.2.1.2
   ```

LAN/9000 installation is verified if both *ver_link* and *ping* succeed. You can further verify the link by running the verifiers and diagnostics described in chapter 6.

Running the LAN Verification Script

LAN/9000 software provides a script, */usr/nettest/ver_link*, for verification of all LAN devices on the system. You should run this script after configuring LAN/9000 and after installation of additional LAN cards. It may also be helpful to run this verification script when you encounter problems with LAN.

This script will perform the following verification tests:

- Check that the backplane contains the supported number of LAN cards.
- Check the state of all LAN hardware and interfaces.
- Check for the existence of device files.
- Test for link level loopback 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]
```

Parameters

-h *hostname* is an optional parameter used to test network level connectivity to other Unix Internet machines. *hostname* specifies the name of the local host (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 HP-UX computers that have installed NetIPC and the rlbdaemon. *nodename* specifies the name of the local node (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, either a warning or error message will appear on your terminal screen. Take note of the message and follow the recommended corrective action.

3

Troubleshooting LAN/9000

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

- Troubleshooting Overview.
- Addressing Do's and Don'ts.
- Troubleshooting Q and A.
- Diagnostic Flowcharts.
- Contacting your HP Representative.

Troubleshooting Overview

Troubleshooting LAN problems can be difficult because a variety of hardware and software components may be involved and because the problem impacting your system may originate in another part of the network.

The first two sections in this chapter provide quick fix solutions to common network problems. "Addressing Do's and Don'ts" identifies common addressing errors made when new systems are added to the network. "Troubleshooting Q and A" provides answers to the most frequently asked troubleshooting questions. Look through the problems identified in these sections to see if your problem fits into any of these categories. If so, you may be able to quickly identify and recover from the problem without any further investigation.

If, however, after looking through the following sections on network addressing errors and frequently asked troubleshooting questions, you are unable to identify your problem, proceed to the troubleshooting flowcharts. The troubleshooting flowcharts provide a logical sequence of steps to follow when troubleshooting LAN/9000. Using the diagnostic flowcharts provided in this chapter, identify whether the problem is with LAN/9000 or any of the connections to MAU, or whether it is in some other part of the LAN network, verify your assumptions and, it if is limited to LAN/9000 software and hardware, correct the problem.

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

Chapter 6 – Network Diagnostic Commands
Appendix A – Installation Error Messages
Appendix B – Diagnostic 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)

Addressing Do's and Don'ts

Figure 3-1 summarizes common network addressing errors. When setting up new systems, be especially careful, when you copy files from one system to another, that you don't reboot the system before you add in a new IP address and host name for the new system. Duplicate address entries will cause connectivity problems in the network.

The two telescopic views of a network map with incorrectly assigned addresses are taken from the large network map in figure 9-9. Refer to chapter 9, "Network Addressing" for detailed descriptions of network and subnet addressing rules.

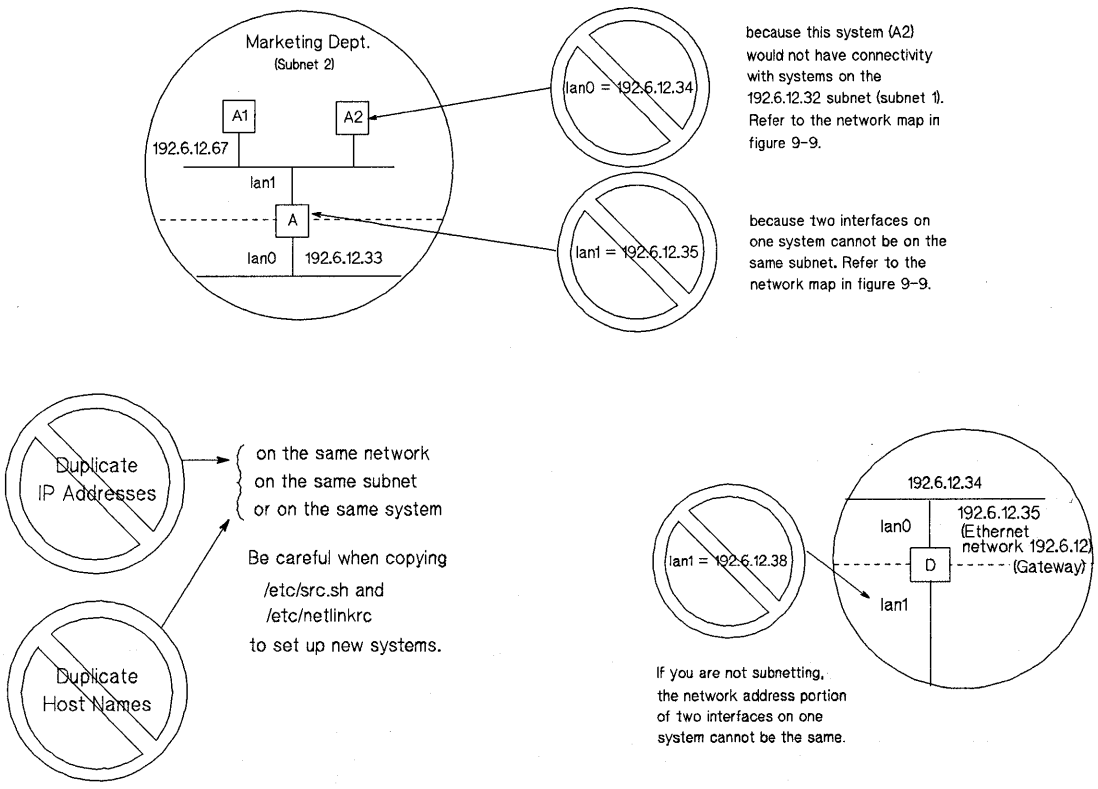


Figure 3-1. Network Addressing Do's and Dont's

Troubleshooting Q & A

Question 1: I tried to attach a new system to our site LAN. To make the installation process faster, I copied over an */etc/src.sh* and */etc/netlinkrc* file from another system on the site LAN and used it on the new system. When I booted up the system, the site LAN went down.

Answer 1: You probably didn't assign a new IP address and host name prior to rebooting the system. If any two systems on the LAN have the same IP address and host name, the LAN will go down. Check the IP address in the */etc/src.sh* and *ifconfig(1M)* entry in the */etc/netlinkrc* file against the IP address of your system and other systems on your network map to be sure that no duplicate IP addresses exist on the LAN.

Related Documentation: Refer to the *ifconfig(1M)* manual page in chapter 5.

Question 2: I have been having problems getting the two LAN interfaces on my system to operate at the same time. Occasionally the ethernet cards stop communicating with remote systems. When this happens, the remote system also cannot communicate with the local system.

Answer 2: Check that the two interfaces on your system do not have the same network number or, if you are subnetting, the same subnet address. If both LAN interfaces have the same value in the network (subnet) address portions of the IP address, the cards may not be enabled simultaneously (although they may both run separately.)

Related Documentation: Refer to chapter 9, "Network Addressing".

Question 3: I recently tried to add a new system onto a subnet on our site LAN, and I am not able to communicate successfully with all LANs on the network.

Answer 3: Check that the subnet address of your system is correct for the subnet to which your system is attached.

Related Documentation: Refer to "Assigning Subnet Addresses" in chapter 9.

Question 4: I tried to add the IP address, 127.0.0.1, and the system won't accept it.

Answer 4: Addresses with the format 127.n.n.n are reserved as loopback addresses. Select another IP address. 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.

Related Documentation: Refer to "Assigning an Internet Address" in chapter 9.

Question 5: I added my EISA card into my Series 700 workstation prior to installing and configuring the software. When I looked at the instructions in chapter 1, I discovered that I should have added the hardware after the software. How can I recover from this situation?

Answer 5: Reboot the system **twice** after you have loaded the LAN software and then go directly to chapter 2 in this manual to configure the card.

Related Documentation: Refer to the *eisa_config(1M)* manual page in chapter 5.

Question 6: How do I locate the station address of my LAN card?

Answer 6: Use the *lanscan(1M)* command to display the station addresses of all LAN cards in the system:

```
/etc/lanscan
```

Related Documentation: Refer to the *lanscan(1M)* manual page in chapter 6.

Question 7: How do I reset the LAN card?

Answer 7: Run the *landiag(1M)* diagnostic by entering the following sequence of commands:

```
/etc/landiag
lan
reset
```

Related Documentation: Refer to the *landiag(1M)* manual page in chapter 6.

Question 8: What's the best way to obtain and format tracing information when I am using the *nettl(1M)* utility?

Answer 8: The HP field engineers recommend the following commands:

To begin routine LAN tracing, execute:

```
nettl  -tn  pduin  pduout   -e all  -f raw0
```

To end routine LAN tracing, execute:

```
nettl  -tf  -e all
```

To format your entire LAN trace (no filtering), execute:

```
netfmt  -N  -f  raw0.TRC0  >  fmt0
```

To format your LAN trace using a filter file, execute:

```
netfmt  -c filterfile  -N  -f  raw0.TRC0 > fmt0
```

nettl(1M) appends *.TRC0* or *.TRC1* to the name you give the raw trace file.

Related Documentation: Refer to chapter 7, "Logging and Tracing Commands."

Question 9: *nettl(1M)* will not log kernel messages. Error messages are appearing during *nettl(1M)* startup.

Answer 9: The *netdiag1* driver may not have been in the *dfile* (or *S800* file) when the kernel was built. To allow network tracing, add *netdiag1* to your *dfile* (or *S800* file), regenerate the kernel, and reboot according to the following steps:

1. Configure the kernel with *netdiag1*.
2. Run the command:

```
ksh /system/NETTRACELOG/customize HP-MC68020 #(S300/400)
or
ksh /system/NETTRACELOG/customize HP-PA #(S600/700/800)
```

3. Reboot.

Related Documentation: Refer to chapter 7, "Logging and Tracing Commands".

Question 10: I tried to use the *dscopy* command between my HP-UX system and an HP 3000 system and I can't get it to copy a file.

Answer 10: The encapsulation method on your HP-UX system may be set to ethernet only (the default is ethernet only). Use the *lanconfig(1M)* command to reset the encapsulation method to `ether ieee`. In addition, be sure to change the *lanconfig(1M)* setting in */etc/netlinkrc* so that the change is permanent.

Related Documentation: Refer to the *lanconfig(1M)* manual page in chapter 5.

Question 11: I think that I have configured my diskless nodes correctly, but they still aren't booting up.

Answer 11: Diskless nodes must be configured on the server as well as on the clients and have the IEEE 802.3 mode "on" in order to boot up. (The default is ethernet only.) Use the *lanconfig(1M)* command to reset the encapsulation method to `ether ieee`. In addition, be sure to change the *lanconfig(1M)* setting in */etc/netlinkrc* so that the change is permanent.

Related Documentation: Refer to the *lanconfig(1M)* manual page in chapter 5.

Question 12: I'm experiencing intermittent networking problems on my computer. What should I check to ensure proper operation of my networking software?

Answer 12: Upper layer software often requires loopback. Check */etc/netlinkrc* to be sure that loopback is enabled. The line in */etc/netlinkrc* should read:

```
/etc/ifconfig  127.0.0.1  lo0
```

Related Documentation: Refer to chapter 9, "Network Addressing".

Question 13: I've noticed a significant drop in system response time and performance. What steps can I take to improve it?

Answer 13: Performance may be affected by many different factors. Sometimes removing pseudo drivers from the kernel for networking software that you may not be using improves performance. The problems may also be in the upper layer software (*ftp* or *telnet*).

Also, during stressful networking activity resulting in packet fragmentation at the IP layer, a disproportionate amount of memory may get fragmented into small chunks and thus become unavailable to the user processes. For this particular problem, you should retune (increase) the value of the *netnemmax* parameter in the *dfile* or (*S800*) file and reboot the system. Use the following description of *netmemmax* to help determine the parameter value.

netmemmax can have the following possible values:

0	(Default) System uses a maximum of 10% of dynamically mallocable memory to reassemble packet fragments.
-1	No limit is placed on the amount of memory the system can use for packet fragment reassembly. (Not a recommended option for this problem.)
X	Allows the user to specify that X bytes of memory be used for packet fragment reassembly. The system rounds *X* to the nearest number of pages and uses this as the user requested threshold.

Question 14: Why is there a significant increase in the number of deferred transmissions and collisions on my network?

Answer 14: On IEEE802.3/Ethernet networks, a collision occurs when two or more stations try to transmit data simultaneously. A deferred transmission occurs if the network is busy when a station attempts to transmit data. The number of collisions and deferred transmissions on a node is directly related to the network load. As the network load increases, the number of collisions and deferred transmissions also increase.

When high-performance systems, such as HP9000 Series 700s, are placed on a LAN with lower-performance systems (HP or non-HP systems), it is possible for the high-performance systems to use a higher percentage of the LAN bandwidth with network traffic intensive applications. High-performance systems generate network traffic at a 10Mbits/s link rate, and lower-performance systems cannot match this rate. Heavily loaded LAN networks can result in lower throughput performance on lower-performance systems.

In general, the short term average load on an IEEE802.3/Ethernet LAN should not exceed more than 70% of the total bandwidth of the LAN. When it does exceed 70% of the total bandwidth, network performance begins to degrade due to an increase in collisions and deferred transmissions. When it consistently exceeds 70% of the total bandwidth, you may need to reconfigure the systems on your LAN. If you notice throughput/performance degradation on your system, contact your local HP Representative for additional assistance and consultation.

Diagnostic Flowcharts

Below is a summary of the types of network tests in the diagnostic flowcharts. To diagnose your problem, first check the connections and configuration on your system (Flowcharts 1 through 5). If this does not solve your problem, use flowcharts 6 through 16 to test/verify connectivity with a remote system.

Table 3-1. 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 & 14	Gateway Remote Loopback Test
15	Probe Proxy Server Test
16	Subnet Test

Configuration Test: Verifies the configuration of the network interface on a host using the *lanscan(1M),* and *ifconfig(1M)* commands.

Network Level Loopback Test: Checks roundtrip communication between Network Layers on the source and target host using the *ping(1M)* diagnostic.

Transport Level Loopback Test: Checks roundtrip communication between Transport Layers on the source and target host using ARPA services *telnet* and *ftp* commands.

Link Level Loopback Test: Checks roundtrip communication between Link Levels on the source and target host using the *linkloop(1M)* diagnostic.

LAN Card Test: Checks the LAN card hardware.

LAN Connections Test: Checks the connections between the LAN media and any component between the LAN media and individual LAN cards.

Gateway Remote Loopback Test: Checks general network connections through a gateway.

Probe Proxy Server Test: Checks problems that may occur with the Probe Proxy Server.

Subnet Test: Verifies the correct use of subnets.

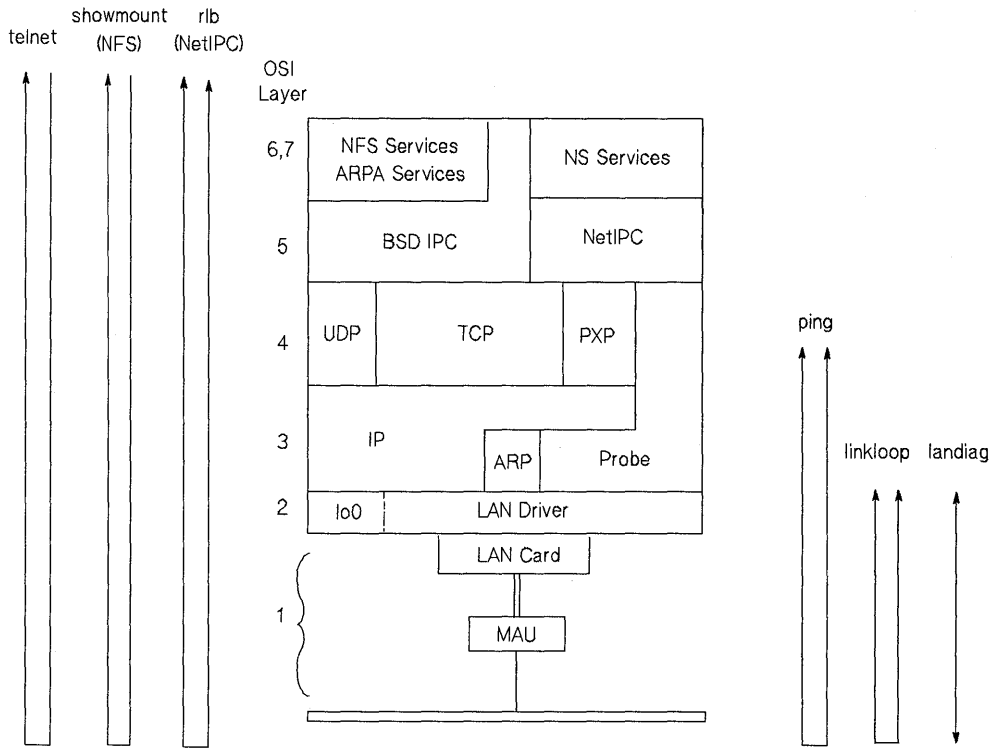


Figure 3-2. Loopback Tests

The loopback tests shown in figure 3-2 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, *ping(1M)*, 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.

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 on the remote system.

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, *linkloop(1M)*, to isolate a problem to OSI Layer 2 or below.

netstat(1M) reports network and protocol statistics regarding traffic and the local LAN interface. As shown in figure 3-3, there are many options to *netstat(1M)*. The options that are most useful are those which display information that is not available through other commands such as *ping* and *landiag*, for example, *-a* and *-r* options. You can also use the *landiag(1M)* command to obtain LAN driver statistics. For more detailed information on these diagnostics, refer to the *netstat(1M)* and *landiag(1M)* manual pages in chapter 5.

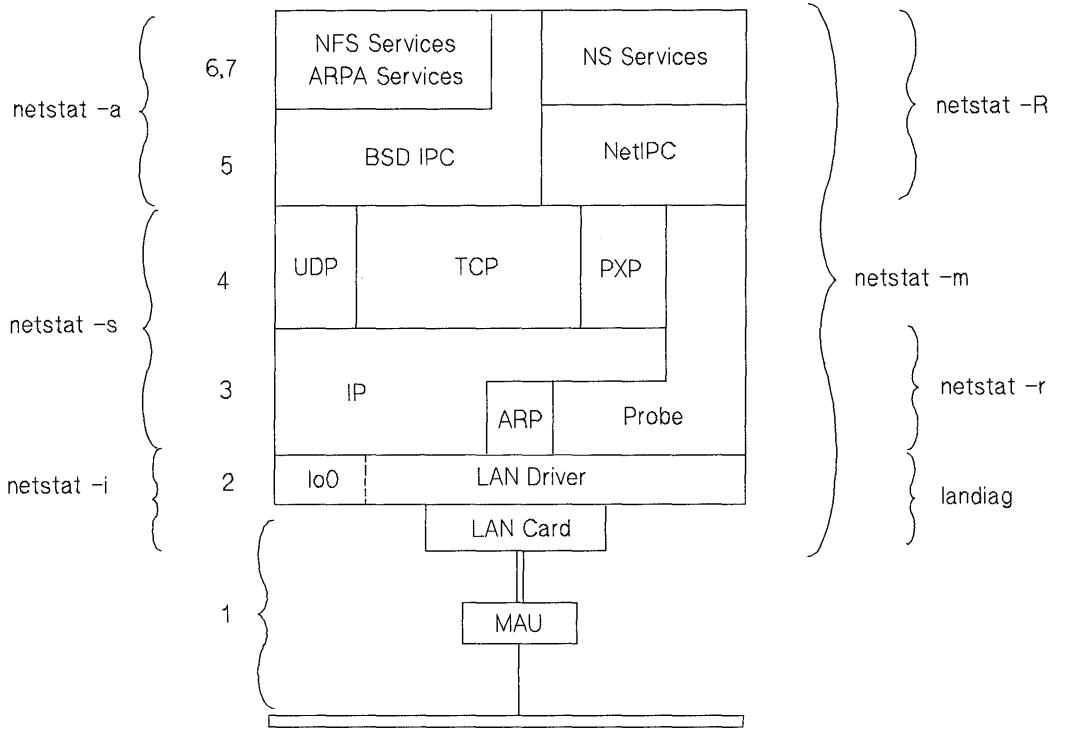


Figure 3-3. Networking Software Statistics

Flowchart 1: Configuration Test

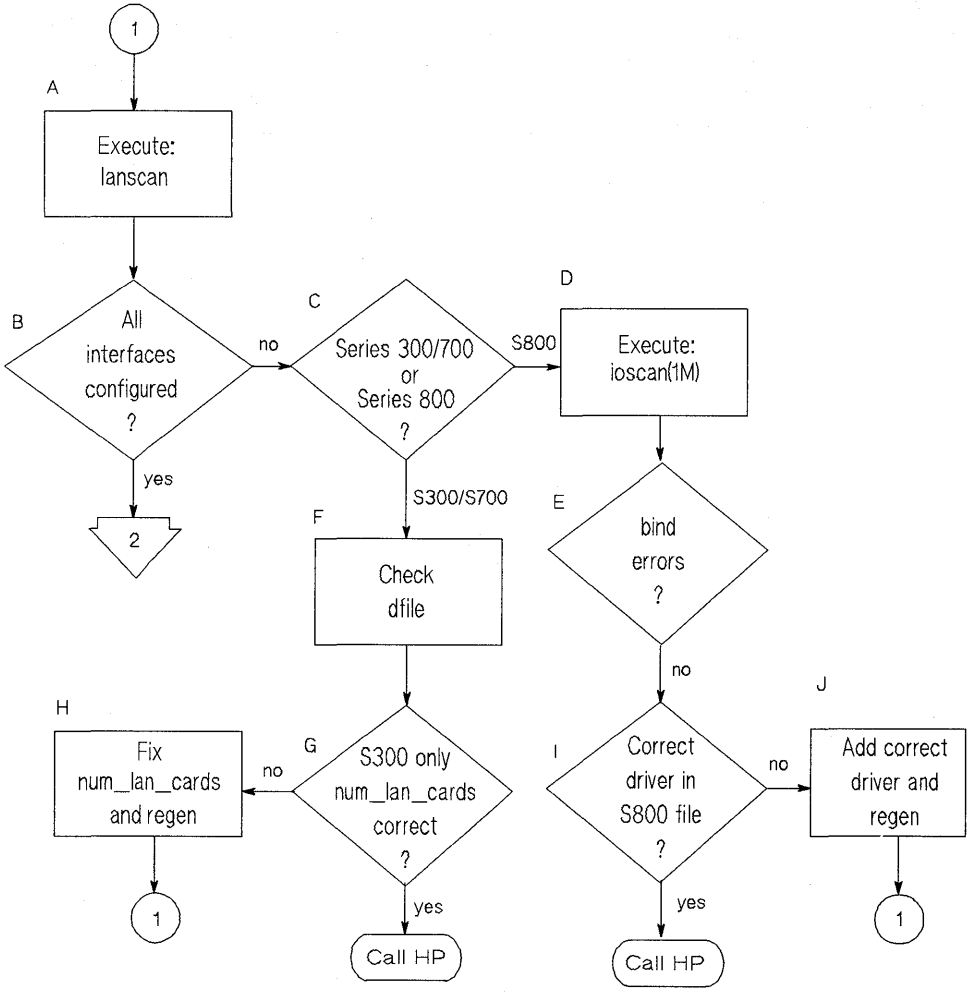


Figure 3-4. Flowchart 1

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
```

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, go to H.

H. **Fix num_lan_cards and regen.** Correct the value in num_lan_cards to reflect the number of cards in the back of the system.

I. **Correct driver in S800 file?** Refer to "Creating a New Kernel for the Series 600/800" in chapter 4 for a list of Series 600/800 LAN drivers.

J. **Add correct driver and regen.** Edit the S800 file to contain the correct driver for your system type.

Flowchart 2: Configuration Test — cont.

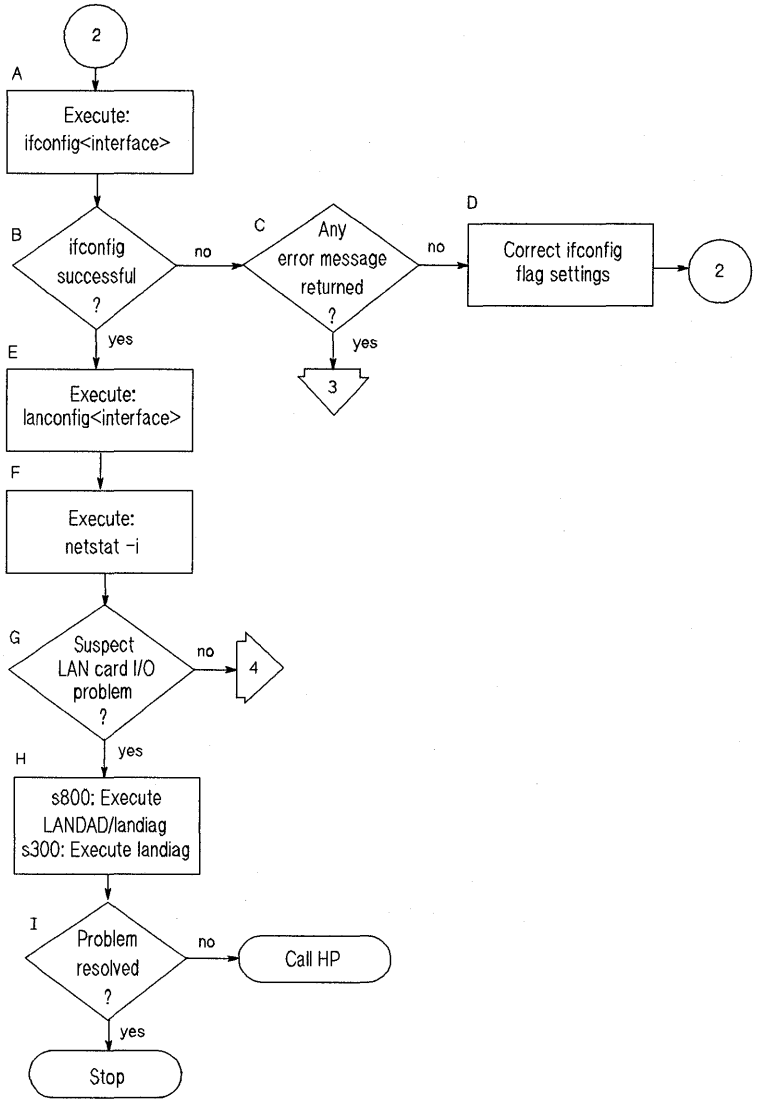


Figure 3-5. Flowchart 2

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
```

If your system is *diskless* or you are trying to connect with NetIPC, *ether* and *ieee* must be enabled. The default is *ether* only.

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.

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

This page left intentionally blank.

Flowchart 3: Configuration Test — cont.

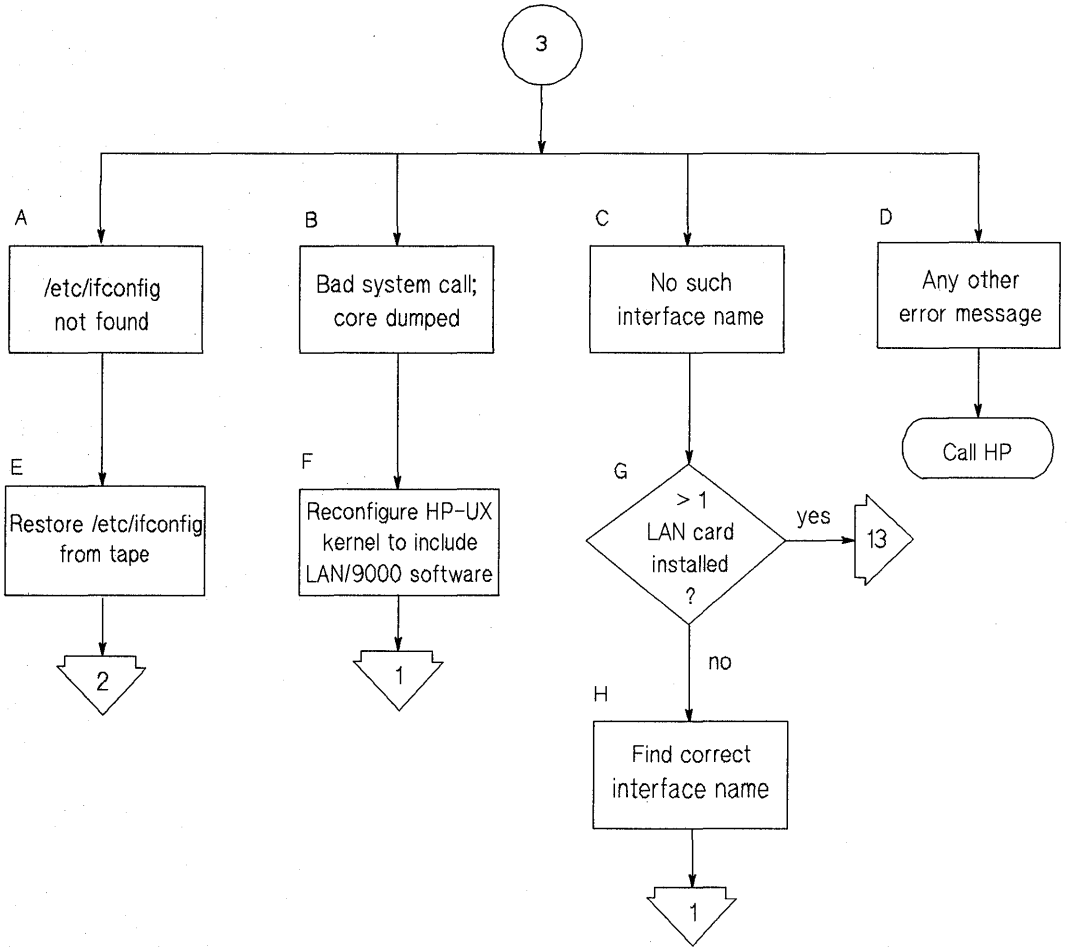


Figure 3-6. Flowchart 3

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.** You can restore *ifconfig* from the last good backup tape or your install/update tape. Go to Flowchart 2.

F. **Reconfigure HP-UX kernel to include LAN/9000 software.** Edit the *dfile* (or *S800*) file to include LAN/9000 software. Refer to chapter 4 for a list of LAN drivers.

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

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

Flowchart 4: Network Level Loopback Test

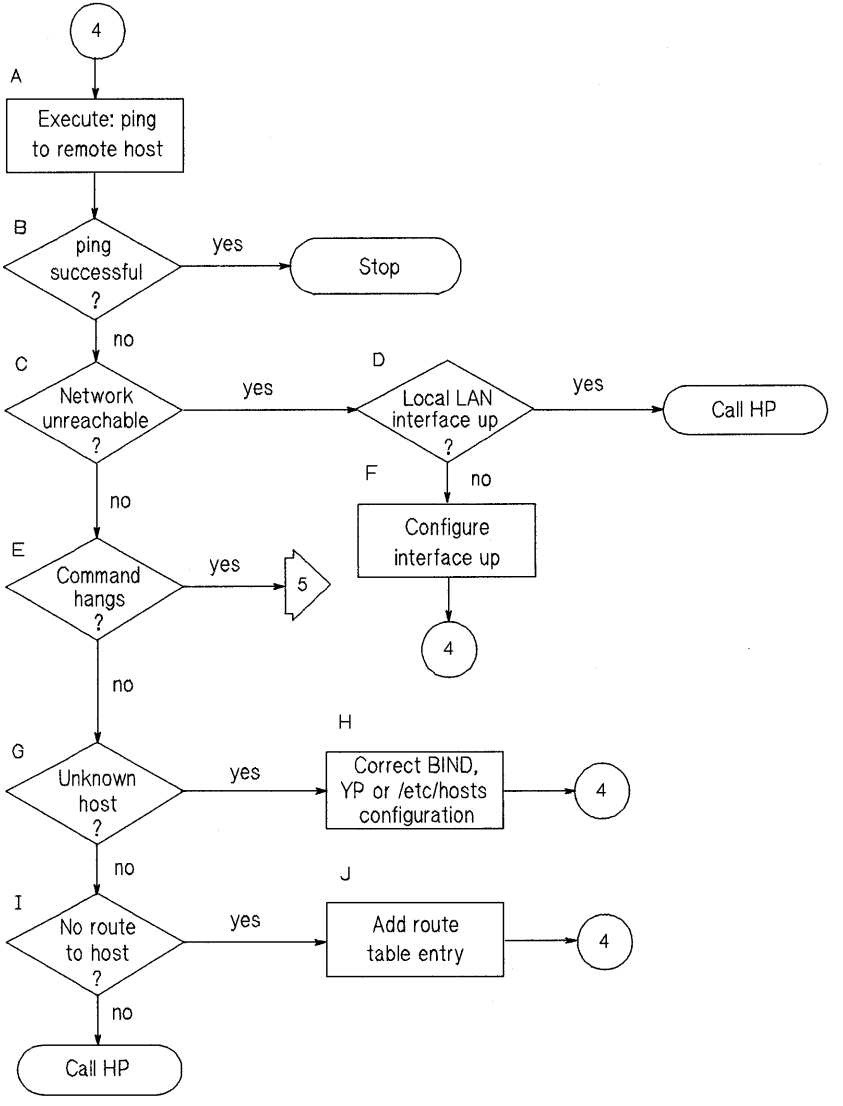


Figure 3-7. Flowchart 4

Flowchart 4 Procedures

A. **Execute: ping to remote host.** Using *ping(1M)*, send a message to the remote host with which you are having problems connecting. For example, suppose this host is known as 192.6.20.2. Enter:

```
/etc/ping 192.6.20.2
```

Note HP recommends using the IP address, rather than the hostname, as part of the problem may be an error in the */etc/hosts* file or connectivity with a name server.

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.

This page left intentionally blank.

Flowchart 5: Network Level Loopback Test — cont.

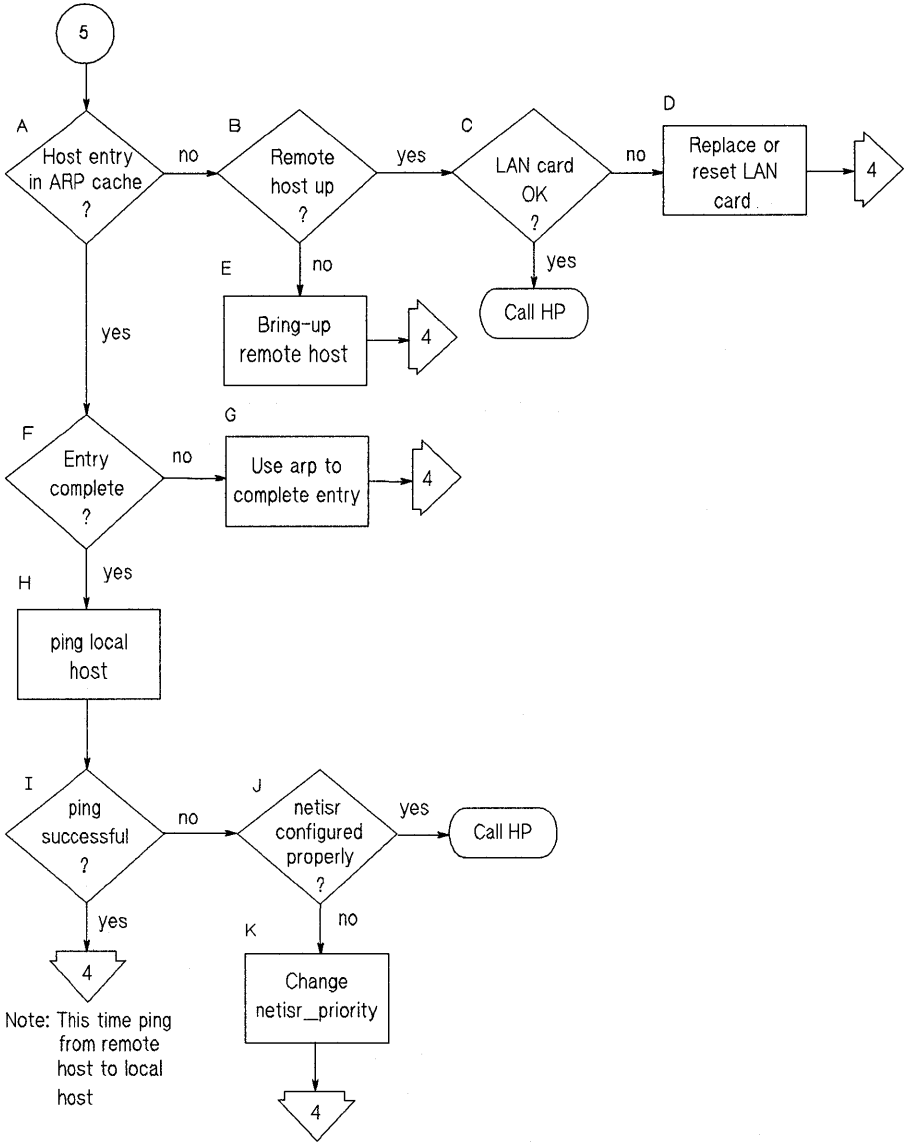


Figure 3-8. Flowchart 5

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 as 192.6.20.2. Enter:

```
/etc/arp 192.6.20.2
```

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 remote LAN card is operational.

D. **Replace or reset LAN card.** When the LAN card is operational, use *landiag (1M)* to reset. Refer to the *landiag(1M)* command description in chapter 6.

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 configured properly?** Use SAM to determine the correct value for *netisr_priority*. Refer to "Installing for Real-Time Use" in

chapter 4 for detailed instructions. Normally, *netisr_priority* is -1, indicating that *netisr* runs as an interrupt. If *netisr_priority* is a value between 1 and 127, use the following command:

```
ps -el
```

to verify that the *netisr_priority* is higher (lower-numbered) than any other networking processes.

K. **Change netisr_priority.** 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`. Regen the kernel and reboot. Refer to "Installing for Real-Time Use" in chapter 4 for instructions on how to change *netisr_priority*.

This page left intentionally blank.

Flowchart 6: Transport Level Loopback Test (using rlb)

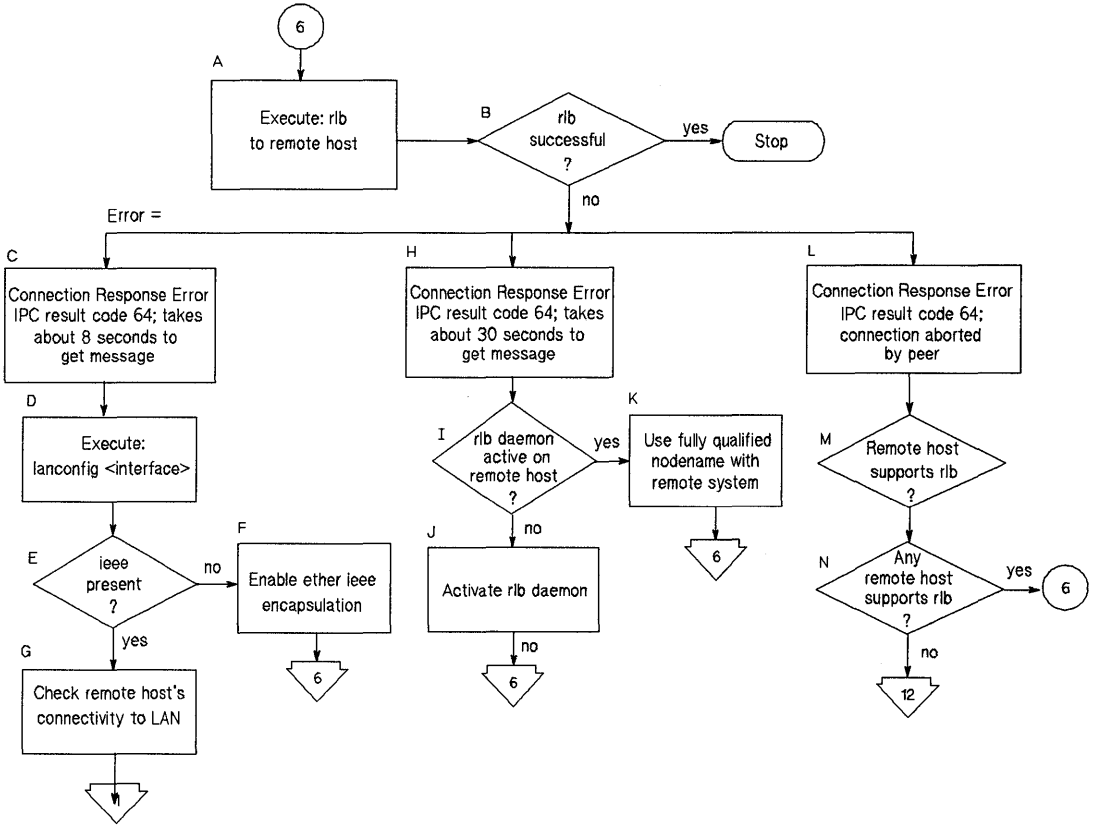


Figure 3-9. Flowchart 6

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 okay 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. **Execute: lanconfig <interface>.** Execute the *lanconfig(1M)* command on the network interface to determine the active encapsulation method(s).

E. **ieee present?** If ieee is present, go to G. If not, go to F to enable it.

F. **Enable ether ieee encapsulation.** Execute the *lanconfig(1M)* command to enable ieee encapsulation as follows:

```
lanconfig lanx ether ieee
```

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

H. **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.

I. **rlb daemon active on the remote host?** Is the rlbdaemon installed and active on the remote hosts? 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.

J. **Activate rlb daemon.** To activate the rlbdaemon, execute:

```
/etc/rlbdaemon
```

K. **Use fully qualified nodename with remote system.** Execute rlb again using the fully qualified nodename of the remote system.

L. **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.

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

N. **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.

This page left intentionally blank.

Flowchart 7: Transport Level Loopback Test (using ARPA)

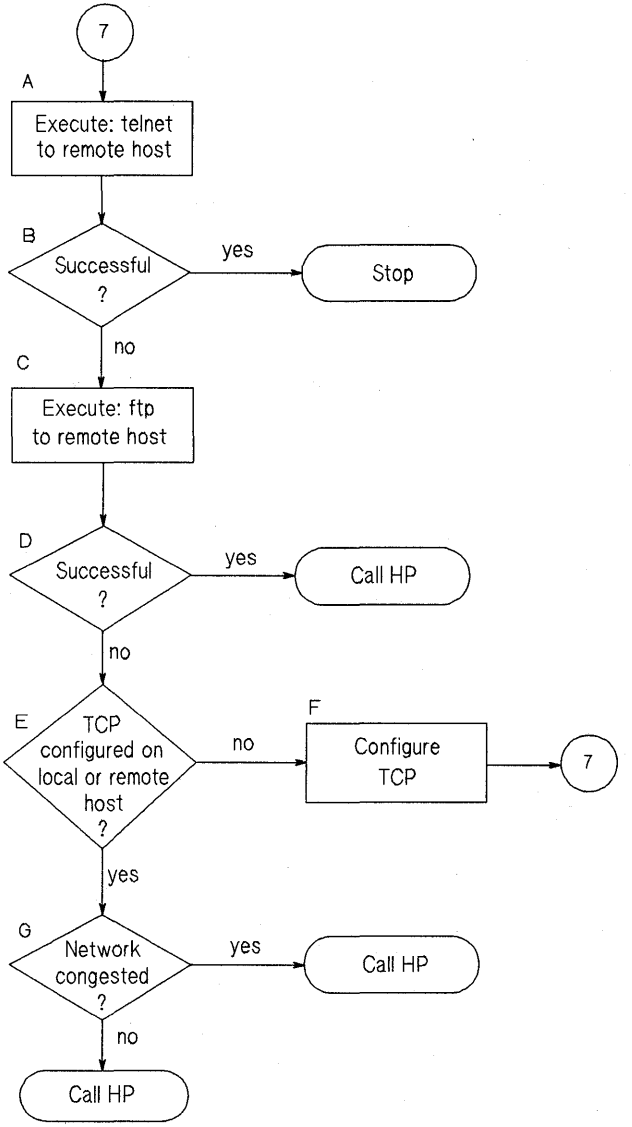


Figure 3-10. Flowchart 7

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 okay through the Transport Layer (OSI Layer 4).

C. **Execute: ftp to remote host.** Unlike *telnet*, *ftp* does not go through a pseudo-terminal 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 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 both *ftp* and *telnet* fail, the */etc/inetd.conf* file may be misconfigured or the *inetd* daemon may not be running on the remote system.

If the problem is not resolved, more detailed diagnostics are required. Again, contact your HP representative.

Flowchart 8: Link Level Loopback Test

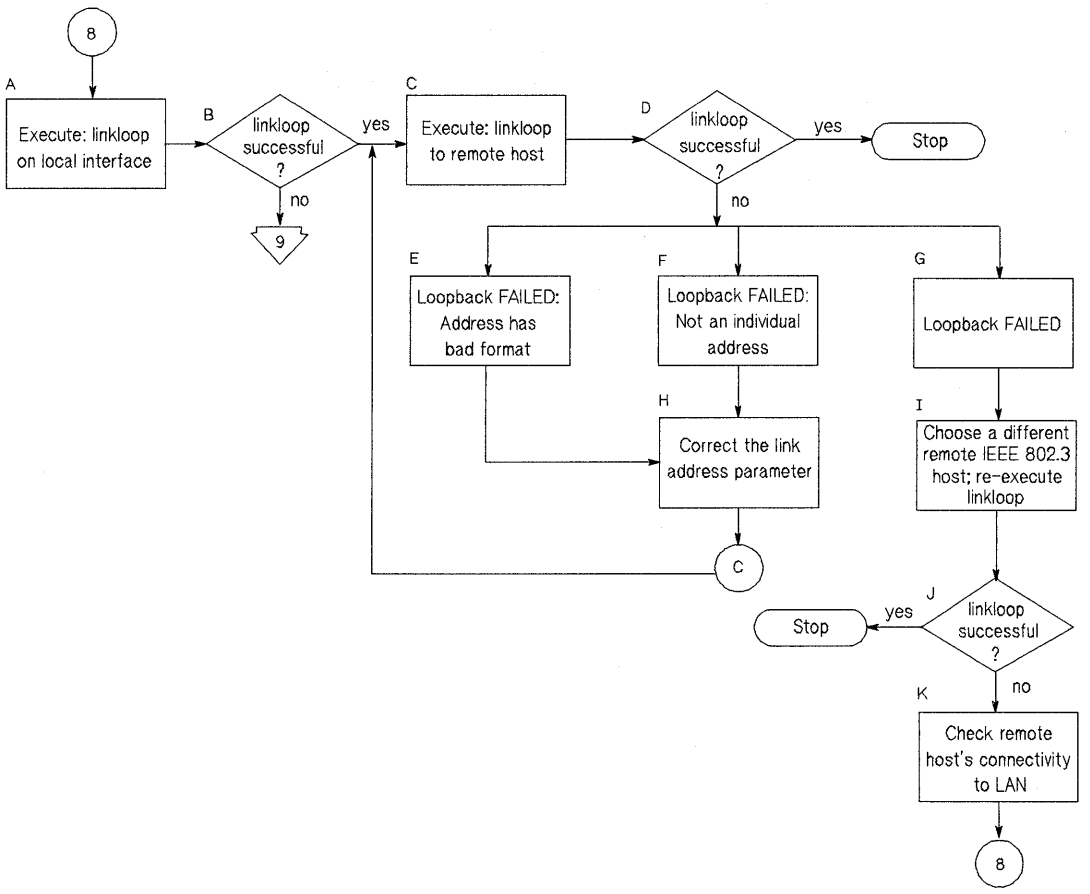


Figure 3-11. Flowchart 8

Flowchart 8 Procedures

A. **Execute: linkloop on local interface.** Execute the *linkloop* command with the station address of the local interface. 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 not, your LAN card may not be operational. Go to Flowchart 9.

C. **Execute: linkloop to remote host.** Enter the link level address (station address) of the remote host in hexadecimal form (preceded by "0x").

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

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

F. **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 F.

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

H. **Correct the link address parameter.** Change the link level address to an allowed value and go to C.

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

J. **linkloop successful?** If the test was successful, stop. Network connectivity is okay through the Link Layer (OSI Layer 2). If not successful, go to I.

K. **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)

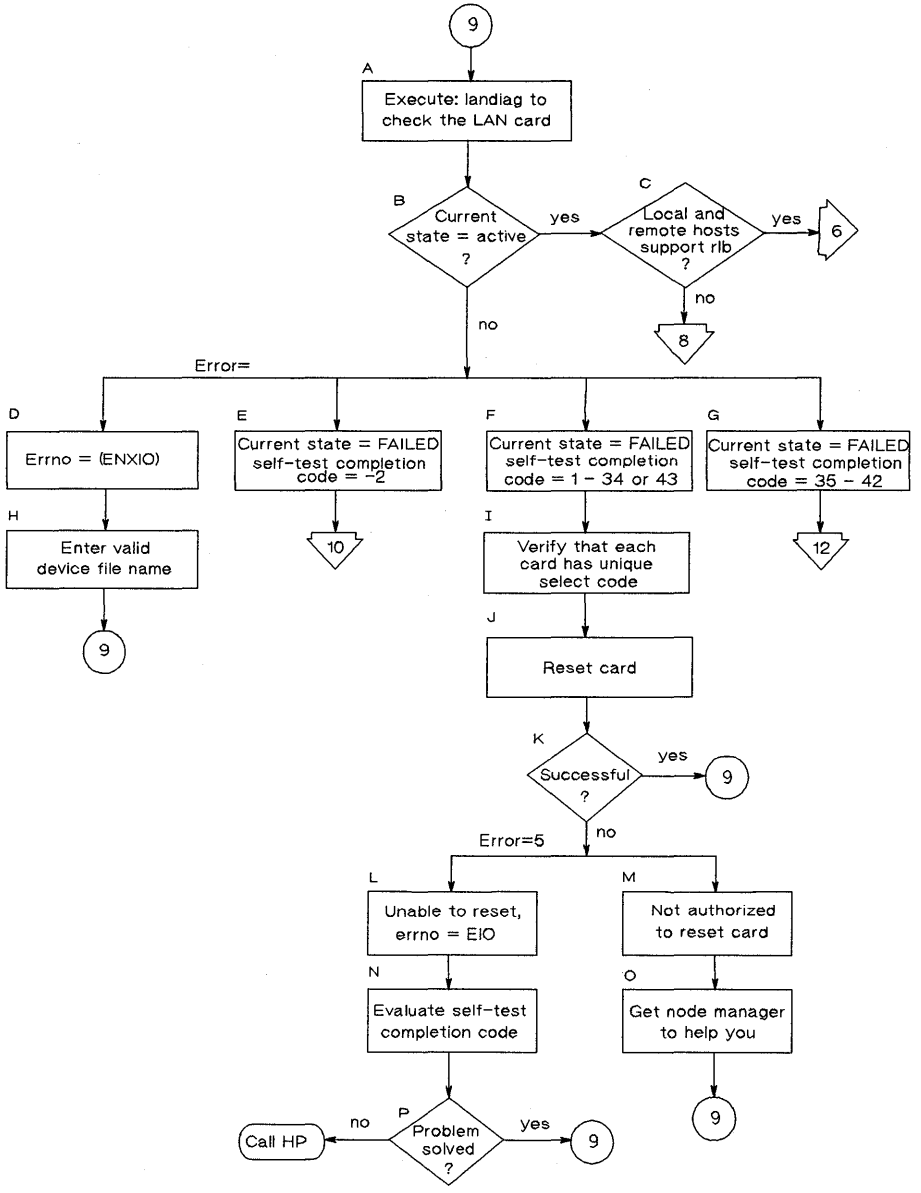


Figure 3-12. Flowchart 9

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 (okay), 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.** Run the *reset* command in *landiag* to re-execute 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.

This page left intentionally blank.

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

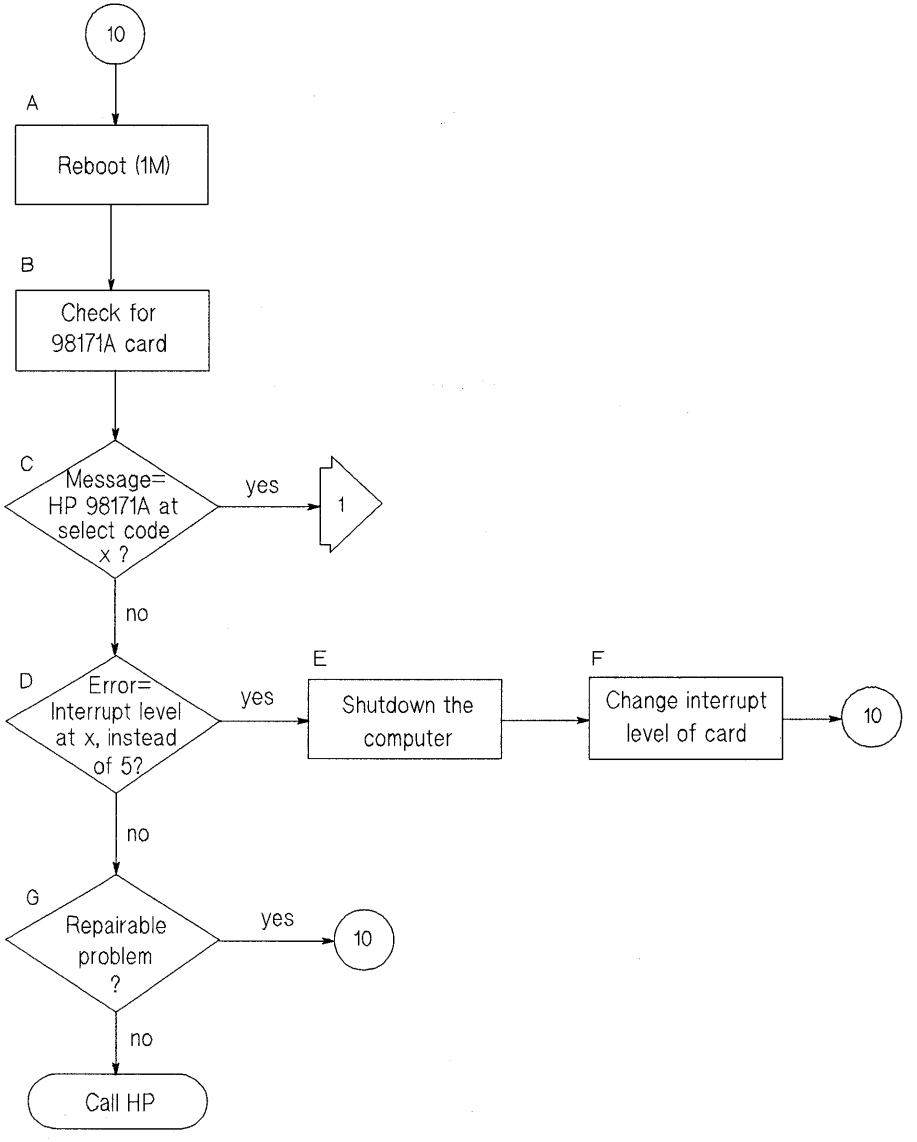


Figure 3-13. Flowchart 10

Flowchart 10 Procedures

A. **Reboot (1M).** Reboot and then enter the *landiag* lan mode and use the *display* command to ensure LAN card status is active. 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.** Execute *reboot -h* and 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 your 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)

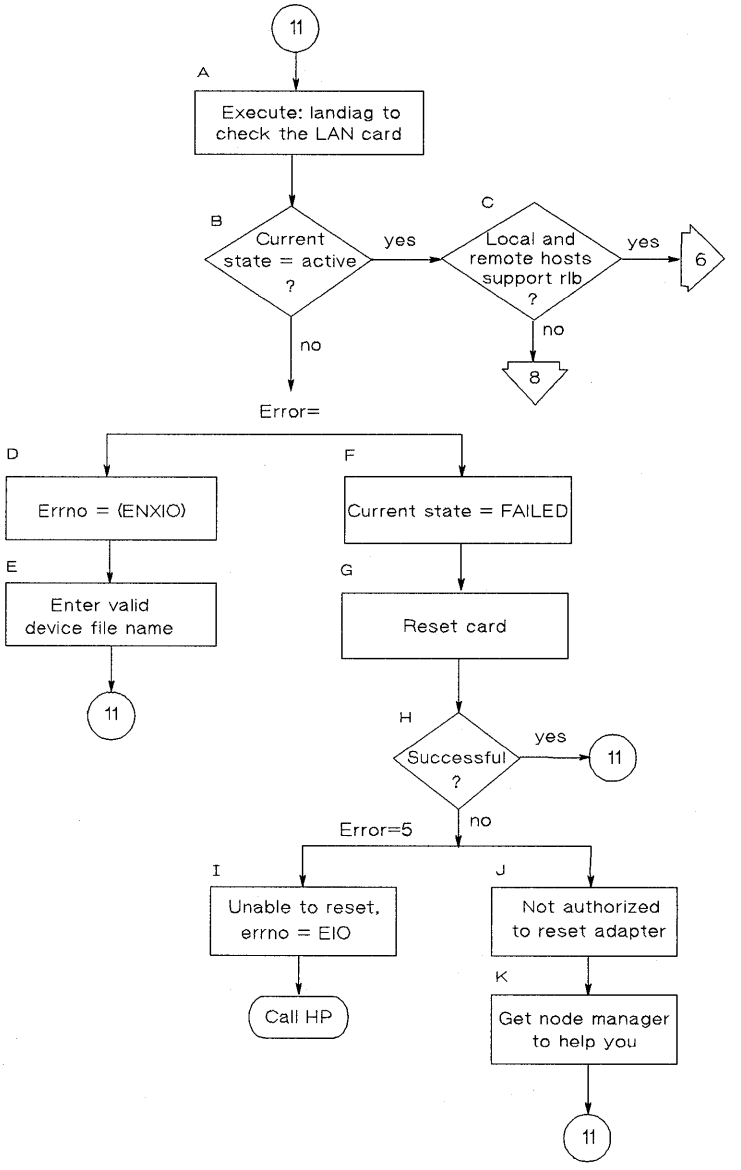


Figure 3-14. Flowchart 11

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 (okay), 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 G.

G. **Reset card.** Run the *reset* command in *landiag* to re-execute 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

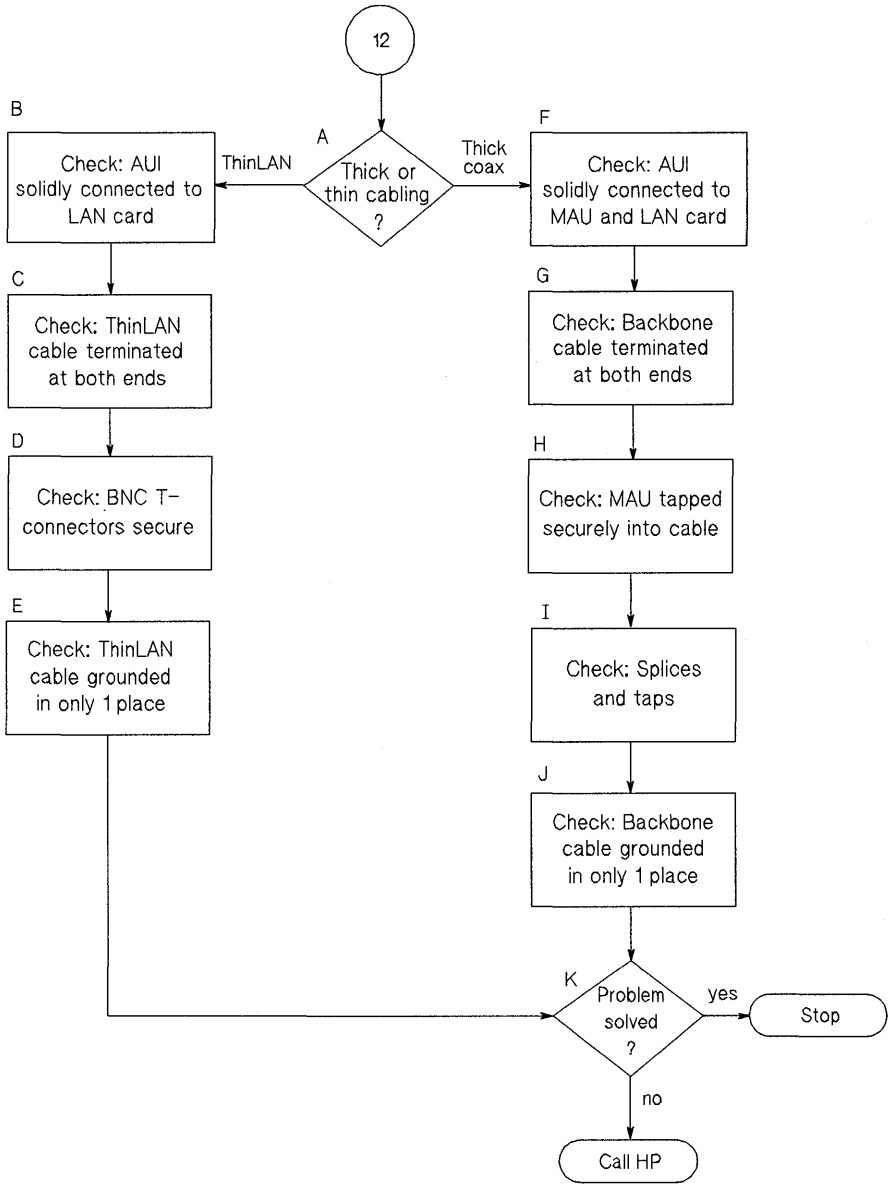


Figure 3-15. Flowchart 12

Flowchart 12 Procedures

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

B. **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.

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

D. **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.

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

F. **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.

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

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

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, an incorrect jumper setting on the 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 Remote Loopback Test

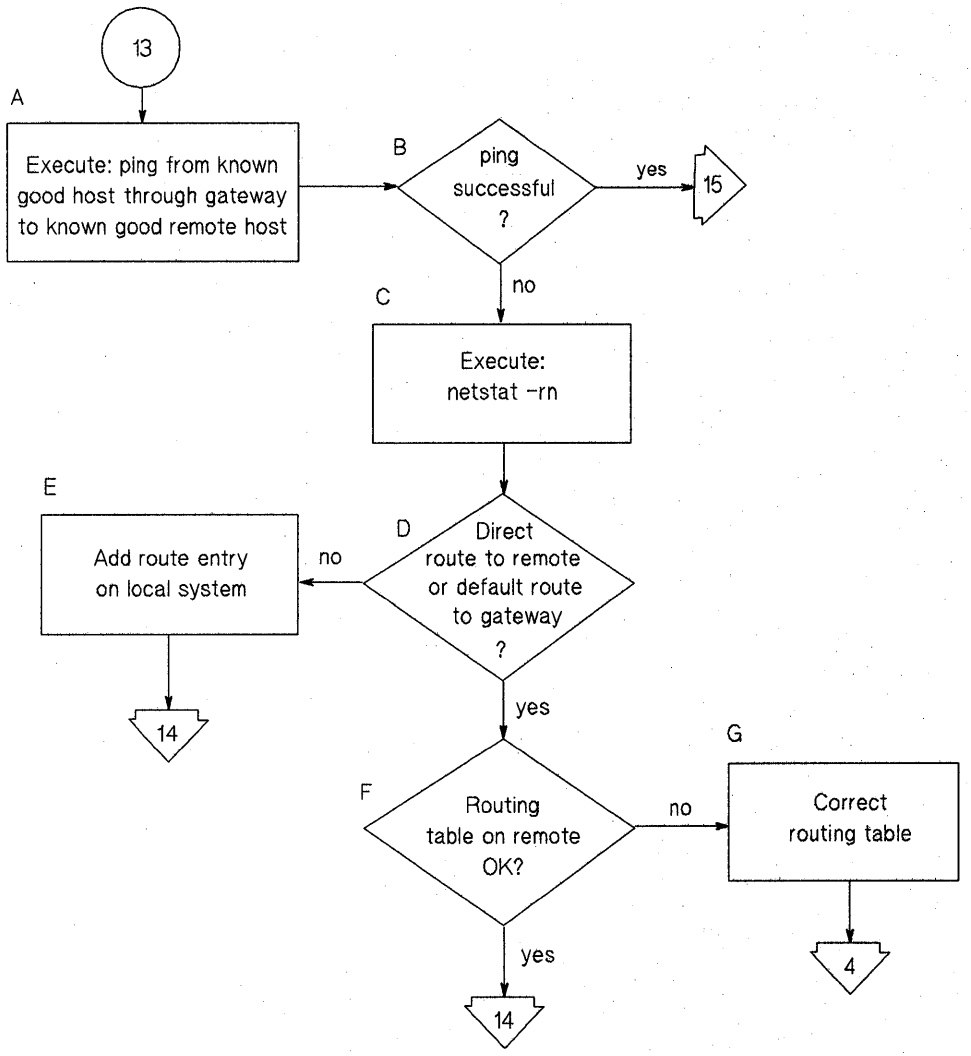


Figure 3-16. Flowchart 13

Flowchart 13 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. **ping successful?** If the executing *ping* returned successfully, the problem may exist in the routing table for the problem host. Go to C.

C. **Execute *netstat -rn*.** To display gateway routing information in numerical form, execute:

```
netstat -rn
```

D. **Direct route to remote or default route to gateway?** If the route exists, go to F. If not, go to E to add a new route.

E. **Add route entry on local system.** Use the *route(1M)* command to add a route entry to the route table on the local system. Refer to *route(1M)* in chapter 5 for a complete description of the command.

F. **Routing table on remote OK?** Check that the routing information on the remote system is OK.

G. **Correct routing table.** If the routing information is incorrect, correct it using the *route(1M)* command.

Flowchart 14: Gateway Remote Loopback Test — cont.

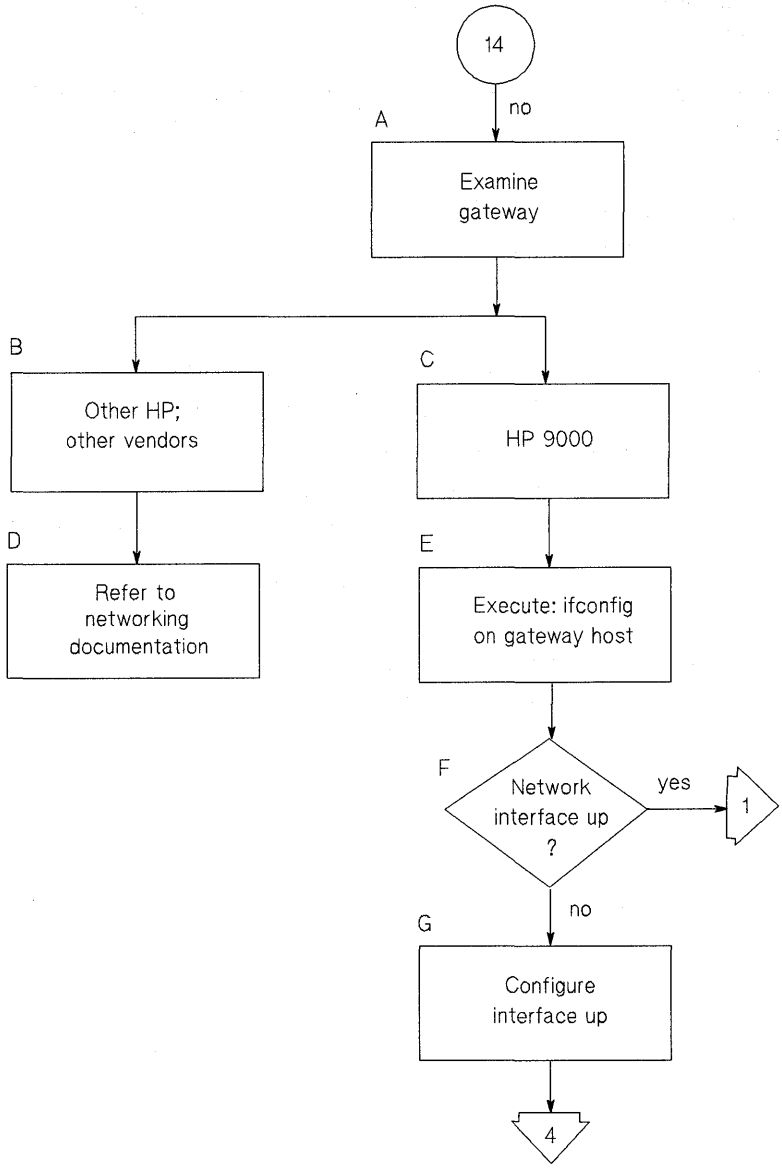


Figure 3-17. Flowchart 14

Flowchart 14 Procedures

A. **Examine gateway.** If the gateway is an HP 9000, go to C. If it is not, go to B.

B. **Other HP; other vendors.** Go to D.

C. **HP 9000.** Go to E.

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

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

F. **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.

G. **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

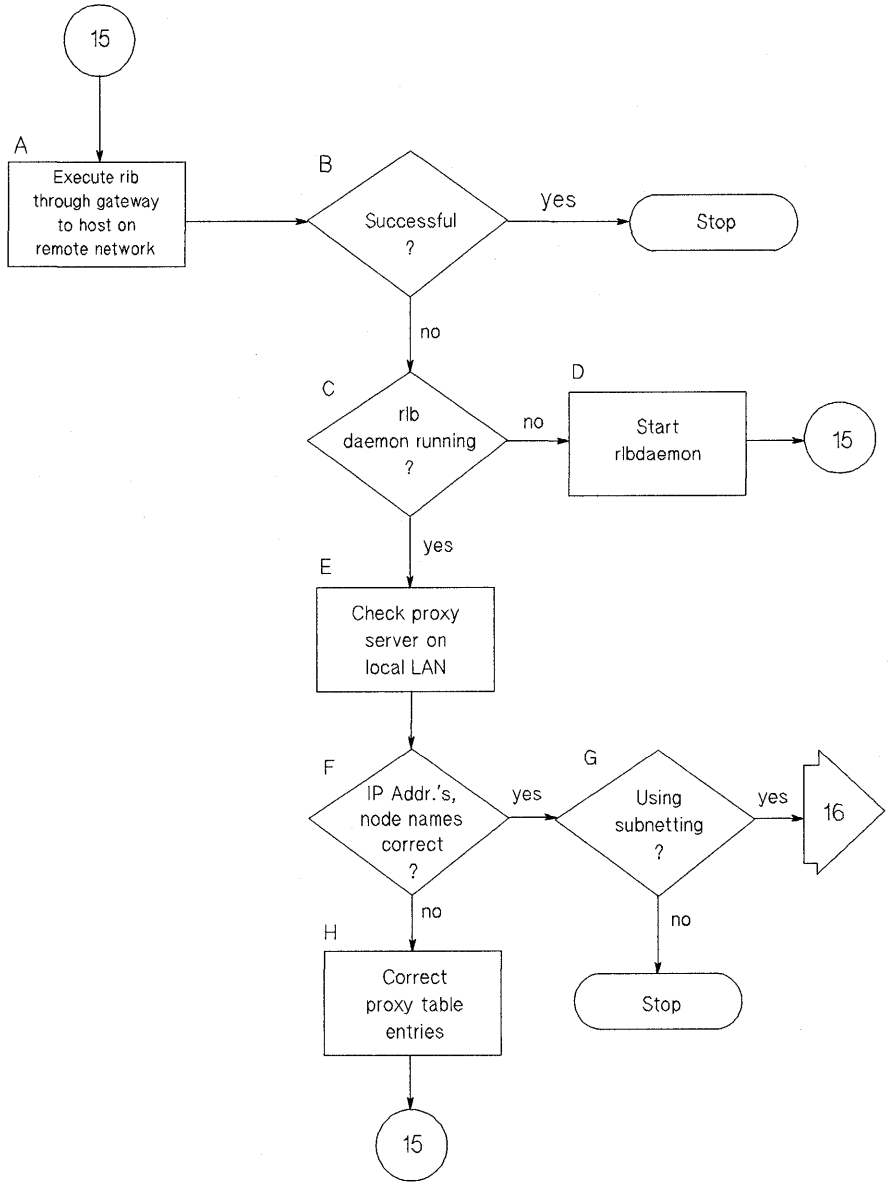


Figure 3-18. Flowchart 15

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 rlbdaemon* 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

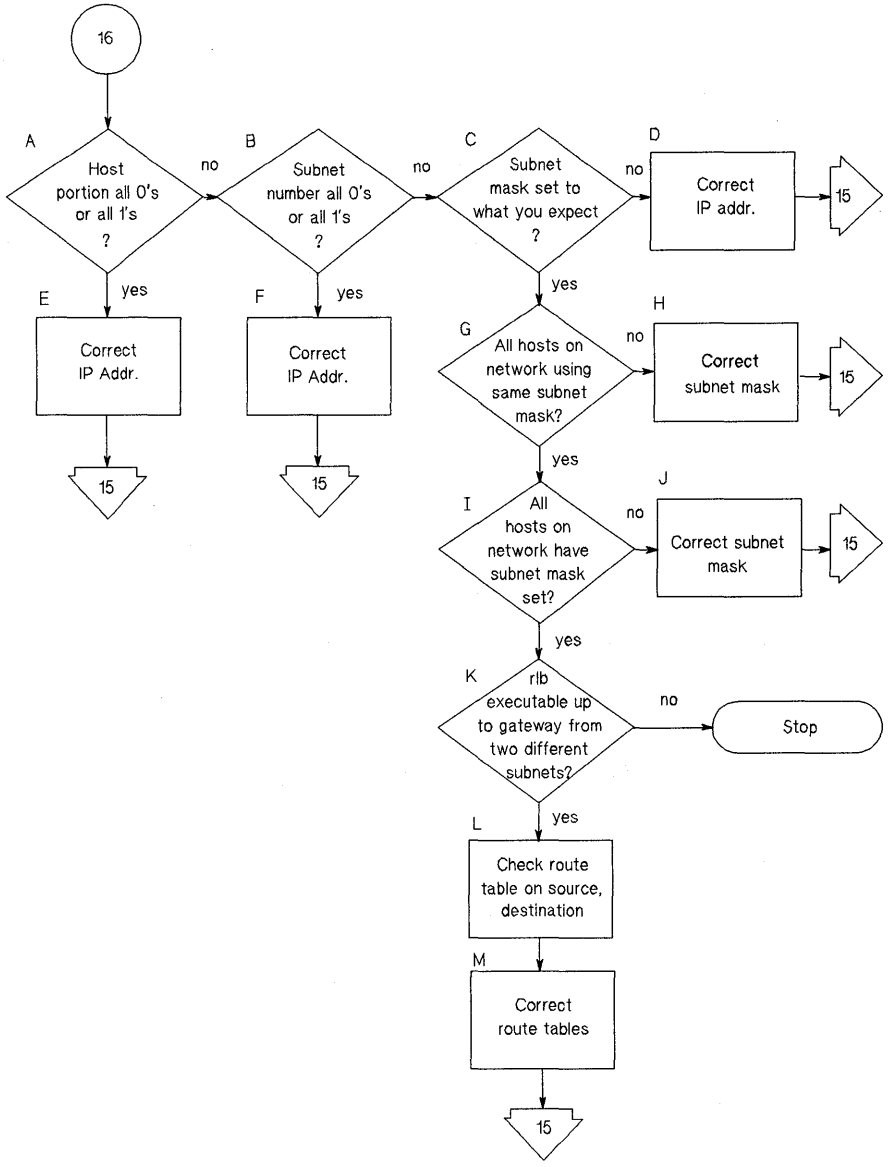


Figure 3-19. Flowchart 16

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 9 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 9 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 9 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/savecore* to save a core dump. Send the output to your HP representative.

4

Manually Configuring LAN/9000

This chapter provides information on manually configuring LAN/9000 software. It contains the following sections:

- Overview of Manual Configuration.
- Creating a New Kernel for the Series 700 or Series 300/400.
- Creating a New Kernel for the Series 800.
- Verifying LAN Device Files.
- Creating the */etc/hosts* File.
- Editing and Executing the */etc/netlinkrc* File.
- Activating Optional Network Features.
- Installing for Real-Time Use.

Overview of Manual Configuration

This chapter describes how to manually create the kernel, LAN device files, and the */etc/hosts* file. It also provides instructions on editing and executing the */etc/netlinkrc* file for LAN/9000.

The following table shows when these steps are completed automatically by the *update* utility and SAM in the procedures described in chapters 1 and 2.

Utility	Task Descriptions
update	Downloads software from tape.
	Configures LAN driver into the kernel.
eisa_config	Adds the card configuration file (Series 700 only).
SAM	Makes device files using the *mknod* command (Series 700 and Series 300/400).
	Adds local host name and IP address to */etc/hosts.*
	Assigns an IP address using the *ifconfig* command.
	Optionally, adds the *route* command to connect to remote hosts.
	Adds the *ifconfig* and *route* commands to the */etc/netlinkrc* file.

Note The LAN card should be installed **after** loading the software filesets and generating the kernel with the *update* utility and **before** configuring the system manually.

Creating a New Kernel for the Series 700 or Series 300/400

Note The instructions below do not apply to clustered systems. If your system is attached to a cluster, follow the instructions in *System Administration Tasks* for Series 700 or Series 300 computers to configure the kernel. Alternatively, you can also create a new kernel using the SAM utility.

Complete instructions on how to create a kernel on Series 700 and Series 300/400 standalone systems are provided in Appendixes H and I. The steps below contain a general outline of kernel configuration steps with detailed information for the LAN/9000 drivers and parameter settings contained within that framework.

If the kernel was not created with the LAN driver in it during the *update* procedure, you can create it manually. To determine whether the LAN driver is in the kernel, check the *dfile* for the `lan01` keyword. To determine whether the LAN driver is installed, execute *what hp-ux* or */etc/dmesg* and check for references to LAN in the output.

The numerical steps below correspond to the numerical steps that describe how to regen your kernel in appendixes H and I. **The alphabetical steps provide detailed information about the networking drivers and parameter settings that should be modified during Steps 3 and 4 of the procedures in appendixes H and I.** If you used some other file to create the kernel previously, copy that file to *dfile* before following the steps below.

1. Ensure that you have super-user capabilities.
2. Change (`cd`) to the */etc/conf* directory.
3. Make a backup copy of your current configuration description file (which is commonly */etc/conf/dfile* or */etc/conf/dfile.SAM*).
   a. 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.
   b. To do so, enter the command:

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

4. Edit the */etc/conf/dfile* to add drivers and/or change system parameters.

   a. If you have not installed LAN/9000 previously, remove the comment delimiters, /* and */, around the keyword:

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

      If the keyword does not exist in the *dfile,* add the lines, `lan01` and `lla`, to the file.

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

      ```
      uipc
      nipc
      inet
      netman
      ni
      dskless
      nfs
      netdiag1
      ```

      Refer to Appendix F for a table showing the correspondence between filesets and subsystem names. Neither NFS nor Diskless are covered in this manual. Refer to *HP-UX System Administration Tasks* for the Series 300, chapter 10, for more information on Diskless and to *Using NFS Services* for more information on NFS.

   c. **Series 300 only:** If you want to add three 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.

5. Make a copy of the existing kernel.

6. Write down the hardware address of the system disk.

7. Run *config* on the configuration description file you edited.

8. Create the new hp-ux kernel (the file *hp-ux*) in the current directory (*/etc/conf*).

9. Bring the system into single-user mode using the *shutdown* command.

10. Wait for the system to display from single-user mode.

11. Copy the new kernel to the */(root)* directory.

12. **Series 700 only**: Go to chapter 1 and complete the instructions in "Step 3: Adding the Card Configuration File" to add the configuration file to the EEPROM memory chip on the EISA interface.

13. Halt the system.

14. Go to chapter 1 and complete the instructions in "Step 4: Installing LAN Hardware."

15. Proceed to the section "Verifying LAN Device Files," in this chapter to configure LAN manually, or chapter 2, "Configuring LAN Using SAM," to configure your system with the SAM utility.

Creating a New Kernel for the Series 600/800

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

Complete instructions on how to create a kernel on Series 600/800 systems are provided in Appendix J. The steps below contain a general outline of kernel configuration steps with detailed information for the LAN/9000 drivers and parameter settings contained within that framework. Alternatively, you can also create a new kernel using the SAM utility.

If the kernel was not created with the LAN driver in it during the *update* procedure, you can create it manually. To determine whether the LAN driver is in the kernel, check the *S800* file in the */etc/conf/gen* directory for the `include lan` statement. To determine whether the LAN driver is installed, execute *what hp-ux* or */etc/dmesg* and check for references to LAN in the output.

The numerical steps below correspond to the numerical steps that describe how to regen your kernel in appendix J. **The alphabetical steps provide detailed information about the networking drivers and parameter settings that should be modified during Step 4 of the procedure in appendix J.** If you used some other file to create the kernel previously, copy that file to *S800* before following the steps below.

1. Ensure that you have superuser capabilities.
2. Change (`cd`) to the */etc/conf/gen* directory.
3. Save the old *S800* file.
4. Edit the *uxgen* input file (*S800* is the default *uxgen* input file) to add drivers and/or change system parameters and save it.

   Remove the comment delimiters /* and */ from the following lines of the *uxgen* input file to include the LAN drivers into the kernel according to the filesets that you have loaded.

- If you are creating a CIO-based kernel (Series 835, 840, 945, 850, or 855 system), you must remove the comment delimiters from `/*include lan0; */`.
- If you are creating an HP-PB-based 8X2 kernel (Series 808, 815, 822, 832, 842, or 852 system), you must remove the comment delimiters from `/*include lan1; */`.
- If you are creating an HP-PB-based 8X7 kernel (Series 817, 827, 837, 847, 857, 867, 877, 887, 890, or 897 system), you must remove the comment delimiters from `/*include lan3; */`.

Refer to table 10-1 for a complete list of HP-UX system types and their corresponding major numbers.

Below is a complete list of ARPA networking *include* statements. Refer to Appendix F for a listing of filesets and corresponding subsystem names. The file, *netdiag1,* should always be in the *uxgen* input file by default.

```
/*include uipc;*/
/*include nipc;*/
/*include inet;*/
/*include ni;*/
/*include nm;*/
/*include nfs;*/
/*include lan;*/
/*include lan0;*/(CIO only - Refer to list above.)
/*include lan1;*/(HP-PB only - Series 8X2 - Refer to list above.)
/*include lan3;*/(HP-PB only - Series 8X7 - Refer to list above.)
```

5. Regenerate the kernel with *uxgen*, using the edited *S800* file as input. Estimated time: 15 to 20 minutes.
6. Make a copy of the existing kernel.
7. Write down the name of the information you'll need in case the new kernel doesn't boot.
8. Copy the new kernel to the root directory.
9. Reboot on the new kernel. If the new kernel fails to boot, boot the system from the backup kernel and repeat the process of creating a new kernel. To do so, follow the instructions in appendix J.

Verifying LAN Device Files

Device files are used to identify the LAN driver, card, and data link protocol. Each driver/card is associated with a device file. By convention, device files are kept in a directory called */dev,* with each device file having a name and device number to uniquely identify the above characteristics.

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.

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. A detailed explanation of the *mknod(1M)* command is provided in the HP-UX *mknod(1M)* manual page.

Series 700 Device Files

For each Series 700 LAN card that is bound successfully to the I/O subsystem at 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 file names.

Example 1: EISA LAN card in slot 3 of Series 750 workstation.

To display the device files, issue the following HP-UX command:

```
ls -l /dev/lan* /dev/ieee* /dev/ether*
```

If you installed an Ethernet/802.3 card in slot 3 of a Series 750 workstation, the display will be as follows:

```
crw-rw-rw-   1  root   sys  52  0x202000  Mar 14  1991   /dev/lan0
crw-rw-rw-   1  root   sys  52  0x202000  Mar 14  1991   /dev/ieee0
crw-rw-rw-   1  root   sys  52  0x202001  Mar 14  1991   /dev/ether0
crw-rw-rw-   1  root   sys  52  0x430000  Mar 14  1991   /dev/lan1
crw-rw-rw-   1  root   sys  52  0x430000  Mar 14  1991   /dev/ieee1
crw-rw-rw-   1  root   sys  52  0x430001  Mar 14  1991   /dev/ether1
```

The fifth column is the major number. The sixth column is the minor number. The first two-digit field in the minor number indicates the IO functionality supported by the card, in this case, 2 for the built-in LAN and 4 for EISA, and the slot number of

the card in the system backplane, in this case, 0 for the built-in LAN and 3 for the card in slot 3.

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.

Series 300/400 Device Files

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: On-board LAN card on Series 300 workstation.

To display the device files, 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 add-on LAN card, and create three device files, */dev/lan*x, */dev/ieee*x, and */dev/ether*x, 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.

Series 600/800 Device Files

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

Example 1: Three CIO LAN cards on Series 800 system

To display the device files, use the following command:

```
ls -l /dev/lan* /dev/ether*
```

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: Two HP-PB LAN cards in Series 800 system.

To display the device files, use the following HP-UX command:

```
ls -l /dev/lan* /dev/ether*
```

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 HP-PB LAN driver Series 8X2 systems (808, 815, 822, 832, 842, and 852) or 32 for the HP-PB LAN driver Series 8X7 systems (817, 827, 837, 847, 857, 867, 877, 890, and 897)). 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).

Creating the /etc/hosts File

You must edit the */etc/hosts* file to add an IP address and hostname for the LAN card that you are installing.

Note If you are using a domain name service (DNS or NIS), you will need to modify */etc/hosts* on the name server system.

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.
- By copying the official host database maintained at the Network Information Control Center (NIC) for ARPA Internet networks, if you are installing ARPA Services/9000. (Refer to "Military Standards and Request for Comments Documents" section of Appendix L 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.

Network and System Names

A system is known by several names, each with its own purpose:

System name	Used for cluster configuration and UUCP communication.
Host name and aliases	Used for most network communication. This might include the Internet domain name used with the DNS/9000 product.
NetIPC node name	Used with NS/9000 NetIPC products.

HP recommends that you try to keep these names as consistent as possible, within their limitations. This will help to minimize confusion.

The examples below show how a system with the name, *host3*, might be referenced in the */etc/hosts* and also other system and networking files and commands:

System name in Install screen: *host3*

/etc/src.sh: *SYSTEM_NAME=host3**

/etc/hosts: *192.6.1.1** *host3** *host3.site2.region4*

/etc/clusterconf: *080009005900:7:host3:c:1:1** (Will only be on a server.)

/etc/netlinkrc: *DOMAIN=site2; ORGANIZATION=region4*

uname -S *host 3**

hostname *host3**

nodename *host3.site2.region4*

Note When you first install a system, the operations indicated above with an asterisk(*) are done for you automatically.

Caution For usage with NetIPC, you must edit the */etc/netlinkrc* file manually to include the correct DOMAIN and ORGANIZATION fields.

/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]...
```

Parameters

IP_address — The IP address that uniquely identifies the node. *IP_address* must be in internet "dot" notation. Refer to chapter 9 for details on IP addresses.

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 — 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.

/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

HP recommends that the */etc/hosts* file be owned by user *root* and have *0444 (-r—r—r—)* access permission. For more information on */etc/hosts*, refer to the hosts(4) manual page in the *HP-UX Reference Pages*.

Note HP highly recommends that you limit access to the */etc/hosts* file by setting the permission to 0444 or (-r—r—r—) for read access only.

/etc/hosts Example

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

- The IP address *192.6.1.1.*
- A hostname field such as *host3*.
- The alias names such as *host3.site2.region4* and *grace.*

looks like:

```
192.6.1.1   host3   host3.site2.region4  grace
```

Editing and Executing the /etc/netlinkrc File

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

a. Starts network logging.

b. Configures the network interface with an IP address.

c. Sets the default encapsulation method.

d. Starts X.25/9000 (if installed by invoking */etc/X25init*).

e. Configures the network routing table if your node is a gateway or on a LAN with a gateway.

f. If the NetIPC fileset has been loaded, assigns a network (NS) node name to be used by *rlb*(1M) and NS/9000.

g. If the NetIPC fileset has been loaded, starts the NetIPC remote loopback daemon and the *rlbdaemon*, respectively.

h. Starts NFS/9000 (if it is installed) by invoking the NFS initialization script */etc/netnfsrc*.

i. Starts the Internet daemon (*inetd*).

j. Starts ARPA Services/9000 (if it is installed) by invoking the */etc/netbsdsrc* initialization script.

k. Continues NFS/9000 startup (if it is installed) by invoking the NFS initialization script */etc/netnfsrc2*.

l. Starts NS/9000 (if it is installed) by invoking the */etc/netnssrc* initialization script.

m. Starts the HP Network Management Agent (if installed) by invoking */etc/netnmrc*.

n. Starts the HP LAN Manager/X (if installed) by invoking */etc/netlmrc*.

o. Starts NCS/9000 (if installed) by invoking */etc/netncsrc*.

p. Starts NetLS/9000 (if installed) by invoking */etc/netlsrc*.

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

When a system is booted up, */hp-ux* calls */etc/inittab* to start the initialization process. This program then calls the */etc/rc* script, which in turn calls */etc/netlinkrc* to initialize networking. The alphabetic references in Figure 4-1 below, map to the corresponding letters in the sample */etc/netlinkrc* file in appendix L.

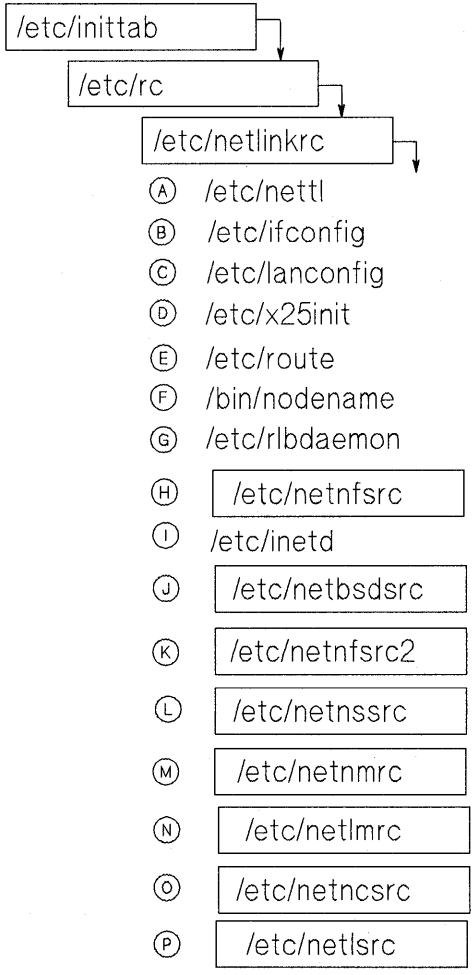


Figure 4-1. LAN Startup Files

Editing /etc/netlinkrc

Before executing */etc/netlinkrc*, you should edit it to identify the network interface name, IP address, and default encapsulation method of your LAN card, and add entries to the network routing table. As the */etc/netlinkrc* file has read-only permission, you must have super-user capability to make modifications to this file.

The steps to add these commands to the */etc/netlinkrc* file are listed below.

1. Adding the *ifconfig(1M)* command.

   To assign an IP address and configure network interface parameters, you must edit */etc/netlinkrc* script. See the comments under "Initialize networking interfaces" in the */etc/netlinkrc* file. A detailed explanation of */etc/ifconfig*, which includes a definition of the netmask parameter and how to use it to assign a subnet mask, is provided in chapter 5.

   Following is a sample */etc/netlinkrc* entry:

   ```
   /etc/ifconfig lan1 192.6.1.1 up
   ```

   where *lan1* is the network interface name and unit, and 192.6.1.1 is the host IP address.

   As upper layer software often requires loopback, be sure that loopback is also enabled in the */etc/netlinkrc* file. The line in */etc/netlinkrc* should read:

   ```
   /etc/ifconfig lo0 127.0.0.1 up
   ```

Note If you configure your system as a gateway, you must include one */etc/ifconfig* entry for each LAN interface and unit (LAN card). Each entry must have a separate interface name and IP address. You can use the *lanscan(1M)* command to display the network interface name and unit of each LAN card installed on your system.

---

2. Adding the *lanconfig(1M)* command.

   To set the default encapsulation method for your network interface, you must edit the */etc/lanconfig* entry in the */etc/netlinkrc* script. A detailed explanation of */etc/lanconfig* is provided in chapter 5.

Following is a sample */etc/netlinkrc* entry:

```
/etc/lanconfig lan1 ether ieee
```

where *lan1* is the network interface name, *ether* indicates that the Ethernet protocol is to be enabled for the network interface and *ieee* indicates that the IEEE 802.3 protocol is also to be enabled for the network interface.

3. Adding the *route(1M)* command.

   If you intend to use your system as a gateway or to communicate with gateways, you must edit the */etc/netlinkrc* script. See the comments under "Initialize network routing" in the */etc/netlinkrc* file. A detailed explanation of */etc/route* is provided in chapter 5.

   Following is a sample */etc/netlinkrc* entry:

```
/etc/route add net 192.6.12.0  192.6.12.132 1
```

   where 192.6.12.0 is the IP network address of the destination network, and 192.6.12.132 is the IP address of the gateway to that destination.

Note This step is required only if your node is a gateway or you intend to use gateways from your node.

Note When the LAN/9000 software is loaded, the only entries in the routing table are the loopback interface, called *lo0*, and the Core I/O LAN interface, called *lan*0. The *lo0* entry corresponds to the *loopback* entry in the */etc/networks* file. When the software is initialized, other entries are created for each LAN card installed: *lan1*, *lan2*, etc.

Note After adding the *ifconfig, lanconfig,* and *route* entries into the */etc/netlinkrc* file, you can either execute the same commands manually, re-execute */etc/netlinkrc*, or reboot your system.

4. Assigning an NS/9000 Node Name

To assign an NS node name to your system, remove the comment delimiter and edit 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.

Following is a sample entry:

```
/bin/nodename  host3.site2.region4
```

where *host3.site2.region4* is the assigned node name.

For more information on node names, refer to table 9-2.

Executing /etc/netlinkrc

Once you have edited the */etc/netlinkrc* script, you should complete the following steps:

1. Verify that a */etc/netlinkrc* statement exists in */etc/rc* to execute */etc/netlinkrc*.
2. Execute `/etc/netlinkrc`.
3. Follow the steps in "Step 2: Verifying the Installation" at the end of chapter 2.

Note The system name (for example, *host3*), should be the value to which SYSTEM_NAME is set in */etc/src.rc*. */etc/src.rc* is invoked by */etc/rc* when the system is booted, and SYSTEM_NAME is used to get the hostname and system name, which are used by */etc/netlinkrc*. 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 Appendix L 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]...
```

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 9 for details on network addresses.

alias — 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 *bin*, group *bin*, and it should have 0444 (-r—r—r—) access permission.

For more information on */etc/networks*, refer to the *networks(4)* manual page in the *HP-UX 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 will 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]...
```

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 — 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 *bin*, group *bin*, and it should 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)* manual page 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 will also 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]...
```

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 — 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 *bin*, group *bin*, and it should 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)* manual page in the *HP-UX 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
```

After activating the optional network features, follow the instructions in "Installing /etc/netlinkrc" earlier in this chapter to complete the installation process.

Installing for Real-Time Use

The network interface daemon, *netisr*, is a packet dispatcher between the physical interface driver (for example, the LAN driver) and the network protocol layer (for example, IP).

Normally, *netisr* runs as an Interrupt Service Routine (ISR) on the Interrupt Control Stack. This provides the best throughput performance for networking activity.

This can, however, interfere with real-time applications. In that case, you may want to run *netisr* as a real-time process.

The priority of *netisr* is controlled by the *netisr_priority* variable in the *S800* configuration file (for Series 600/800) or the *dfile* configuration file (for the Series 300/400/700). The variable's values have the following meaning:

-1	*netisr* runs as an interrupt service routine.
1 to 127	*netisr* runs as a process at the specified real-time priority.

Note *netisr* must run as a higher priority (lower number) than other network services on the same system.

Changing netisr_priority Using SAM

To view or change the *netisr_priority* variable using SAM, perform the following steps:

1. At the HP-UX prompt, type: `sam`
2. Highlight *Kernel Configuration* and activate the **OPEN** button.
3. Highlight *Configurable Parameters* and activate the **OPEN** button.
4. Scroll down the parameter list until *netisr_priority* is in view. The *netisr_priority* is listed in the "Current Value" column.
5. Highlight *netisr_priority*.
6. Choose `Modify Configurable Parameter` from the "Actions" menu to open the Modify Configurable Parameter window.

7. Choose *Specify new value* and enter the new *netisr_priority* value in the *Value* field.

8. Activate the **OK** button to start the task.

9. Activate the **OK** button from the Messages window when the task is completed.

10. To exit SAM, choose `Exit` from the "List" menu to return to the Create a New Kernel window.

11. Create a new kernel and then activate the **Exit SAM** button.

Changing netisr_priority Manually

If you wish to run *netisr* at priority 100 and change this value manually, enter the following line in the appropriate configuration file (*S800* or *dfile*):

```
netisr_priority  100
```

If you change *netisr_priority*, the new value will take effect when the system is rebooted. If *netisr* is running as a process, you can change its priority between 1 and 127 with the *rtprio(1M)* command.

5

Configuration Commands

This chapter describes LAN/9000 configuration commands. It contains the following sections:

- Overview of LAN Configuration Commands.
- *eisa_config(1M)*.
- *ifconfig(1M)*.
- *lanconfig(1M)*.
- *route(1M)*.
- *subnetconfig(1M)*.

Overview of LAN Configuration Commands

LAN/9000 provides the following configuration commands.

eisa_config(1M)	**Series 700 only.** Interprets information stored in configuration files and uses it to configure system resources needed to properly interact with EISA cards on Series 700 workstations.
ifconfig(1M)	Assigns an address to a network interface and configures network interface parameters.
lanconfig(1M)	Sets/resets the packet encapsulation method for a network interface.
route(1M)	Assigns the host or network where packets will be routed.
subnetconfig(1M)	Sets/resets the value of the internal flag which controls the subnet behavior of a host.

eisa_config(1M)

The `eisa_config` utility is a specialized program for configuring EISA cards on HP-UX Series 700 systems equipped with EISA backplanes. This utility is used only on Series 700 systems.

Syntax

```
eisa_config [-c cfgfile] [-n scifile]
```

Parameters

`-c` *`cfgfile`*
: Checks the configuration (CFG) file to verify that it follows correct grammar and can be used by `eisa_config`. This option does not affect current configuration in any way.

`-n` *`scifile`*
: Uses the contents of *scifile* instead of non-volatile memory (NVM) to set up EISA configuration and is most commonly used for creating identical configurations on multiple systems.

Description

`eisa_config` is a specialized program used to configure EISA I/O cards on Series 700 workstations equipped with EISA backplanes. You must use it whenever the EISA configuration is changed, such as when cards are added, removed, or moved to a different location on the system. `eisa_config` should be run before any physical card configuration or installation changes are made.

EISA cards usually have no switches or jumpers for resource assignment. Instead, each EISA card has a corresponding configuration (CFG) file that tells the system how the card can be used and what resources it needs. `eisa_config` is the HP-UX system program that interprets the various CFG files for all cards in the system, and then builds a conflict-free configuration. Even though they may be physically present in the computer, EISA cards cannot be used by the HP-UX operating system until configuration by `eisa_config` is complete.

Refer to the *eisa_config* manual page for detailed information on command syntax and CFG file requirements.

Example

To add the CFG file to the EEPROM memory chip on the EISA interface for the card in slot 1 of your Series 700 system, follow the steps below:

1. Run the `eisa_config` utility in interactive mode using the following command:

   ```
   /etc/eisa_config
   ```

   The screen display shows a slot number, configuration (CFG) file, and contents description for each EISA card.

Note If you inserted the hardware card into the EISA interface prior to installing the LAN/9000 software using the `update` software using the `update` utility, `eisa_config` may automatically recognize the card. If so, you should proceed to Step 4 below.

---

2. Verify that `eisa_config` displays either !HWPC000 at slot 0 for the S720/730 EISA system card or !HWPC010 at slot 0 for the S750 EISA system card.

3. Add the LAN card configuration file (!HWP1850) to the EEPROM memory chip on the EISA interface, if it is not already added, using the *add* command followed by the number of an empty slot for the card, after the EISA prompt. HP recommends that you use the lowest empty slot.

   In the example below, 1 is the slot number of the EISA interface:

   ```
   add !HWP1850 1
   ```

   The addition of this card can cause previous card configurations to change. This situation should be handled as described in the E/ISA configuration documentation and in conjunction with any specific product installation manuals corresponding to those cards.

4. Execute the *show board* command followed by the slot number of the card to display the card's basic attributes. In the example below, 1 indicates the slot number of the EISA interface.

   ```
   show board 1
   ```

5. To save the new card configuration, execute the command:

   ```
   save
   ```

   No switches or jumpers have to be changed.

6. To leave the `eisa_config` utility, execute the command:

   ```
   quit
   ```

   Upon exiting `eisa_config`, a list of required steps will appear on the screen.

   For the S700 networking cards, steps 1 and 2 on the list do not apply. Detailed hardware installation instructions for steps 3 through 6 are included in the next section.

ifconfig(1M)

The *ifconfig(1M)* command assigns an address to a network interface, and configures and displays network parameters.

Syntax

```
ifconfig  interface  [address_family]  [address|hostname] [parameters]
```

Parameters

interface

Specifies 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 by the order of the cards in the slots of the backplane.

You can use the *lanscan(1M)* command to display the string and unit of each networking interface that is bound to the system.

address_family

Specifies the address format used to interpret addresses. The only address family currently supported is *inet* (DARPA-Internet family).

address|*hostname*

Specifies the address as either a host name present in the host name database, hosts(4), or a DARPA Internet address expressed in dot notation.

up

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

`down`	Marks 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.
`arp`	Enables the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default).
`-arp`	Disables the use of the Address Resolution Protocol.
`metric` *n*	Sets 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 additional hops to the destination network or host.
`debug`	Enables driver dependent debugging code; usually, this turns on extra console error logging.
`-debug`	Disables driver dependent debugging code.
`netmask` *mask*	Specifies 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.
broadcast	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.

Description

The *ifconfig* command 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.

You can also use the *ifconfig* command to display the current configuration for a network interface when no optional parameters are supplied.

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

Note When troubleshooting LAN, follow the steps outlined in chapter 3 in sequence, using this command description as secondary reference information, until the problem is diagnosed and resolved.

Example

To assign the Class C IP address *192.6.1.17* and the subnet mask *255.255.255.240* to the network interface *lan*1, issue the following *ifconfig* command:

```
ifconfig lanl 192.6.1.17 netmask 255.255.255.240 up
```

lanconfig (1M)

The *lanconfig* command sets/resets the packet encapsulation method for a network interface.

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 ether
```

route(1M)

The *route(1M)* command adds and deletes entries to the network routing table.

Syntax

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

Parameters

`-f` Specifies that *route* will "flush" the routing table of route table entries that specify a remote host as a gateway. If this is used with one of the commands described below, the tables are flushed before the command is performed.

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.

`host` Specifies that *destination* is 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 *HP-UX 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 *HP-UX Reference Pages* for details on internet "dot" notation.

count Integer indicating whether the gateway is a local interface or a remote system. If *count* is greater than 0, the gateway is a remote system. If *count* equals 0, the address is an interface on the local host. Default: 0.

Description

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 can make additional entries at installation time using *route(1M)* in the */etc/netlinkrc* file (refer to "Editing and Installing the /etc/netlinkrc File" in Chapter 4).

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

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 *HP-UX Reference Pages.*

Example

```
/etc/route   add   default   site-gwy   1
```

subnetconfig(1M)

The *subnetconfig* command sets/resets the value of the internal flag which controls the subnet behavior of a host.

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 or 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 subnet map shown in figure 9-8 in chapter 9, "Network Addressing," all the subnets on the 192.6.12 network are considered local subnets by default. On Host A, if the system administrator configures remote subnets using the *subnetconfig* command, then only the hosts belonging to subnets 192.6.12.64 and 192.6.12.128 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 less overhead at the gateways.

Example

```
subnetconfig  remote
```

6

Network Diagnostic Commands

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

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

landiag(1M)	Resets or reports status of the LAN card.
lanscan(1M)	Displays information about LAN cards that are successfully bound to the system.
linkloop(1M)	Verifies network connectivity through the Data Link Layer (OSI Layer 2).
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).

Note For Series 600/8X2 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 and may only be used by HP support personnel and those customers with the appropriate class license for the systems specified by the license. *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.

landiag(1M)

The *landiag(1M)* command allows you to diagnose and correct problems with the LAN card. Using *landiag(1M)*, you can:

- Reset the card.
- 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)*.

Note The default device file is always */dev/lan0*. Use the *name* command in LAN Interface Test mode to change the default, if necessary.

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 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 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 LAN Interface 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.

     lan      =  LAN Interface Diagnostic
     menu     =  Display this menu
     quit     =  Terminate the Diagnostic
     terse    =  Do not display command menu
     verbose  =  Display command menu

Enter command:
```

The *lan* command invokes the LAN Interface Test Mode.

Following is a description of each Test Selection Mode command.

LAN Command

Causes *landiag(1M)* to enter the LAN Interface Test Mode. The LAN Interface Test Mode is described later in this chapter.

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.
```

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 *landiag(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. The default device file is always */dev/lan0*. Use the *name* command to reset the device file to another device file identifier.

```
LAN Interface test mode. LAN Interface device file = /dev/lan0

     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 (default is */dev/lan0*). Use the *name* command to set the device file to a different device file identifier. 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 Appendixes C and D. Use the *Name* command to change the LAN interface device and display status information about the new device file.

As soon as *landiag* reads the lu number on the device file, the display status header for the LAN interface card is printed. For example:

```
                LAN INTERFACE STATUS DISPLAY
                  Thu, Nov 21, 1991    11:16:22

Device file                          = /dev/lan1
Lu number                            = 1
```

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.

```
                LAN INTERFACE STATUS DISPLAY
                 Thu, Nov 21, 1991    11:16:22

Device file                          = /dev/lan1
Lu number                            =  1
Current state                        = FAILED !!!
Unable to read failure code.
errno = 22
```

- 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.

This example shows the display for a LAN interface card in the active state on a Series 300/400 system.

```
                    LAN INTERFACE STATUS DISPLAY
                     Thu,Nov 21,1991     11:16:22

Device file                       = /dev/lan1
Select code                       = 21
Current state                     = active
LAN station address, hex          = 0x080009090ABC
Number of multicast addresses     = 0
Frames received                   = 654006
Frames transmitted                = 654033
Undelivered received frames        =7
Untransmitted frames              = 7
CRC errors received               = 0
Transmit collisions               = 1528
One transmit collision            = 68
More transmit collisions          = 730
Excess retries                    = 0
Deferred transmission             = 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
```

After the status display fills the screen, it prompts you with the message:

```
PRESS enter to continue
```

Press **[Return]** to look at the remaining statistics.

Refer to appendix D for screen displays on Series 800 and Series 700 systems. This appendix also includes descriptions of each field in the displays.

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/lan0*. *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/lan0:
```

You have the following options. You can:

- 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.

lanscan(1M)

The *lanscan (1M)* diagnostic displays the following information about LAN devices that are properly bound to system I/O services:

- Hardware Path.
- Station Address.
- Device *lu*.
- Hardware State.
- Network Interface Name, Unit, and State.
- Network Management ID.
- Encapsulation Methods.
- Major Number of the Device File.

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 network interface cards that are successfully bound to the system. If a network interface 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 and the last

category are specific to the network interface card, and the other four categories are specific to the network interface associated with the card.

Following is a listing of the possible network interface state and hardware state combinations along with explanations about changing the interface state using the *ifconfig(1M)* command:

- If the current network interface state is DOWN and the current hardware state is UP, use the *ifconfig(1M)* command to bring up the network interface.
- If the current hardware state changes to UP while the system is running, the network interface state will remain DOWN until it is brought up with the *ifconfig(1M)* command.
- If the current hardware state is UP, the network interface state can be changed to either UP or DOWN using the *ifconfig(1M)* command
- If the hardware state is DOWN, the network interface state will be brought down by the system.

Note *lanscan(1M)* does not display information about LAN devices that do not have software support such as LAN interface cards that fail to bind properly at boot-up time.

Examples

The examples below include different usages of the *lanscan(1M)* command as well as sample output on each type of system. Refer to chapter 10, "LAN Device and Interface Terminology", for more detailed information about the fields included in the screen display.

Example 1

The following command executes the *lanscan(1M)* command on the system */hp-ux.text* file.

```
/etc/lanscan  /hp-ux.text
```

Example 2

The following command executes the *lanscan(1M)* 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

These examples show displays from the *lanscan(1M)* command on the Series 700, the Series 600/800 (both HP-PB and CIO), and the Series 300/400.

Series 700:

```
/etc/lanscan

Hardware  Station          Dev  Hardware  Net-Interface    NM  Encapsulation    Mjr
Path      Address          lu   State     NameUnit State   ID  Methods          Num
2.0.2     0x0800090190E8   0    UP        lan0     UP      4   ETHER            52
4.1.0     0x08000909054A   1    UP        lan1     UP      5   ETHER            52
```

Series 800 (HP-PB systems):

```
/etc/lanscan

Hardware  Station          Dev Hardware  Net-Interface    NM  Encapsulation    Mjr
Path      Address          lu  State     NameUnit State   ID  Methods          Num
32        0x080009116F21   0   UP        lan0     UP      1   ETHER            51
48        0x080009002678   1   UP        lan1     UP      2   ETHER IEEE802.3  51
```

Series 800 (CIO systems):

```
/etc/lanscan

Hardware  Station          Dev Hardware  Net-Interface    NM  Encapsulation     Mjr
Path      Address          lu  State     NameUnit State   ID  Methods           Num
2/4.6     0x080009116F21   0   UP        lan0     UP      1   ETHER IEEE802.3   50
```

Series 300/400 (DIO systems):

```
/etc/lanscan

Select  Station          Dev  Hardware  Net-Interface    NM  Encapsulation     Mjr
Code    Address          lu   State     NameUnit State   ID  Methods           Num
21      0x080009116F21   0    UP        lan0     UP      1   ETHER IEEE802.3   --
```

Following is a description of each field in the display:

Field Name	Description
Hardware Path	Series 700 only. In the example, 2.0.2 indicates the IO module ID is 2, the slot number into which the card is inserted is the built-in LAN, and the functional ID of the card is 2. Series 800 non-8X7 systems only (HP-PB). In the example, 32 indicates the LAN card is in hardware module 8. Series 8X7 systems will have a hardware address such as 44.1 or 52.1. Series 800 (CIO) only. In the example, 2/4.6 indicates the bus converter is in hardware module 2, the CIO channel card is in hardware module 4 on the converter and the CIO LAN card is in slot 6 of the module.
Select Code	Series 300/400 only. Specifies the value of the dipswitch setting on the LAN card. In the example, 21 indicates the DIO LAN card on the mother board.
Station Address	Unique 12-digit hexadecimal address stored in the NOVRAM chip on the LAN card.
Device lu	Device logical unit associated with a LAN card. Series 700: The lu number is the same as the interface unit number. Series 800: This is equivalent to the manager index assigned to the card by the IO subsystem. On Series 800 computers the Device lu does not necessarily match the Network Interface Unit. Series 300/400: The lu number are assigned according to the order of the select codes on the network cards.
Hardware State	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.

Net-Interface NameUnit State	Network interface associated with a LAN card. It always has the name *lan*. The network interface unit is assigned according to the order in which the LAN cards are detected by the IO system. On Series 700s, the LAN interface controller on the Core IO card is detected first and is assigned an interface unit value of 0 and EISA cards have interface unit numbers ranging from 1 to 4. On Series 800s, the network interface unit of the first LAN card may be either 0, if the Series 800 system was not ordered with the 802.3 built-in card option, or 1. On Series 300/400s, the network interface unit is determined by the relative order of the Select Code. 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. You may configure the network interface state with the *ifconfig(1M)* command.
NM ID	Specifies a unique ID assigned by the system for the network management of each network interface.
Encapsulation Methods	Specifies the configured encapsulation method for a network interface. For LAN/9000 it will be ETHER, IEEE802.3, or ETHER IEEE802.3.
Mjr Num	Major number of the LAN device file. A "--" in this field indicates that a major number does not apply to this LAN device. The major number of Series 8X7 systems is 32.

linkloop(1M)

The *linkloop(1M)* diagnostic allows you to run Link Layer (OSI Layer 2) loopback tests between HP 9000 computers. The default device file is always */dev/lan0*.

Syntax

```
linkloop [-n count] [-f devfile] [-t timeout] [-s size] [-r rif]
[-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, press **[CTRL]-C**. The default value for *count* is one.

`-f devfile`

Specifies which device file to use. The device file must be a LAN device file. If no device file is entered, *linkloop* uses */dev/lan0* 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 data message to send. The maximum data size is dependent on the type of LAN link which is being used. The default value is the maximum data byte count that can be used for the particular link.

`-r rif`

Used as routing information field on token ring networks. This information allows a user to specify the particular bridge route over which token ring packets should be delivered. The *rif* value is given as an even number of hexadecimal bytes separated by colons, up to a maximum of 16 bytes.

`-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. If there are header of length errors, appropriate messages are 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 via the specified device (/dev file). The link level address of a remote computer can be found on the network map or worksheet, or by executing the *lanscan* command on the remote computer. 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 most 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. This parameter is required.

Description

linkloop(1M) uses link-level test frames to check connectivity within a local area network. It tests only link-level connectivity.

Example

To test the local computer's connectivity to a remote computer with the station address 0x080009000222 over */dev/lan1*, enter the following command:

```
linkloop   -f  /dev/lan1   0x080009000222
```

If the test is successful, the following message is displayed:

```
Link Connectivity to LAN station: 0x080009000222    -- OK
```

If an incorrect link address is entered as the *linkloop(1M)* parameter or if the link level test fails, the following error message is displayed:

```
Link Connectivity to LAN station: 0x1000900000F3    -- FAILED
```

linkloop(1M) can be aborted with the interrupt signal. The interrupt signal is entered by pressing the **[CTRL]-C** key. If *linkloop(1M)* is aborted, the current results are displayed.

netstat(1)

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

Syntax

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

Parameters

`-A` Lists the address of any protocol control blocks associated with sockets. Used for debugging. When used with the *-a* option, includes server processes. When used with the *-n* option, displays host addresses and port numbers numerically. See Example 2.

`-R` Lists all socket names in the NetIPC socket registry. Refer to the *NetIPC Programmer's Guide* for details. Used to display NetIPC information. See Example 7 below.

`-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.

`-I interface`	Shows information about this interface only. This option is used with *interval* as shown below.
`-n`	Displays network and host addresses as numbers.
`-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 *HP-UX Reference Pages*.
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 since the last line was printed. Every 24th line contains a summary of the statistics since the node was last powered up. Default: one sampling.
system	Kernel you wish to examine. Default: `/hp-ux`.
core	Kernel memory you wish to examine. Default: `/dev/kmem`

Description

netstat(1) reports network and protocol statistics regarding packet traffic and the local networking 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*, *-i*, *-m*, *-I*, and *-r* cannot be used in combination. If more than one option is specified, the precedence is; *-m*, *-I*, *-i*, *-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 database */etc/hosts*, or *network.port* if a socket address specifies a network but no host. The network address is found in the database */etc/networks*.

If the *-n* option is specified, the address is printed in internet 'dot' 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)

Using the *-i* option causes the state of the network interface to be shown as below.

```
netstat -i

Name   Mtu    Network       Address    Ipkts   Ierrs   Opkts   Oerrs   Coll
lan0   1500   192.6.142.1   node-142   10343   0       4134    0       0
lo0    1536   loopback      localhost  0       0       0       0       0
```

Following is a description of each field in the display:

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, you do not need to be concerned about this number.
Network	The network address of the LAN station 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.
Coll	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 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 3, Flowchart 2, to troubleshoot the interface.

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

Using the *-An* option causes the numeric representation of addresses for any protocol control blocks associated with sockets to be shown as displayed below.

```
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
```

Using the *-a* option causes the causes the state of all sockets to be shown as displayed below.

```
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
udp     0        0        *.who           *.*
```

Using the *-an* option causes the state of all sockets to be shown, with local addresses displayed numerically, as displayed below:

```
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
```

Following is a description of each field in the display:

Field Name	Description
PCB	

The address of any Protocol Control Blocks. Displayed only when the *-A* option is specified, displayed in hexadecimal.

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).

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* database 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 ARPA Services *telnet(1)*, *remsh(1)*, *rlogin(1)*, *rexec(1)*, *rwho(1)*, and *ftp(1)* if ARPA Services/9000 is installed.

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* database 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 has been created, but no address has been bound to it.
LISTEN	Socket is listening for connection requests.
SYN_SENT	Establishing a connection.
SYN_RECVD	Establishing a connection.
ESTABLISHED	Connection exists. Data can be sent to and read from a socket. Set for an active socket.
FIN_WAIT1	Local application has closed connection. Cannot send any more data. Can still receive data. Set during the graceful close of a connection.
CLOSE_WAIT	Remote application has closed connection. Can still send data. Will only receive queued data. Set during the graceful close of a connection.
FIN_WAIT2	Cannot send or receive data. Getting ready to close the connection. Set during the graceful close of a connection.
CLOSING	Cannot send or receive data. Getting ready to close the connection. Set during the graceful close of a connection.
LAST_ACK	Cannot send or receive data. Getting ready to remove the socket. Set after the graceful close of a connection.

TIME_WAIT	Idle period after closing a connection. This idle time guarantees that all information and signals have been received. Set after the graceful close of a connection.

Reporting Routing Information (Example 3)

Using the *-r* option causes gateway routing information to be shown as displayed below:

```
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 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* database 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* database 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 *HP-UX Reference Pages*.
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 *HP-UX Reference Pages* for details.

Reporting Memory Statistics (Example 4)

Using the *-m* option causes statistics recorded by the network memory management routines to be displayed as shown below:

```
netstat -m

Memory in use:
          49 socket structures in use
          75 protocol control blocks in use
           4 routing table entries in use
           3 interface address in use
Mbufs in use:
          29 mbufs allocated to data
          10 mbufs allocated to packet headers
           4 mbufs allocated to <mbuf type 18>

9 mapped pages in use
0 requests for memory denied
0 requests for memory delayed
0 calls to protocol drain routines
```

Reporting Protocol Statistics (Example 5)

Using the *-s* option causes the statistics for each protocol to be listed as shown in the display below:

```
netstat -s

tcp:
          5734 packets sent
                  3568 data packets (77397)
                  2 data packets (1486 bytes) retransmitted
                  2106 ack-only packets  (2037) delayed)
                  0 URG only packets
                  1 window probe packet
                  3 window update packets
                  54 control packets
          5539 packets received
                  2649 acks (for 48980 bytes)
                  11 duplicate acks
                  0 acks for unsent data
                  2649 packets (31922 bytes) received in-sequence
                  0 completely duplicate packets (0 bytes)
                  0 packets with some dup. data (0 bytes duped)
                  13 out-of-order packets (0 bytes)
                  0 packets (0 bytes) of data after window
                  0 window probes
                  6 window update packets
                  0 packets received after close
                  0 discarded for bad checksums
                  0 discarded for bad header offset fields
                  0 discarded because packet too short
          23 connection requests
          28 connection accepts
          51 connections established (including accepts)
          58 connections closed (including 32 drops)
          0 embryonic connections dropped
          3397 segmets updated rtt (of 3421 attempts)
          1 retransmit timeout
                  0 connections dropped by rexmit timeout
          0 persist timeouts
          0 keepalive timeouts
                  0 keepalive probes sent
                  0 connections dropped by keepalive
udp:
                  0 incomplete headers
                  0 bad data length fields
                  0 bad checksums
ip:
                  160116 total packets received
                  0 bad header checksums
                  0 with size smaller than minimum
                  0 with data size < data length
```

```
                0 with header length < data size
                0 fragments received
                0 fragments dropped (dup or out of space)
                0 fragments dropped after timeout
                912 packets forwarded
                3 packets not forwardable
                0 redirects sent

icmp:
                63 calls to icmp_error
                0 errors not generated 'cuz old message was icmp
                Output histogram:
                        echo reply:  1214
                        destination unreachable: 63
                0 messages with bad code fields
                0 messages < minimum length
                0 bad checksums
                0 messages with bad length
                Input histogram:
                        echo reply: 2
                        destination unreachable:  77
                        echo: 1214
                1214 message responses generated
arp:
                0 Bad packet lengths
                0 Bad protocol requests
                0 Bad headers
probe:
                0 Packets with missing sequence number
                0 Packets invalid length
                0 Packets with unsupported type
                0 Packets with bad sequence number
                0 Memory allocations failed
pxp:
                0 incomplete headers
                0 bad data length fields
                0 bad checksums
```

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.

Monitoring Packet Traffic (Example 6)

Using the *-I interface interval* option causes packet traffic statistics to be reported for the specified interface every *interval* seconds.

```
netstat  -I  lan0  5

input   (lan0)      output              input    (Total)     output
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
```

Following is a description of each field:

Field Name	Description
input packets	The number of packets received by the interface card.
(lan0) errs	The number of errors detected on incoming packets on the specified interface.
output packets	The number of packets transmitted by the interface card.
(Total) errs	The number of errors detected on incoming packets on all interfaces.
colls	The number of collisions that resulted from packet traffic.

Sending the interrupt signal, usually by pressing the **CTRL-C** 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. 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.

ping(1M)

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

Syntax

```
/etc/ping  [-r] [-v] [-o] host [packetsize ] [count]
```

Parameters

`-r` Used to bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (such as after the interface was dropped by *gated(1M)*.

`-v` Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

`-o` Inserts *record route* IP option in outgoing packets, summarizing routes taken when the program exits. It may not be possible to get the round-trip path if all hosts on the route taken do not implement the *record route* IP option. A maximum

host Specifies 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.

packetsize 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, causing the remote system to crash.

Range: 8 to 4096 bytes.

Default: 64 bytes.

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

Range: 1 to (2^{31} -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 **[CTRL]-C** 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 **[CTRL]-C** 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 **[CTRL]-C** 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 *linkloop(1M)* sends a packet to the Data Link Layer (OSI Layer 2) of a specific node.

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  192.20.20.106: icmp_seq=0. time=21. ms
100 bytes from  192.20.20.106: icmp_seq=1. time=20. ms
100 bytes from  192.20.20.106: icmp_seq=2. time=19. ms
100 bytes from  192.20.20.106: icmp_seq=3. time=18. ms
100 bytes from  192.20.20.106: icmp_seq=4. time=20. ms
100 bytes from  192.20.20.106: icmp_seq=5. 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 **[CTRL]-C** key, terminates the output.

Following is a description of each field in the display:

Field Name	**Description**
192.20.20.106	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]**.

Terminating Commands

Note The interrupt signal is often used to terminate diagnostic utilities. This chapter assumes you have set the **[CTRL]-C** 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 **[CTRL]-C**) to abort the current command, at any *rlb(1M)* prompt. When you press **[CTRL]-C**, *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 **[CTRL]-C**. 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 of 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.

     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 **[CTRL]-C**), 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 **[CTRL]-C**).
- 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 **[CTRL]-C**) to abort the operation.
- Press **[CTRL]-C** 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 **[CTRL]-C**), 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 **[CTRL]-C**), 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 **[CTRL]-C**) 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 *HP-UX 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 **[CTRL]-C**), 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 **[CTRL]-C**), 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 **[CTRL]-C**). 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 **[CTRL]-C**) 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 **[CTRL]-C**). 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 **[CTRL]-C**) 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.

LANDAD

LANDAD may only be used by HP support personnel and those customers with the appropriate class license for the systems specified by the license. *LANDAD* stands for Local Area Network Device Card 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 8x2 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

Logging and Tracing Commands

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 OTS/9000.

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 *netfmt* 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 places this command in */etc/netlinkrc* so that when you reboot the system the logging facility is started automatically. 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 */etc/conf/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.

A networking subsystem consists of a significant module of a networking product. Refer to appendix K for a complete list of LAN/9000 subsystems.

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 by */etc/netlinkrc*.

To Start Logging

The following command may be entered interactively at the terminal or 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
```

To stop logging, execute the following command:

```
nettl  -stop
```

For *nettl* command syntax, refer to the command reference description in the following section of this chapter.

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 *nettl* process that sends all log messages for that particular subsystem to a file or to the system console. Refer to appendix K for a complete list of LAN/9000 subsystem names for logging.

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* and */usr/adm/nettl.LOG01*. You can change the default files by editing the */etc/conf/nettlgen.conf* file. This file is described in the *HP-UX Manual Reference Pages*.

The *nettl* daemons always write to the *default_log.LOG00* file. When that file is full, the daemons copy the file contents to *default_log.LOG01* and purge the contents of

default_log.LOG00. If *default_log.LOG01* already exists, the contents are overwritten. If that file does not exist, it is created. A new *default_log.LOG00* file is then opened.

This process writes to the *default_log.LOG00* file continuously and copies to the *.LOG01* file while the *nettl* daemons are running. This procedure 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. You can change the maximum size by editing the */etc/conf/nettlgen.conf* file.

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.

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.

To View the Formatted Log Data

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 can 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     -f /usr/adm/nettl.LOG00
```

This command causes *netfmt* to format the default log file, */usr/adm/nettl.LOG00*.

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
```

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 used by other HP-UX products, 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.

To Start 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. Enter the *nettl* command with the traceon option interactively at the terminal. In this example, tracing is turned on for ingoing and outgoing packets for the *ns_ls_driver* subsystem, and the trace is sent to the *myfile.TRC0* file.

```
nettl  -traceon pduin pduout  -entity  ns_ls_driver   \
       -f /usr/adm/myfile
```

There is no default trace file; you must enter a trace file when you turn tracing on. To stop tracing on all subsystems, execute the following command:

```
nettl  -traceoff  -entity  ns_ls_driver
```

For *nettl* command syntax, refer to the command reference description in the following section of this chapter.

When tracing begins, two additional *nettl* daemons begin executing. If you subsequently issue a *ps -ef* 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.

To trace 802.3/ethernet packets, use the *ns_ls_driver* subsystem, and to trace loopback packets use the *ns_ls_loopback* subsystem.

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.TRC0* and *filename.TRC1*.

The *nettl* daemons write to the *filename.TRC0* file. When that file is full, the daemons copy the file contents to the *filename.TRC1* file and purge the contents of the *filename.TRC0* file. If the *filename.TRC1* file already exists, the contents are destroyed. If that file does not exist, it is created. A new *filename.TRC0* is then opened.

The process of writing to the *filename.TRC0* and copying to the *filename.TRC1* continues for the duration of the trace. This procedure insures that the oldest trace data on the system is always in the *.TRC1* file, and the latest trace data is always in the *.TRC0* 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 or 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 or reduce the number of packet bytes being traced using the *-m length* 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.

To View the Formatted Trace Data

You can use the *netfmt* command to view the trace data. This command formats the data in a readable form that is suitable for viewing at a terminal screen or printing. The trace files can 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 by using the filter files described in later in "Filter Command Lines."

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.

```
nettl -tn pduin pduout -e ns_ls_driver | netfmt -c \
lantrace_filters -N > fmt0
```

The command above causes the *nettl* daemons to collect *pduin* traces and *pduout* traces from the *ns_ls_driver* 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 "nicely" filtered and formatted record to the file, *fmt0*. The filters are specified in the *lantrace_filters* file.

The *nettl* command itself does not generate any trace output. The trace output is generated by the *netl_reader* process. The *ntl_reader* is executed as a background process if an output file is specified; otherwise the *nettl* command is replaced by the *ntl_reader*.

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 used by other HP-UX products, 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 [kind...]  -entity subsystem [subsystem...]...
          ...[-file filename] [-size limit] [-tracemax maxsize] [-m length]
          -traceoff  -entity subsystem  [subsystem...]
          -log  class [class...]   -entity subsystem [subsystem...]
```

Options

`-start`

Starts the *nettl* daemon, initializes the tracing and logging facility, and enables logging for all subsystems. The *nettl* daemons run in the background and maintain the network tracing and logging system. Log messages are sent to the file named *default_log.LOGxx*. The suffix *xx*, 00 or 01, will be appended onto the filename. The default logging class for LAN/9000 subsystem names is `error` and `disaster` or 12. Refer to appendix K for a listing of LAN subsystems. See the *-log* option.

This option may be abbreviated as `-st`. It is used alone without other options.

Note HP strongly recommends 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.

NOTE: HP strongly recommends that the tracing and logging facility not be turned off, as information about disasters will be lost. To minimize the impact on the system, all subsystems can be set to capture only *disaster* class log messages.

This option may be abbreviated as `-sp`. This option is used alone without other options.

`-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. *info* is optional. 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`. This option is used alone without other options.

`-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 a series of keywords or a 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
all	0xffffffff	All Possible Trace Kinds
loopback	0x00800000	Packet looped back at the driver level because it is intended for the same host.

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).

This option may be abbreviated as -tn. This parameter requires the -entity option. Other options are recognized but not required.

-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. This parameter requires the -entity option.

-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.

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 series of keywords or a numeric mask. The default logging classes are ERROR and DISASTER (12). 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 shutdown.

This option may be abbreviated as `-l`.

`-file` *name*

Creates a file with the *name* specified and a suffix of *.TRCx* (where x = 0 or 1) when tracing is initialized. 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. If the trace buffer is not large enough to handle all incoming trace records, trace records can be lost. You should use this option when you wish to stop dropped packets. 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. A number between 50 and 100 bytes is enough to capture the packet header. The default is the entire packet. This option may only be used with the *-traceon* option.

`-tracemax maxsize` — Specifies the maximum size of both trace files (*filename.TRC0* and *filename.TRC1*) combined. *maxsize* stands for the number of kbytes the combined size may be. The default size is 1000. The range is from 1 to 99999. This option may only be used with the *-traceon* option.

This option may be abbreviated as `-tm`.

`-entity subsystem [subsystem]` — Specifies the subsystem(s) the other options are acting on. Some of the subsystems for LAN/9000 are:

```
NS_LS_TCP     NS_LS_IP

NS_LS_PROBE   NS_LS_NFS

NS_LS_NFT
```

This option may only be used with the *-traceon* or *-log* option. This option may be abbreviated as `-e`. To obtain a list of all LAN subsystems, refer to appendix K. To obtain a complete list of all subsystems on the system, run *nettl -ss ALL*.

CAVEATS

Tracing or logging to a file may not be able to keep up with a busy system, especially when extensive tracing information is being gathered. If some data loss is encountered, the trace buffer size may be increased using the *-size* option. Be selective about the number of subsystems being traced as well as the log class messages being captured.

The *nettl(1M)* and *netfmt(1M)* commands read the */etc/conf/nettlgen.conf* file each time they are issued. If the file becomes corrupted these commands will no longer be operational.

Files

/etc/conf/nettlgen.conf	Tracing and logging subsystem configuration file.
/usr/adm/conslog.opts	Default console logging options filter files as specified in */etc/conf/nettlgen.conf*.
/usr/adm/nettl.LOG00	Default log file as specified in */etc/conf/nettlgen.conf*.
/dev/nettrace	Kernel trace pseudo-device file.
/dev/netlog	Kernel log pseudo-device file.

Refer to the following HP-UX manual reference pages for more detailed information: *nettl(1M)*, *netfmt(1M)*, *nettlconf(1M)*, and *nettlgen.conf(4)*.

netfmt(1M)

Formats common tracing and logging binary files.

Syntax

```
netfmt [-I subsys_file] [-c config_file] [-p]
       [-t records] [-F] [-l] [-v] [-n] [-N]
       [-l [LT]] [-f input_file]
```

Options

`-I subsys_file` Specifies the file containing a description of all subsystems and the option processing and formatting functions to call the library that contains them. You may use this option to specify an alternate subsystem file configuration file if the original becomes corrupted. If omitted, the system reads the default file, */etc/conf/nettlgen.conf*, to provide this information.

`-c config_file` Specifies the file that contains formatter filter configuration commands. The *config_file* must be a complete pathname. If you omit `-c` and the file contains trace data, *netfmt* uses the *$HOME/.nettrc* file. If the file being formatted contains log data, *netfmt* uses the *$HOME/.netlogrc* file.

`-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. All other options are 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. Used for internal debugging of *netfmt*.

`-n` Shows network addresses and ports as numbers. Without this option, *netfmt* attempts to interpret these fields symbolically.

`-N` Enables "nice" formatting where Ethernet/IEEE802.3, SLIP, IP, ICMP, TCP, UDP, PXP, ARP, and Probe packets are displayed symbolically. All remaining user data is formatted in hexadecimal and ASCII.

`-1 (minus one)` Attempts to tersely format each traced packet on a single line. If -L and/or -T options are used, the output lines will be more than 80 characters long.

- `-T` Places a timestamp on tersely formatted packets. Used with the -1 (minus one) option.
- `-L` Prefixes local link address information to terse tracing output. Used with the -1 (minus one) option.

`-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, *.LOG0*X or *.TRC*X, must be included in the *input_file* specification.

Files

/etc/conf/nettlgen.conf	Default subsystem configuration file.
/usr/adm/conslog.opts	Default console logging options filter file.
$HOME/.nettrc	Default configuration file for trace data if no configuration file is given on the command line with the *-c* option.
$HOME/.netlogrc	Default configuration file for log data if no configuration file is given on the command line with the *-c* option.

Refer to the following HP-UX manual reference pages for more detailed information: *nettl(1M)*, *netfmt(1M)*, *nettlconf(1M)*, and *nettlgen.conf(4)*.

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, *$HOME/.nettrc* or *$HOME/.netlogrc*, 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 *.nettrc* or *.netlogrc*. 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.

Global Filtering

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 all subsystems 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.

Subsystem Filtering

Note The global filtering described above takes precedence over individual subsystem tracing and logging filtering described below

Subsystem filters are provided to allow filtering of data for individual subsystems or groups of subsystems. It is possible for each subsystem configured on the system to have an individual subsystem filter. Such a subsystem filter would have the subsystem name as the keyword and would be configured by the *nettlconf(1M)* command. Subsystem filters are valid only when the corresponding subsystems have been installed and configured on the system.

Below is a list of filter types, each associated with a particular protocol layer:

Table 7-1. LAN Subsystem Filters

Filter Layer	Filter Type	Description
Layer 1	dest	hardware destination address
	source	hardware source address
	interface	software network interface
Layer 2	ssap	IEEE802.2 source sap
	dsap	IEEE802.2 destination sap
	type	Ethernet type
Layer 3	ip_saddr	IP source address
	IP source address	IP destination address
	ip_daddr	
Layer 4	tcp_sport	TCP source port
	tcp_dport	TCP destination port
	pxp_sport	PXP source port
	pxp_dport	PXP destination port
	udp_sport	UDP source port
	udp_dport	UDP destination port
	connection	a level 4 (TCP, UDP, PXP) connection

Refer to the *netfmt(1M)* manual reference page for definitions of each filter type.

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 `ns_ls_driver` subsystem, (all types of tracing are enabled by OR'ing bit masks), and sends binary trace messages to file */usr/adm/trace.file*.

```
nettl   -tn  all   -e  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:

```
Trace Information:
 Trace File name:          /usr/adm/trace.file
 Uid:              0       Buffer Size:       32768
 Dropped Messages  0       Messages Queued:    0

 Subsystem Name            Trace Mask:
 ns_ls_ip                     0xffffffff
 ns_ls_driver                 0xffffffff
```

Example 5

This example stops tracing:

```
nettl  -tf  -e all
```

Example 6

The following command reads the file */usr/adm/trace_file.TRC1* from the binary data, uses the *conf.file* as the filter configuration file, and "Nicely" formats the output.

```
netfmt   -N   -f /usr/adm/trace_file.TRC1   -c conf.file
```

Example 7

The following command formats the last 50 records in the file */usr/adm/nettl.LOG00* (the default log file) and sends them to the file, *fmt0*.

```
netfmt    -f /usr/adm/nettl.LOG00   -t 50  >  fmt0
```

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  -f  /usr/adm/nettl.LOG00 > /dev/console
```

In the example above, DISASTER.ONLY contains:

```
#------------------------------------------------------------
#  SUBSYSTEM      REQUEST_TYPE      ARGUMENT1       ARGUMENT2
#------------------------------------------------------------

FORMATTER        filter           class            !*
FORMATTER        filter           class            DISASTER
```

Example 9

The following sequence of commands will start the logging utility, modify the log classes to include warning and informational messages for the LAN driver subsystem, and display the formatted data on the terminal screen.

```
nettl  -start
nettl  -log  w  i     -entity  ns_ls_driver
[Capture information]
netfmt   -f  /usr/adm/nettl.LOG00
```

Example 10

The following sequence of commands will start tracing of ingoing and outgoing packets, save the output in a file, turn tracing off after information about network data has been captured, and "nicely" display the trace information on the terminal screen using *conf.file* as the filter configuration file.

```
nettl  -start
nettl  -tn  pduin  pduout   -entity  ns_ls_driver  \
     -f /usr/adm/mytrace
[Send test packet]
nettl  -tf  -e  ns_ls_driver
netfmt  -N  -c  conf.file  -f  /usr/adm/mytrace.TRC0
```

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

```
Formatter  filter  type [!] {value | *}
```

Filter Types

type — Indicates a keyword such as 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*. Not valid for `device_id`, `process_id`, or `user_id`.

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 *formatter 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 shutdown.

`kind` Trace type or mask type. A mask is a hexadecimal representation of a (set of) trace kind(s). You may enter either a single keyword or mask as the *kind* value. The trace kinds and their corresponding masks are:

`hdrin`	`0x80000000`
`hdrout`	`0x40000000`

	pduin	0x20000000
	pduout	0x10000000
	loopback	0x00800000

`subsystem`

A complete list of subsystem names is available in appendix K. 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, giving all subsystems 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.
ss	stands for seconds from 00 to 59.
dddd	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

Formatter filter          time_from     10:31:58   11/23/88
Formatter filter          time_through  10:41:00   11/23/88
Formatter filter          class         !*
Formatter filter          class         INFORMATIVE
Formatter filter          subsystem     !*
Formatter 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     ARGUMENT2

Formatter filter          kind          !*
Formatter filter          kind          pduin
Formatter filter          subsystem     !*
Formatter filter          subsystem     ns_ls_driver
Formatter filter          process_ID    10289
```

Refer to the *netfmt(1M)* manual pages for examples of LAN subsystem filters.

8

Product Description

This chapter provides a description of the LAN/9000 product. It includes:

- Product Structure.
- Product Protocols and the OSI Model.

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 8-1. Types of LAN Cards

HP 9000 Model	LAN Card
Series 300/400 models	98171A (DIO) LAN Card
All Series 600 models	36967A-20C (CIO) LAN Card
Models 808, 815, 822, 832, 842, 852	36967A-20N (HP-PB) LAN Card
Models 817, 827, 837, 847, 857, 867, 877, 887, 897	J2146A (HP-PB) LAN Card
Model 890 1, 2, 3, 4 CPU	28639-60001 (HP-PB) LAN Card
All other Series 800 models	36967A-20C (CIO) LAN Card
Series 700 models (S720 w EISA adapter option, S730, & S750)	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 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 motherboard. For Series 600/800 models, it is a separate card. Series 700 workstations only have one Core IO (LAN) card.

Most HP 9000 models can accommodate additional "add-on" LAN cards for gateway use. For Series 600/800 models, the add-on cards provide LAN functionality only. 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, with the available slots, 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, disk, or CD-ROM, depending on your HP 9000 model. LAN/9000 software includes programmatic interfaces, network protocol modules and tools for LAN administration.

Programmatic Interfaces

HP *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: IEEE 802.3/Ethernet Driver, ARP, IP, UDP, PXP, Probe, and TCP. 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:

- ***nettl(1M)***: This utility allows you to log network events and trace record packets as they enter and exit the LAN driver.
- ***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 address information.
- ***lanconfig(1M)***: This utility allows you to change the packet encapsulation method for a network interface.
- ***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.
- ***arp(1M)***: This utility allows you to display and modify the ARP cache.

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. The *LANDAD* diagnostic is not available on S8X7 and S890 systems.

Product Protocols and the OSI Model

Table 8-2 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 8-2. 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	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	Ethernet/ IEEE 802.3	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-iX, 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. A TCP socket accepts data buffers up to 56 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 two nodes want to communicate via TCP, 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 proprietary 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 is an ARPA standard which 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.

9

Network Addressing

This chapter introduces network addressing concepts. It contains the following sections:

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

Networking Terminology

Following are descriptions of important networking terms.

Nodes

A *node* is a computer on the network. *Local node* (or host) refers to the computer 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* is a communication path through which messages can be sent and received. A hardware network interface has a hardware device associated with it, such as a LAN or FDDI card. A software network interface does not include a hardware device, for example the *loopback* interface. For every IP address instance, there must be one network interface configured.

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. A loopback interface can be configured for testing reasons even if the system is not connected to a network.

For a network interface associated with a LAN card, the network protocol, for example, 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.

On Series 600/800 systems, 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.

- In an example Series 600/800 CIO system, the system might have a CIO channel card in hardware module 4 and another in hardware module 8. If there are two LAN cards in slots 4 and 5 of the channel card in hardware module 4 and three LAN cards in slots 3, 9, and 10 in the channel card in hardware module 8, the hardware paths of these cards are 4.4, 4.5, 8.3, 8.9, and 8.10 and the network interface unit numbers are 0, 1, 2, 3, and 4, respectively.
- In an example Series 800 HP-PB system, the system might have three HP-PB LAN cards in hardware module 4, 6, and 8. The hardware paths of these LAN cards are 16, 24, and 32 (4 times the module number), and the network interface unit numbers are 0, 1, and 2 respectively.

Note On Series 600/800 systems, the network interface unit of a LAN card does not necessarily match its device logical unit (LU) number.

On Series 300/400 systems, 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.

On Series 700 systems, the network interface unit is assigned in the order in which the LAN cards are detected by the I/O subsystem. The LAN Interface Controller on the Core I/O card is detected first, followed by that on the EISA cards, if any. The network interface unit for the Core I/O 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 associated with a LAN card.

Gateway

A *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. Refer to node D in the network maps in figure 9-3 and figure 9-9 for examples of gateways. A gateway system has to have at least two network interfaces configured, one for each network to which it belongs.

Routing Table

Each node on the LAN has a routing table. A *routing table* contains information about the route to nodes on other LANs. The connections to other LANs are made through gateways. When additional gateways are added by using SAM or by editing the */etc/hosts* file, or network addresses change, the routing table must be updated.

Clusters

A *cluster* consists of two or more workstations linked together by a local area network but having only one root file system. From the point of view of the file system, all the machines appear as one system. From the point of view of processors and processing space, each machine is distinct.

The cluster *root server* (sometimes referred to as the cluster server, root server, or server) is the cluster node that has the root file system.

The cluster root server supports other workstations (the cluster clients), which may , but need not, have their own disks. All cluster clients boot, over LAN, from kernels residing in the cluster server's file system. Each client has its own kernel.

A cluster *client* that has no local disks is known as a diskless client or diskless node. A cluster client that has one or more local disks that other clients are sharing is known as an auxiliary sway server (or sway server), an auxiliary file server, or generically as an auxiliary server--depending on how the disks are being used.

All cluster nodes (the server and all the clients) have access to files on the cluster root server's and any auxiliary file server's disks.

For detailed information on HP-UX clusters, refer to *Managing Clusters of HP 9000 Computers: Sharing the HP-UX File System* for Series 700 workstations or *Managing Clusters of HP 9000 Computers: Sharing the HP-UX File System* for Series 300/400 workstations.

Network Addresses and Node Names

Several types of names and addresses are used in networking software. This can be confusing to first-time users. Table 9-1 illustrates which address type is used by each layer of the OSI model. A description of each address type and how it is used by LAN and the services which run on it follows in table 9-2. Refer to "Network and System Names" in chapter 4 for additional information on how these names are assigned.

Table 9-1. Network Addresses and the OSI Model

	OSI Layer Name	OSI Layer Function	Address Type Used
7	Application	network programs	hostname
6	Presentation	data interpretation	hostname, nodename
5	Session	connection control	socket address
4	Transport	end-to-end transfer	port address
3	Network	routing and switching	internet (IP) address
2	Data Link	data packaging and error detection	link level address
1	Physical	physical connection	link level address

Table 9-2. Network Address Types, Descriptions and Examples

Address Type	Description	Recorded in	Used By
link level address	Also referred to as *MAC 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 config; 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 (see network address description below) and which host it is (see host address description below). 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 *ifconfig* command.

Table 9-2. Network Address Types, Descriptions and Examples (Cont'd)

Address Type	Description	Recorded in	Used By
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. Refer to "Internet Addresses" in this chapter for a definition of Internet address classes. If the IP address 192.6.23.3 is Class C, then the network address portion is 192.6.23.	*/etc/networks*	Routing facility. Displayed by *netstat -in* and *netstat -rn.*
host address	Also, *host number.* 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. Refer to "Internet Addresses" in this chapter for a definition of Internet address classes. If the IP address 192.6.23.3 is Class C, then the host address portion is ... .3.	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 *ifconfig* command.

Table 9-2. Network Address Types, Descriptions and Examples (Cont'd)

Address Type	Description	Recorded in	Used By
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*. For example, if your local address is listed as 192.6.23.3.1023, then .1023 is the port address.	*/etc/services*	Service requests. Displayed by *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. Refer to the *sockaddr* struct in the *Berkeley IPC Programmer's Guide* for examples.	socket address variables	Interprocess communication.

Table 9-2. Network Address Types, Descriptions and Examples (Cont'd)

Address Type	Description	Recorded in	Used By
system name	Also referred to as the *system host name* and *system node name*. This is the name your HP-UX system is known by and is assigned using the HP-UX *uname* command. An example *system name* is: *host3*. Assigned automatically by the system.	*/etc/src.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: *host3*. Assigned by using the *hostname* command.	*/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: *host3.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. The host address identifies a node within the network. A network address is concatenated with a host address to form the internet address and to 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 9-1.

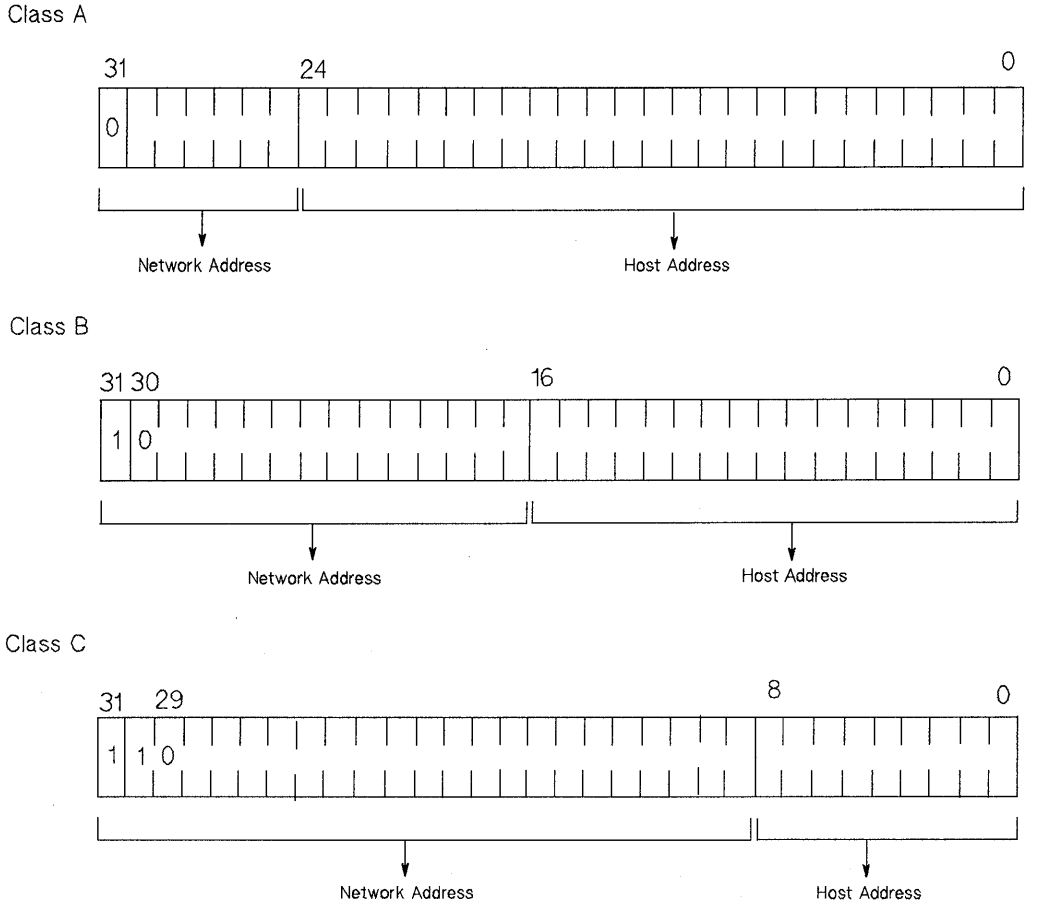


Figure 9-1. Internet Address Classes

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 9-3. Internet Address Classes

Class	Networks	Nodes per Network	Address Range
A	127	16777215	1.0.0.1 – 126.255.255.254
B	16383	65535	128.1.0.1 – 191.255.255.254
C	2097151	255	192.0.1.1 – 223.255.255.254
Reserved	–	–	224.0.0.0 – 255.255.255.254

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:

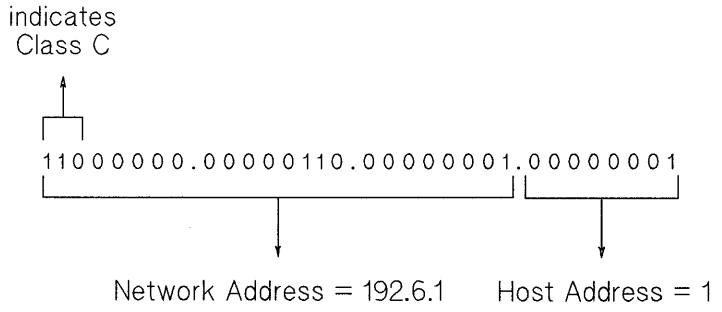


Figure 9-2. Bit Representation of Internet Address

Assigning an Internet Address

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

Note When specifying internet addresses, do **not** use leading zeroes within address fields. For example: 192.006.012.023 is **incorrect**; 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.
- If your system is attached to more than one physical network via a gateway, the network addresses of these interfaces may not be the same. Refer to figure 9-3 below for a gateway example.
- All nodes in the same network, however, must have the same network address.
- Do not assign the network addresses 0 or 255 (Class A), 0.0 or 255.255 (Class B), or 0.0.0 or 255.255.255 (Class C) to any network. Those addresses are reserved.
- Do not assign Class A network address 127. This address is reserved for the loopback interface.

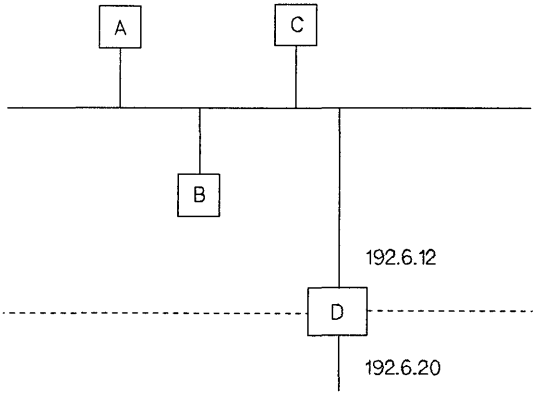


Figure 9-3. Assigning Network Addresses

Note To obtain class B and class C IP addresses, you must contact Government Systems, Incorporated (GSI). To obtain an application form, send an e-mail message to *service@nic.ddn.mil* or write GSI at the address below. Allow at least eight working days for GSI to process an IP address request.

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Chantilly, VA 22021
(800) 364-3642
(703) 802-4535

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 the host addresses 0.0.0 or 255.255.255 (Class A), 0.0 or 255.255 (Class B), or 0 or 255 (Class C) to any nodes; **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. When messages with subnet addresses are routed across the network, the internet address is AND'd with the subnet mask to determine the subnetwork address. (0 values in subnet mask convert corresponding bits in IP address to 0.)

Figure 9-4 shows how a class C example Internet address and example subnet mask combine to form a subnet address. Detailed descriptions of a subnet mask and each address class follow.

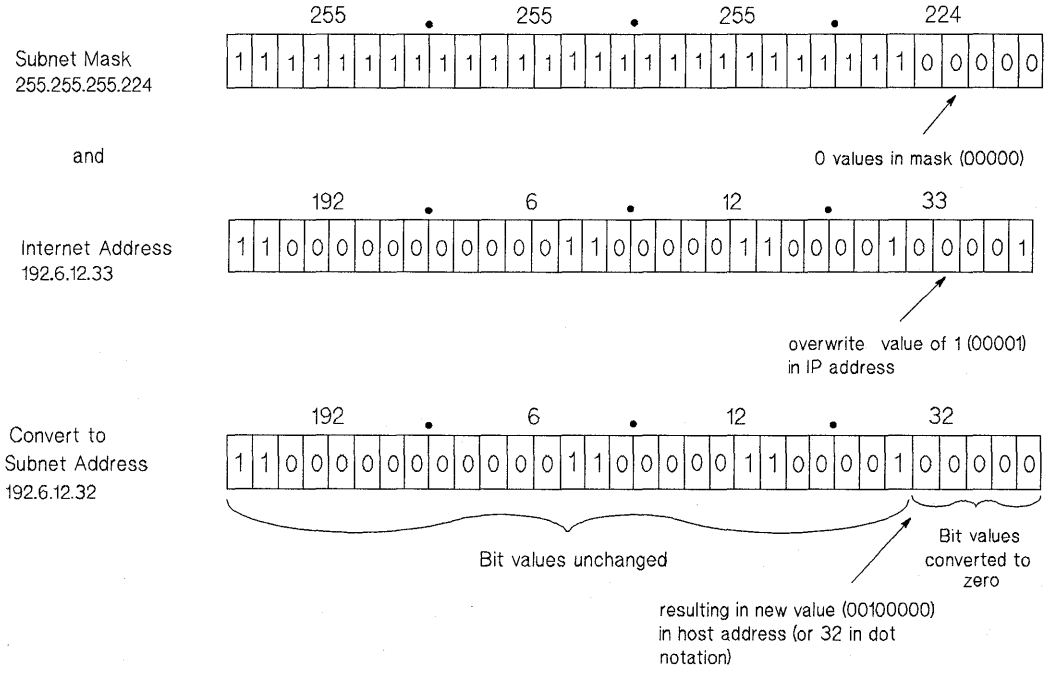


Figure 9-4. Internet Address 192.6.12.33 AND'd with Subnet Mask 255.255.255.224

When the internet address is AND'd with the subnet mask, the zero values in the host portion of the subnet mask will "overwrite" the corresponding bits of the host portion of the internet address and the resulting subnet address will be 192.6.12.32 as shown in figure 9-4 above. Non-zero values in the subnet mask indicate that the corresponding bits in the internet address do not change.

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.

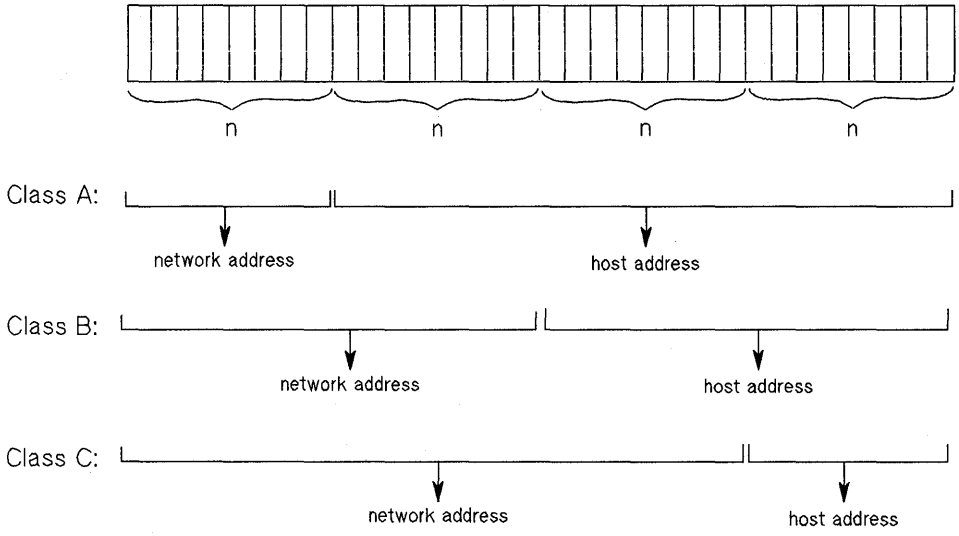


Figure 9-5. Internet Address Fields

The subnet address is based on the host address portion of the internet address. The host address portion subdivides into subnet number and host number fields to accommodate a given number of subnetworks and a given number of nodes per subnetwork. The size of the subnet number field is determined by the subnet mask, which is explained later in this section. The subnet number field must contain a minimum of two bits.

In the example below, the IP address, 192.6.12.33, has a subnet number of 1.

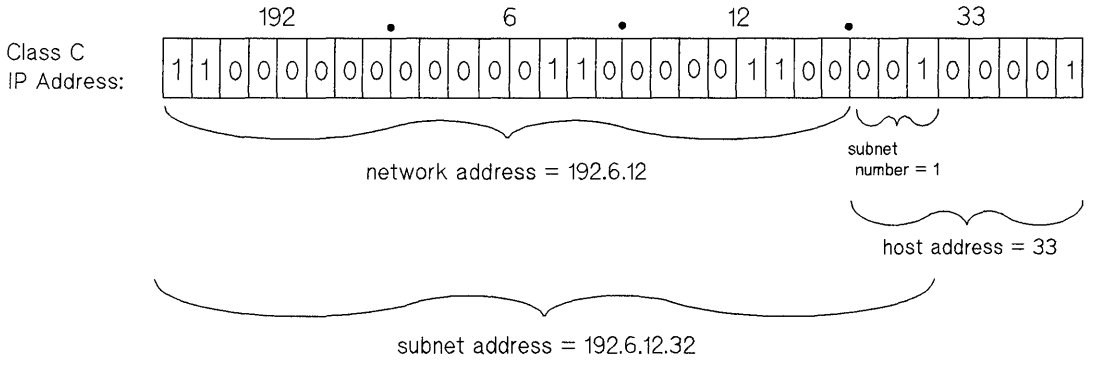


Figure 9-6. Subnet Address and Subnet Number of Class C Internet Address 192.6.12.33

Refer to figure 9-4 for an illustration of how the subnet number is AND'd with the IP address to form the subnet number.

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.
- If your system is attached to more than one physical network, the subnet addresses of the interfaces on your system may not be the same.
- Do not assign a subnet address where all the bits of the subnet number are 0 or all the bits are 1.

Using three of the eight bits of the host address of a Class C address for the subnet number, table 9-4 below lists the valid internet address ranges for up to 6 subnets and 30 nodes per subnet.

Table 9-4. Subnet Addressing

Class C internet address: n.n.n. xxx xx

Subnet Number (bitwise binary)	Subnet Address (dot notation)	Internet Address Range (dot notation)
000xxxxx	n.n.n.0	(subnet address not allowed)
001xxxxx	n.n.n.32	n.n.n.33 - n.n.n.62
010xxxxx	n.n.n.64	n.n.n.65 - n.n.n.94
011xxxxx	n.n.n.96	n.n.n.97 - n.n.n.126
100xxxxx	n.n.n.128	n.n.n.129 - n.n.n.158
101xxxxx	n.n.n.160	n.n.n.161 - n.n.n.190
110xxxxx	n.n.n.192	n.n.n.193 - n.n.n.222
111xxxxx	n.n.n.224	(subnet address not allowed)

Assigning Subnet Masks

Subnet addressing is implemented by specifying a 32-bit subnet mask in the *ifconfig* command when a LAN interface card is assigned an 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:

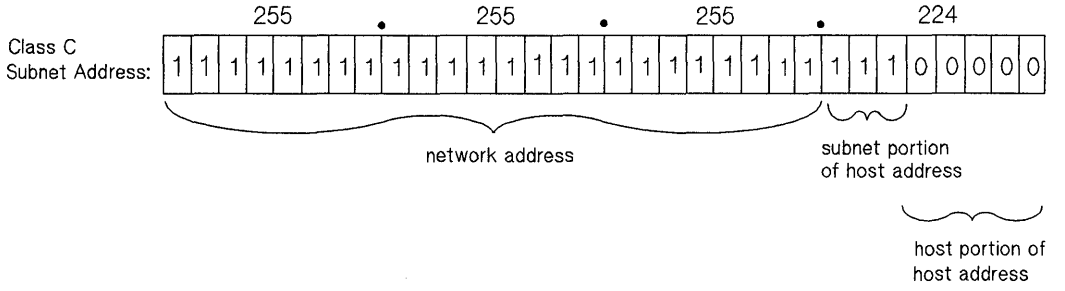


Figure 9-7. Subnet Mask

Example Subnets

The following example shows four subnetworks within the 192.6.12 network along with the *ifconfig(1M)* and *route(1M)* commands necessary to configure these subnetworks in the */etc/netlinkrc* file. The complete network map is shown in figure 9-9.

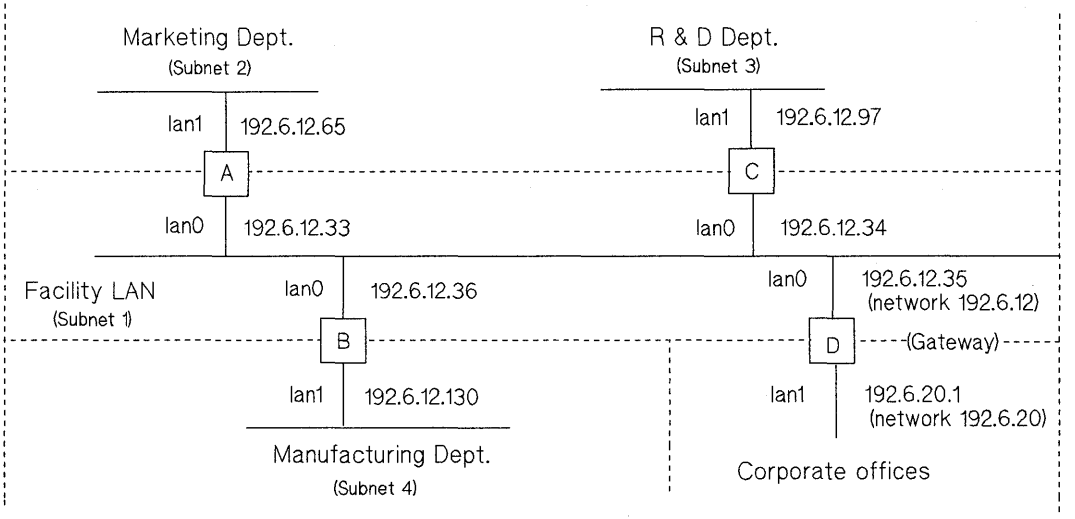


Figure 9-8. Network Map for Subnetting

Company division network = 192.6.12
  Subnet mask: 255.255.255.224

Facility LAN subnet number = 1
  Host address range: 33 to 62
  Host A internet address: 192.6.12.33 for network interface lan0
  Host B internet address: 192.6.12.36 for network interface lan0
  Host C internet address: 192.6.12.34 for network interface lan0
  Host D internet address: 192.6.12.35 for network interface lan0

Marketing Department subnet number = 2
  Host address range: 65 to 94
  Host A internet address: 192.6.12.65 for lan1

R & D Department subnet number = 3
  Host address range: 97 to 126
  Host C internet address: 192.6.12.97 for lan1

Manufacturing Department subnet number = 4
  Host address range: 129 to 158
  Host B internet address: 192.6.12.130 for lan1

Configuring Hosts on Subnetworks

To set the subnet masks, you include them in the *ifconfig* command in the */etc/netlinkrc* file 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 192.6.12.33 netmask 255.255.255.224
/etc/ifconfig lan1 192.6.12.65 netmask 255.255.255.224
```

Host B:

```
/etc/ifconfig lan0 192.6.12.36 netmask 255.255.255.224
/etc/ifconfig lan1 192.6.12.130 netmask 255.255.255.224
```

Host C:

```
/etc/ifconfig lan0 192.6.12.34 netmask 255.255.255.224
/etc/ifconfig lan1 192.6.12.97 netmask 255.255.255.224
```

Host D:

```
/etc/ifconfig lan0 192.6.12.35 netmask 255.255.255.224
/etc/ifconfig lan1 192.6.20.1
```

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

Configuring Gateways on Subnets

Besides using the appropriate subnet masks, each gateway needs to be configured so that it can properly route messages among the several subnetworks. Following are descriptions of two types of routing: *explicit* routing and *dynamic* routing. When using explicit routing, you must specify the IP address of each gateway to which you are directly connected. When using dynamic routing, you only need to specify the IP address of one gateway, and the system learns the IP address of other gateways from the specified gateway.

Explicit Routing

There are many ways to set up routing. For example, you might add the following *route(1M)* commands to the */etc/netlinkrc* file on Host A in figure 9-9:

```
/etc/route  add  net  192.6.12.128  192.6.12.36  1   # through Host B
/etc/route  add  net  192.6.12.96   192.6.12.34  1   # through Host C
/etc/route  add  net  default       192.6.12.35  1   # through Host D
```

The *1* in each entry specifies an indirect route. For example, messages for the system on the 192.6.12.128 subnetwork will first be sent to Host B (192.6.12.36), and from there they will be forwarded to the destination system.

Dynamic Routing

Alternatively, and perhaps the easiest way to manage growth on the 192.6.12 network, you might add the following entries to each */etc/netlinkrc* file.

Hosts A, B and C:

```
/etc/route  add  default  192.6.12.35  1   # through Host D
```

Host D (Site gateway):

```
/etc/route  add  net      192.6.12.64     192.6.12.33   1   # through Host A
/etc/route  add  net      192.6.12.128    192.6.12.36   1   # through Host B
/etc/route  add  net      192.6.12.96     192.6.12.34   1   # through Host C
/etc/route  add  default  192.6.20.1
```

If you add a new subnetwork to the Facility LAN at a later time, you will only need to add an appropriate routing entry on Host D. It will not be necessary to configure the other subnet gateways.

With this configuration, each subnet gateway (Hosts A, B, and C) will initially route messages for a system outside its subnet to Host D. The subnet gateway, however, will learn of the more direct routes automatically when Host D redirects the messages to one of the other subnet gateways. Subsequent messages for the destination system will be routed directly to the appropriate subnet gateway.

For example, referring to Figure 9-9, suppose messages are sent from system A1 (192.6.12.67) to system B1 (192.6.12.131). The first message will actually be routed to Host D (through Host A). Host D then will redirect the message through Host B. At the same time, Host D will notify Host A that Host B is a more direct route for messages to system B1. Subsequent messages to system B1 will be routed directly to Host B.

Redirected routes are called dynamic routes. You can see these dynamic routes by executing the command, *netstat -r*, on Host A. Dynamic routes are indicated in the display with a *D* flag.

Proxy ARP Server

The default direct route entry on Host D assumes that there is a *proxy ARP* server on the 192.6.20 network. If there is none, additional indirect route entries can be configured for each gateway that is directly connected to the 192.6.20 network.

For example, referring to figure 9-9, you might add the following indirect routes to send messages to Division 2 and Division 3.

```
/etc/route  add  net  192.6.14  192.6.20.2   # through Host Div2
/etc/route  add  net  192.6.13  192.6.20.3   # through Host Div3
```

Example Network Map

This sample network combines the networks, subnets, and clusters previously described and illustrated in this chapter along with a sample worksheet that provides configuration information necessary to attach these systems to the networks.

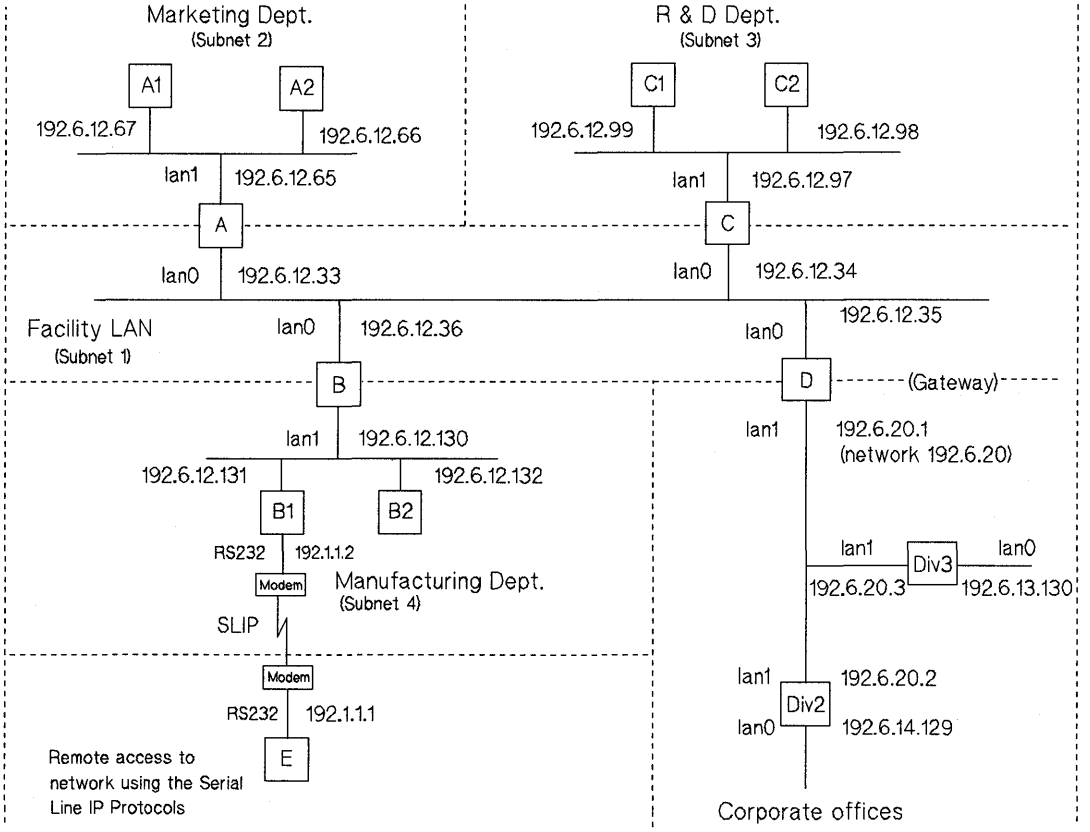


Figure 9-9. Network Map

Example Worksheet							
Hostname	**Interface Alias**	**Internet Address**	**Station Address**	**Hardware Path**	**Op Sys**	**9000 Model**	**Cnode Type**
A	mkt_32 mkt_64	192.6.12.33 192.6.12.65	08000909030D 080009080102	2.0.2 4.4.0	9.0	750	Server
A1	mkt_a1	192.6.12.67	080009005201	2.0.2	9.0	710	Client 1
A2	mkt_a2	192.6.12.66	080009001001	2.0.2	9.0	710	Client 2
B	mfg_32 mfg_128	192.6.12.36 192.6.12.130	080009005201 080009000C24	2.0.2 4.2.0	9.0	750	-
B1	mfg_b1 b1_slip	192.6.12.131 192.1.1.2	080009001001 NA	2.0.2 NA	9.0 9.0	720 720	-
B2	mfg_b2	192.6.12.132	080009002125	2.0.2	9.0	720	-
C	rd_32 rd_96	192.6.12.34 192.6.12.97	080009002108 0800090012AB	2.0.2 4.3.0	9.0	750	Server
C1	rd_c1	192.6.12.99	08000900079C	21	9.0	375	Client 1
C2	rd_c2	192.6.12.98	08000900601A	22	9.0	425	Client 2
D	div1_32 div1_gw	192.6.12.35 192.6.20.1	080009000740 080009000B30	44.1 52.1	8.02	837	-
Div2	div2_128 div2_gw	192.6.13.130 192.6.20.2	080009006041 080009007104	44.1 52.1	8.02	837	-
Div3	div3_128 div3_gw	192.6.14.129 192.6.20.3	080009004020 080009010312	44.1 52.1	8.02	837	-
E (SLIP)	e_slip	192.1.1.1	NA	NA	9.0	720	-

Subnet mask = 255.255.255.224

Figure 9-10. Network Map Worksheet

Clusters, Subnets, and LAN/9000

All nodes in a cluster must be connected on the same physical LAN. You can have more than one cluster per LAN, and computers that are not part of any cluster can also be on the cluster's LAN (we'll use the term standalone from now on to describe computers that are not part of any cluster).

Hewlett-Packard recommends that the cluster be on a small local LAN, with the root server acting as a gateway to other networks. In this way, there will be less contention and, therefore, better performance.

Note All Series 700 cluster clients must be connected to the cluster LAN via the built-in interface, not via the add-on EISA card.

10

LAN Device and Interface Terminology

Following is a description of terms used by the I/O subsystem to identify LAN cards and device files associated with LAN cards.

- Hardware Path.
- Select Code.
- Device Logical Unit.
- LAN Device Files.
- Network Interface.

Refer to these descriptions as necessary when completing LAN installation, administration and diagnostic procedures.

Hardware Path

On Series 600/800 and Series 700 systems, the I/O subsystem identifies each LAN card by its *hardware path.* The hardware path is assigned by the system according to the physical location (slot) of the card in the hardware backplane. Below are definitions of the hardware path on each system type.

Series 800 HP-PB Systems

To determine the hardware path of an HP-PB LAN card you multiply the location of the system bus slot number by 4. For example:

32 Specifies that the HP-PB LAN card is in hardware module 8.

Series 800 CIO Systems

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 card on the system bus. The second number, the slot number, is determined by the slot number of the CIO LAN card on the CIO card module. For example:

4.3 Specifies that the CIO card is located in hardware module 4 (also known as a slot) on the system bus and hardware slot 3 in the system backplane.

Series 700 Systems

For Series 700 systems, the hardware path is composed of three parts: an I/O module identifier, a slot identifier, and a card functionality identifier.

Core I/O card

The module identifier for a Core I/O card is always 2, the slot number for a Core I/O card is always 0 and the functionality identifier is either 2 for Core I/O cards or 0 for EISA cards. For example:

2.0.2 Specifies the hardware path of the Core IO (LAN) card.

Add-on EISA Card

For add-on EISA cards, the module ID is 4, the slot number is a value from 1 through 4, and the card functionality ID is always 2. For example:

4.3.0 Specifies the hardware path for an add-on EISA card in slot 3.

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. For example:

21 Specifies the DIO LAN card on the motherboard.

29 Specifies a second DIO LAN card.

The LAN card on a motherboard 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.

Use the *lanscan(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 LU number, such as 0, 3 and 2 respectively. The LU numbers of these cards are unique only within the larger CIO card category.

Series 600/800 Systems

On Series 600/800 systems, the logical unit of a LAN card is assigned by the I/O subsystem immediately after the system is booted up. The LU number may 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 and stores each LU number that has been assigned to a hardware path.

Note If a Series 600/800 system shuts down and a LAN card is removed, the LU number assigned to that card is 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 LU numbers of the LAN cards on a system may not be consecutive.**

A missing LU number, such as 1 in the example above, indicates that a LAN card that was configured during a previous system boot-up, was removed at a later time from the current hardware configuration.

Use the *insf(1M)* or *rmsf(1M)* command to change or remove an LU number.

For example:

a. If a system is booted up with a LAN card with hardware path 4.3, the system will assign lu0 to the card.

b. If the system is shutdown and the same LAN card is moved from hardware path 4.3 to 4.5, the system will assign lu1 to the LAN card when the system is rebooted.

lu0, which was assigned during a previous system bootup, **is still reserved for hardware path 4.3** although there no longer a LAN card on this path.

c. If the system is shutdown yet again and a second LAN card is installed on hardware path 4.3, the system will assign lu0 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 number is still 1.

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.

Series 700 Systems

On the Series 700, the system assigns the LU number in the order in which the LAN cards are detected by the I/O subsystem. The LU number is also the same as the interface unit number on these systems.

Series 300/400 Systems

On Series 300/400 systems, the system assigns LU numbers to LAN cards according to the value (lowest to highest) 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.

Note Series 700 systems do not recall cards assigned during a previous system boot-up.

Use the *lanscan(1M)* command to display the device LU assigned by the system to each LAN card that is successfully bound to the system when the system is booted-up.

LAN Device Files

The system uses 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) being used.

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

For example, three cards in a Series 800 CIO system might have the following device files:

```
ls -l /dev/lan* /dev/ether*

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
```

On Series 800 systems, as shown in the example above, two special device files are created for the device LU of each card when the system is booted up. For a LAN card with an LU number specified as *lux*, device files */dev/lanx* and */dev/etherx* are created automatically. If the device LU is displayed as a dash instead of as a numerical value, the system core dump occurred before the boot-up process was completed and no device LU was assigned by the system.

The device file number is composed of a major number and a minor number. In the example above, the major number is 50 indicates the CIO LAN driver. The minor number of this built-in LAN card, 0x000000, has a middle two-digit field indicating the device LU number (0 for the built-in LAN), and a final two-digit field indicating the data link protocol (0 for IEEE 802.3 or 1 for Ethernet).

Major Numbers

Table 10-1 below shows the major numbers of the LAN drivers on each platform:

Table 10-1. Major Numbers of LAN/9000 Drivers

System Type	Major Number
Series 8XX CIO (835, 840, 845, 859, 855, etc.)	50
Series 8XX HP-PB (808, 815, 832, 842, 852, etc.)	51
Series 8X7 HP-PB (817, 827, 837, 847, 857, 867, 877, 887, 890, and 897)	32
Series 300/400 IEEE	18
Series 300/400 ETHER	19
Series 700	52

Minor Numbers

The minor number is 24 bits wide and consists of various fields depending on the system type.

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, and the least significant 8-bit field indicates IEEE 802.3 (0) or Ethernet (1).

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).

Series 700: 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 10-2. Series 700 Device File Bit Structure

Bits	Contents
Bits 23-20	Contains the I/O 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 I/O 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)

The system follows certain conventions when creating LAN device files automatically. When the user creates device files manually, s/he must enter the correct major number and correct minor number but does not have to follow any other conventions.

Device files are used by Link Level Access users to access the LAN driver, and some network services and diagnostic tools.

To create LAN device files, use the *mknod(1M)* command. Additional information on device files is included in the section on "Verifying LAN Device File Creation" in chapter 3.

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

Diagnostic 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	**illegal packet size**
CAUSE	The value entered for *packet_size* in the *ping(1M)* command exceeded the limit for that argument. The size limit is 4096. *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.

1019	MESSAGE	**LAN driver parity error detected on interface unit *if_unit*.**
	CAUSE	(Disaster) Driver detected a RAM hardware error in the card.
	ACTION	If the problem persists, 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.
1025	MESSAGE	**Ethernet controller could not be configured with desired SCP address.**
	CAUSE	(Disaster) The ethernet controller failed to read in the SCP address.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card and then reset the card using the landiag(1M) command. If resetting is ineffective, notify your HP representative..

1026	MESSAGE	**The ethernet controller failed to pass its self-test**
	CAUSE	(Disaster) Ethernet controller self-test failed.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card and then reset the card using the *landiag(1M)* command. If resetting does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1027	MESSAGE	**Ethernet controller failed to pass its internal loopback test.**
	CAUSE	The ethernet controller internal loopback test timed out.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card and reset the card using the landiag(1M) command. If the reset does not solve the problem, reboot. If rebooting is ineffective, notify your HP representative.
1028	MESSAGE	**LAN driver failed to loopback packet through the ethernet serial interface.**
	CAUSE	(Disaster) Ethernet controller loopback test through the media interface timed out.
	ACTION	Use the *lanscan(1M)* command to find the logical unit number of the LAN card and reset the card using the *landiag(1M)* command. If the reset does not solve the problem, reboot. If rebooting is ineffective, 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.

2004	MESSAGE	**External loopback test failed; Use landad for diagnosis**
	CAUSE	(Error) The loopback test through the external MAU failed.
	ACTION	Check the LAN cable to make sure its properly connected to the LAN card. Use the *lanscan(1M)* command to find the logical unit number of the LAN card and reset the card using the *landiag(1M)* command. If rebooting is ineffective, notify your HP representative.
2005	MESSAGE	**Ethernet controller control command timed out; resetting the driver.**
	CAUSE	An ethernet controller action command timed out. The LAN driver is currently resetting the LAN card.
	ACTION	If the problem persists, notify your HP representative.
2006	MESSAGE	**The ethernet controller received an illegal sized frame.**
	CAUSE	(Error) The ethernet controller received a frame whose Receive Buffer Descriptor (RBD) did not have its EOF bit set.
	ACTION	The frame was dropped by the LAN driver. If the problem persists, notify your HP representative.
2007	MESSAGE	**Ethernet controller bug; resetting the LAN driver.**
	CAUSE	(Error) A known ethernet controller bug has been encountered. The LAN driver is currently resetting the LAN card.
	ACTION	If the problem persists, notify your HP representative.

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.
4010	MESSAGE	**No system memory could be allocated for packet reception.**
	CAUSE	(Warning) The Receive Frame Area (RFA) for the ethernet controller is currently empty. The LAN driver is currently trying to replenish the RFA.
	ACTION	The LAN driver will automatically try to replenish the RFA every 1 second until the RFA reaches it's minimum threshold. If the problem persists, 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 802.2 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 802.2 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.

D

LAN Interface Card Statistics

This appendix contains descriptions of the status fields and statistics fields for LAN interface cards. You can display the statistics kept by the local LAN card with the *display* command in the LAN Interface Test Mode of the *landiag(1M)* diagnostic.

The display for the Series 300/400 LAN interface status is shown in Figure D-1 below, 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. Descriptions of each field follow the screen displays.

```
                    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           = 0x080009000636
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      = 0x080009001234
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   /* Not on 8x7 systems */
IEEE 802.3 TEST packets         = 1   /* Not on 8x7 systems */
Unable to respond TEST/XID pkts = 0   /* Not on 8x7 systems */
Illegal sized frames            = 0
Unable to find transmit buffers = 0   /* Not on 8x7 systems */
One of zero receive buffers     = 0   /* Not on 8x7 systems */
```

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       = 0x080009005678
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
```

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 600/800 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 600/800 only. The number of times that the card exhausted its transmit buffer space.
One or zero receive buffers	Series 600/800 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	External loopback packet corrupted.
43	Hardware failure.
44	Hardware failure.

F

LAN Filesets

This appendix describes the contents of each LAN fileset loaded during the *update* process and the correspondence between LAN filesets and the include statements/keywords used during kernel generation.

Fileset Descriptions

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.

You may select from the filesets listed below:

Table F-1. Fileset Descriptions

Fileset	Description
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 manual pages.
NETINET	Internet header files and demo programs (application development); Internet commands (ping) and kernel code; MIB kernel code.
NET	Network support header files; network support commands and kernel support; and manual 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 manual pages.
NETTRACELOG	Network tracing and logging support.
SLIP-RUN	SLIP commands and template configuration files.

Refer to table F-2 for a list of filesets required with each type of networking software.

Table F-2. Correspondence Between Networking Software and Networking Filesets

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
SLIP	BSDIPC-SOCKET NETINET NET NETTRACELOG

*DUX is an HP internal protocol used to communicate between a cluster cnode and server.

Include Statements/Keywords

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. Table F-3 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 table F-3 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 4 for additional information on filesets and procedures to generate a new kernel.

Table F-3. Correspondence Between LAN/9000 Filesets, S800 File Include Statements, and S300/S700 dfile Keywords

Fileset Name	S800 File Include Statement	S300/S700 dfile Keyword	Additional Information
BSDIPC-SOCKET	include uipc;	uipc	required for fileset
NETINET	include inet; include nm;	inet netman	required for fileset required for Network Management
NET			
NETIPC	include nipc;	nipc	required for fileset
NETTRACELOG	include netdiag1;	netdiag1	required for fileset
LAN	include lan; include lan0; include lan1; include lan3;	 lan01 lla (S300 only) num_lan_cards x	required for fileset required for CIO cards required for HP-PB cards; Series 8X2 required for HP-PB cards; Series 8X7 required for fileset required for fileset required for 3, 4 or 5 LAN cards; x = number of LAN cards; default is 2; valid range is 1 to 5
SLIP-RUN fileset	include ni;	ni	required for SLIP

G

Network Daemons and Library Routines

This appendix 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*, *nftdaemon*, *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.
nftdaemon	The daemon that supports the *dscopy* command.
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.

H

Reconfiguring the S700 Kernel (Standalone)

Caution The instructions below do not apply to kernel reconfiguration on clustered systems. Follow the instructions in *System Administration Tasks* for Series 700 systems to configure the kernel if your system is attached to a cluster.

Reconfiguring the kernel requires that you reboot your system. Note, however, the impact on other users before you shut down and reboot your system, especially the following:

- If others are logged into your system, rebooting it interrupts their work. If you have a small number of users or clients on your system, it is best to notify your users in person of the impending system shutdown. It is possible that users can be using an application and not be aware of the message sent by the *shutdown* command.
- If your system is an Internet Protocol router, rebooting it affects any IP traffic routed through your system.

To reconfigure the kernel using HP-UX commands:

1. Ensure that you have super-user capabilities.
2. Change your directory to */etc/conf*:

```
cd /etc/conf
```

Caution You must get out of the root directory because you will be creating a new kernel. Otherwise you will overwrite the current kernel.

3. Make a backup copy of your current configuration description file (which is most commonly */etc/conf/dfile*).

   Enter the following command:

   ```
   cp /etc/conf/dfile /etc/conf/dfile.old
   ```

   HP highly recommends the use of */etc/conf/dfile* as the kernel configuration file so it remains up to date with the executing kernel, */hp-ux*. Some system software depends on */etc/conf/dfile* representing the currently executing kernel.

4. Edit */etc/conf/dfile* to add drivers and/or change system parameters.

   The *dfile* is the configuration description file that generally reflects your system. (For more information about the dfile, refer to *config(1M)* in the *HP-UX Reference* manual).

5. The backup kernel is typically */SYSBCKUP*, but you can choose another name. Make a copy of the existing kernel.

Caution Do not perform this step if your system is booted from the /SYSBCKUP backup kernel. If you do, you could overwrite the only bootable kernel for your system.

   Enter the following command:

   ```
   cp /hp-ux /SYSBCKUP
   ```

   Backup kernel filenames can have no more than 11 characters for the boot ROM to recognize them. Write down the filename of the backup kernel.

6. Write down the hardware address of the system disk. There are three possible hardware address formats for accessing the system disk:

scsi.*busadr*	For a system disk attached to the Core I/O SCSI disk drive (for example, scsi.6 for a disk at bus address 6).
eisa.*slot_num.busadr*	For a system disk attached to an EISA SCSI or HP-IB card (for example, eisa.2.4 for an EISA card in slot 2 with a disk at bus address 4).

eisa.*slot_num*	For a system disk connected via the EISA LAN card (for example, eisa.3 for the EISA LAN card in slot 3).

You will need this information if the new kernel fails to boot.

7. Run *config* on the configuration description file you edited:

```
/ dfile
```

Executing *config* creates the files *conf.c* and *config.mk*. Be sure you have the correct version these files by typing `ll` (that is "el, el") from the */etc/conf* directory and verifying the last modified date and time.

Refer to *config(1M)* in the *HP-UX Reference* manual for additional information.

8. Create the new hp-ux kernel (the file *hp-ux*) in the current directory (*/etc/conf*):

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

As it is executing, *config.mk* displays the following two lines:

```
Compiling conf.c ...
Loading hp-ux ...
```

9. Bring the system into single-user mode using the *shutdown* command:

```
cd /
shutdown grace_period
```

where *grace_period* is the number of seconds the system will wait before shutting down. Specifying a grace period is optional; the default is 60 seconds. The *shutdown* command sends all users currently logged into the system a warning message that the system is shutting down. You can rely on the system default message, or you can customize the message.

10. Wait for the system to display from single-user mode.

11. Copy the new kernel to the / (root) directory:

```
cd /etc/conf
cp hp-ux /hp-ux
```

12. Halt the system:

```
reboot -h
```

13. Turn off the computer. If you are installing or removing interface cards or peripheral devices, do it now. Refer to the documents shipped with the products being installed and the *Installing Peripherals* manual for specific instructions.

Warning **Be sure to follow the ESD (Electrostatic Discharge) precautions when handling cards and devices. ESD precautions are described in the hardware installation and configuration guides.**

---

14. Turn on the power. The system will attempt to boot the new kernel.

If the new kernel fails to boot, boot the system from the backup kernel and repeat the process of creating a new kernel. See "Booting the Backup Kernel Using the Boot ROM."

Booting the Backup Kernel Using the Boot ROM

To select the backup kernel using the boot ROM:

1. Turn off the power to the computer, wait a few seconds, then turn the power back on.
2. Press the [ESC] key. In a few seconds, this message appears:

   ```
   Terminating selection process.
   ```

   A short time later, this message appears:

   ```
   Searching for potential boot devices.
   To terminate search, press and hold the ESCAPE key.

   Device Selection   Device Pat    Device Type and Utilities

   ------------------------------------------------------------
   ```

   Your computer is now searching for devices that might hold file systems from which it can boot HP-UX. As they are found, they appear in a list. A list of devices might look like this:

   ```
   P0          scsi.6.0            QUANTUM PD210S
   P1          scsi.5.0            QUANTUM PD210S
   P2          scsi.4.0            DDS_tape_drive_identifier
   P3          scsi.3.0            TOSHIBA CD-ROM DRIVE:XM
   P4          lan.123456-789abc   homebase
   ```

   This process can take several minutes. When the search ends, this list of actions appears:

   ```
        b)      Boot from specified device
        s)      Search for bootable devices
        a)      Enter boot administration mode
        x)      Exit and continue boot sequence
        ?)      Help

   Select from menu:
   ```

- If no devices are listed, there is a problem:

- Check for duplicate SCSI bus addresses or loose connections.
- Check and verify that the power switch is ON for all peripherals.

If you have performed the address, connection, and power checks, and still no devices are listed, the problem is serious. Contact your service representative for assistance.

- If no disk devices are listed, and your system is equipped with disk drives, your workstation is failing to communicate with its disks. Recheck the SCSI connectors, terminators, and hardware addreses and try again.

3. Enter the command to boot from a particular device, using the ISL interface to select a kernel other than /hp-ux. To specify boot enter boot or b followed by the index to the left of the hardware address (for example P0 or P1), followed by isl to specify the ISL interface to select a kernel to boot. For example,

```
Select from menu: b P0 isl
```

The ISL prompt, ISL>, should be displayed.

4. To list the available kernels to boot from in the root (/) directory, or any directory in the root file system, use the ls option to the *hpux* command. The output is similar to the *ls -alFH* command, except the owner, group, and date information is not displayed. Bootable kernels are those entries with an appended asterisk (*).

For example,

```
ISL> hpux ls disk(;)/.

          Secondary Loader 9000/700
          Revision 1.1

          drwxr-xr-x    9      2048 ./
          drwxr-xr-x    6      2048 ../
          drwxr-xr-x    2      4096 lost+found/
          -rw-rw-r--    1       746 .profile
          drwxrwxr-x    2      1024 bin/
          drwxr-xr-x   12      1024 dev/
          drwxrwxr-x    5      1024 etc/
          drwxrwxrwx    2        64 tmp/
          drwxrwxr-x    3      1024 usr/
          Hrwxrwxr-x    3      1024 foo+
          -rwxr-xr-x    1    884736 hp-ux*
```

```
-rwxr-xr-x    1    884736 SYSBCKUP*
-rwxr-xr-x    1   1032192 hp-ux.test*
```

In this example there are three available kernels to boot: /hpux, /hpux.test, and /SYSBCKUP.

For more information on the *hpux* command, refer to *hpux_700(1M)* in the *HP-UX Reference* manual.

5. Enter the *hpux* command to boot your backup operating system.

   To boot from the internal disk drive at SCSI address 6, type:

   ```
   ISL > hpux boot disk(scsi.6;0)/SYSBCKUP
   ```

   The backup kernel will begin to boot. When it displays the login prompt, login again and try to reconfigure the kernel again.

Caution When you reconfigure the kernel for the second time using the steps described in "Reconfiguring the S700 Kernel." Do not create a backup of the current kernel. Since you are currently booted from the backup kernel, copying */hp-ux* to a backup kernel might overwrite the only bootable kernel on your system!

If your computer still fails to boot, there is something wrong with either the file system or the hardware. If you suspect a file system failure, or if you think that something is wrong with the hardware, refer to your owner's guide.

I

Reconfiguring the S300/400 Kernel (Standalone)

Caution The instructions below do not apply to kernel reconfiguration on clustered systems. Follow the instructions in *System Administration Tasks* for Series 300/400 systems the kernel if your system is attached to a cluster.

Reconfiguring the kernel requires that you reboot your system. Note, however, the impact on other users before you shut down and reboot your system, especially the following:

- If others are logged into your system, rebooting it interrupts their work. If you have a small number of users or clients on your system, it is best to notify your users in person of the impending system shutdown. It is possible that users can be using an application and not be aware of the message sent by the *shutdown* command.
- If your system is an Internet Protocol router, rebooting it affects any IP traffic routed through your system.

To reconfigure the kernel using HP-UX commands:

1. Ensure that you have superuser capabilities.
2. Change your directory to `/etc/conf`:

   ```
   cd /etc/conf
   ```

Caution You must get out of the root directory because you will be creating a new kernel. Otherwise you will overwrite the current kernel.

3. Make a backup copy of your current configuration description file (which is most commonly */etc/conf/dfile*).

   Enter the following command:

   ```
   cp /etc/conf/dfile /etc/conf/dfile.old
   ```

   HP highly recommends the use of */etc/conf/dfile* as the kernel configuration file, so it remains up to date with the executing kernel, */hp-ux*. Some system software depends on */etc/conf/dfile* representing the currently executing kernel.;dfile

4. Edit `/etc/conf/dfile` to add drivers and/or change system parameters.

   The *dfile* is the configuration description file that generally reflects your system. (For more information about the *dfile*, refer to *config(1M)* in the *HP-UX Reference* manual).

5. Make a copy of the existing kernel.

Caution Do not perform this step if your system is booted from the /SYSBCKUP backup kernel. If you do, you could overwrite the only bootable kernel for your system.

   Enter the following command:

   ```
   cp /hp-ux /SYSBCKUP
   ```

   Write down the filename of the backup kernel.

Note You must name your backup kernel /SYSBCKUP if you want to be able to boot the backup kernel from the Boot ROM.

---

6. Run *config* on the configuration description file you edited:

   ```
   /etc/config dfile
   ```

   Executing *config* creates the files *conf.c* and *config.mk*. Be sure you have the

correct version these files by typing `ll` (that is "el, el") from the */etc/conf* directory and verifying the last modified date and time.

Refer to *config(1M)* in the *HP-UX Reference* manual for additional information.

7. Create the new hp-ux kernel (the file hp-ux) in the current directory (*/etc/conf*):

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

   As it is executing, *config.mk* displays the following two lines:

   ```
   Compiling conf.c ...
   Loading hp-ux ...
   ```

8. Bring the system into single-user mode using the *shutdown* command:

   ```
   cd /
   shutdown grace_period
   ```

   where *grace_period* is the number of seconds the system will wait before shutting down. Specifying a grace period is optional; the default is 60 seconds. The *shutdown* command sends all users currently logged into the system a warning message that the system is shutting down. You can rely on the system default message, or you can customize the message.

9. Wait for the system to display from single-user mode.

10. Copy the new kernel to the / (root) directory:

    ```
    cd /etc/conf
    cp hp-ux /hp-ux
    ```

11. Halt the system:

    ```
    $ reboot -h
    ```

12. Turn off the computer. If you are installing or removing interface cards or peripheral devices, do it now. Refer to the documents shipped with the products being installed and the *Installing Peripherals* manual for specific instructions.

Warning **Be sure to follow the ESD (Electrostatic Discharge) precautions when handling cards and devices. ESD precautions are described in the hardware installation and configuration guides.**

13. Turn on the power. The system will attempt to boot the new kernel.

If the new kernel fails to boot, boot the system from the backup kernel and repeat the process of creating a new kernel. See "Booting the Backup Kernel Using the Boot ROM."

To boot a Series 300/400 backup kernel, select /SYSBCKUP from the boot ROM. The Series 300/400 boot ROM does not support booting from filenames other than /SYSHPUX and /SYSBCKUP.

Booting the Backup Kernel Using the Boot ROM

If your system is a Series 300/400 and the new kernel fails to boot:

1. Turn the computer off and then on (cycling power).
2. Hold down the space bar during bootup to enter the boot ROM **attended mode**. This halts the automatic boot mechanism and allows you to manually select the operating system to load.
3. Type in the two-character code associated with the backup kernel SYSBCKUP.

Your backup kernel will begin to boot. When you are given the login prompt, login again and try to reconfigure the kernel again.

Caution If you reconfigure the kernel for the second time using the steps described in "Reconfiguring the S300/400 Kernel," DO NOT create a backup of the current kernel. Since you are currently booted from the backup kernel, copying */hp-ux* to a backup kernel could overwrite the only bootable kernel on your system!

J

Reconfiguring the S800 Kernel (Standalone)

Reconfiguring the kernel requires that you reboot your system. Note, however, the impact on other users before you shut down and reboot your system, especially the following:

- If others are logged into your system, rebooting it interrupts their work. If you have a small number of users or clients on your system, it is best to notify your users in person of the impending system shutdown. It is possible that users can be using an application and not be aware of the message sent by the *shutdown* command.
- If your system is an Internet Protocol router, rebooting it affects any IP traffic routed through your system.

To reconfigure the kernel using HP-UX commands:

1. Ensure that you have superuser capabilities.
2. Change your directory to */etc/conf/gen*:

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

Caution You must get out of the root directory because you will be creating a new kernel. Otherwise you will overwrite the current kernel.

---

3. Make a backup copy of your current configuration description file (which is most commonly *S800*).

   Enter the following command:

   ```
   $ cp /S800 /S800.Bckup
   ```

4. Edit the *S800* file to add drivers and/or change system parameters and save it.

   The *S800* is the configuration description file that generally reflects your system. (For more information about the *S800* file, refer to *config(1M)* in the *HP-UX Reference* manual).

5. After making changes, regenerate the kernel with *uxgen*, using the edited *S800* file as input:

   ```
   /etc/uxgen S800
   ```

   The *uxgen* program generates a new kernel, calling it *hp-ux*, and puts it in the directory */etc/conf/S800*.

6. Make a copy of the existing kernel.

Caution Do not perform this step if your system is booted from the /SYSBCKUP backup kernel. If you do, you could overwrite the only bootable kernel for your system.

   To do so, enter the following command:

   ```
   $ cp /hp-ux /SYSBCKUP
   ```

   Write down the filename of the backup kernel.

7. Write down the information you need in case the new kernel doesn't boot. You'll need the name of the device driver for the system disk ("disc1" for HP-IB, "disc2" for HP-FL, "disc4" for HP-FL on HP-PB computers, "disc3" for SCSI), hardware address of the system disk (Primary Boot Path), section number of the root file system, and the name of the backup kernel (typically "/SYSBCKUP").

   To determine the device driver and hardware address of the system disk and to find the section number for the root file system on the system disk, you can use the `/etc/devnm` and `/etc/lssf` commands as follows:

   Enter the following command to get the device file name for the root file system by entering:

   ```
   user/etc/devnm /
   ```

and you will get something like:

```
/dev/dsk/c0d0s13 /
```

Then, by entering:

```
user /etc/lssf /dev/dsk/c0d0s13
```

you will get output to the screen like the following that will provide the information you need:

```
disc1 lu 0 unit 0 section 13 address 4.0.0 /dev/dsk/c0d0s13
```

The driver ("disc1"), the hardware address (4.0.0), and the disk section number ("s13") are in the output.

8. Copy the new kernel to the / (root) directory:

```
$ cd /etc/conf/S800
$ cp hp-ux /hp-ux
```

9. Reboot the system to use the new kernel:

```
$ /etc/reboot
```

If the new kernel fails to boot, boot the system from the backup kernel and repeat the process of creating a new kernel. To do so, refer to the instructions in the following section.

Booting the Backup Kernel

If your system is a Series 600/800 and the new kernel fails to boot:

1. Turn the system's power off, and then on again.
2. Interrupt the autoboot process by pressing any key.
3. Respond "Y" to the prompt: "Boot from primary path?"
4. Respond "Y" to the prompt: "Interact with IPL?"
5. Using the information you acquired before shutting the system down (see previous section), formulate the ISL command using the syntax described as follows:

   ```
   hpux disc1(x.y.z;s)/SYSBCKUP
   ```

   where *x.y.z* is the hardware path of the system disk and *s* is the section number of the root file system. If the system disk is HP-FL, then use *disc2* instead of *disc1*. If you named the backup kernel something other than /SYSBCKUP, use that name instead. Once you have the backup kernel running, you can re-edit the *S800* file and try the process again.

Note Instead of *uxgen*, you may want to use the *regen(1M)* utility which is friendlier and does more for you than *uxgen*; *regen*, however, allows you less flexibility. Refer to the *HP-UX Reference* manual for more information about *regen*.

K

Logging and Tracing Subsystems

This appendix lists the LAN/9000 logging and tracing subsystems that can be referenced with the `-entity` parameter of the *nettl(1M)* command.

Logging Subsystems

NS_LS_NFT	Logs internal events and errors in NFT.
NS_LS_NI	Logs outbound and inbound SLIP packets.
NS_LS_TCP	Logs TCP state changes.
NS_LS_IP	Logs ICMP input and sending ICMP error packets.
NS_LS_PROBE	Logs ARPs with duplicate IP address (Sender's IP address matches Receiver's IP address).
NS_LS_NFS	Logs RPC and NFS packets.

Tracing Subsystems

NS_LS_LOOPBACK	Traces loopback packets.
NS_LS_DRIVER	Traces 802.3/Ethernet packets.
TOKEN	Traces Token Ring packets.
FDDI	Traces FDDI packets.

Sample netlinkrc File

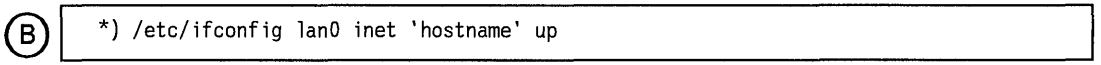
This appendix provides an example */etc/netlinkrc* file. The alphabetic references in the left margins map to the corresponding letters in figure 4-1 in chapter 4.

```
#! /bin/sh

# @(#)netlinkrc: $Revision: 1.6.109.6 $ $Date: 92/03/24 14:04:59 $
# $Locker:  $

#
# Shell script for initialization of link networking product.
#

# net_init flag is used for Instant Ignition. If net_init is set,
# then netlinkrc return "exit 1". In order for Instant Ignition
# to work correctly, netlinkrc needs to check the STATUS variable
# after each program or scripts it calls.
#
net_init=0

if [ -f /etc/clusterconf ]
then
        ROOTSERVER='/bin/cnodes -r'
        NODENAME='/bin/cnodes -m'
        DOMAIN='/bin/cnodes -r'
        ORGANIZATION=diskless
else
        ROOTSERVER='hostname'
        NODENAME=$ROOTSERVER
        DOMAIN='/bin/uname -n'
        ORGANIZATION=standalone
fi

#
# Start logging daemon *before* any other networking initialization.
# See nettl(1m) for more information.
#
```

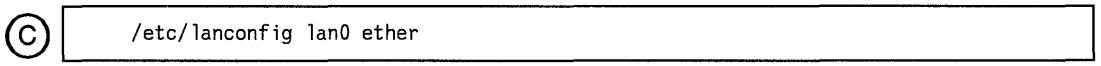

```
/etc/nettl -start
```

```
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
   net_init=1
fi
#
# Remove the existing /etc/netstat_data file.  The first time
# netstat is executed, a new /etc/netstat_data file will be
# created.
#
/bin/rm -f /etc/netstat_data

#
# Initialize networking interfaces.
#
# (STEP 1)
#
# The ifconfig(1m) command assigns an IP address to a LAN interface and
# configures network interface parameters.  The lanconfig(1m) command defines
# the packet encapsulation method for the LAN interface.
#
# The "case $NODENAME" construct below allows each node in a diskless cluster
# to execute node specific calls if necessary.  Add entries to
# the case construct for specific nodes in the diskless cluster only if
# needed.  For example, if a specific node has more than one LAN interface,
# the node must execute separate commands for each of the interfaces.
#
# For example:
#
#   case $NODENAME in
#     $ROOTSERVER) /etc/ifconfig lan0 inet 192.6.1.3 up
#                  /etc/lanconfig lan0 ether
#                  /etc/ifconfig lan1 inet 15.4.64.1 netmask 255.255.248.0 up
#                  /etc/lanconfig lan1 ether
#                  ;;
#                *) /etc/ifconfig lan0 inet 'hostname' up
#                   /etc/lanconfig lan0 ether ieee
#                   ;;
#   esac
#   /etc/ifconfig lo0 inet 127.0.0.1 up
#
# assigns to the two interfaces lan0 and lan1 on a rootserver the DARPA
# Internet addresses 192.6.1.3 and 15.4.64.1 respectively; the lan0
# interfaces on all other nodes (* is the wildcard) are assigned their
# respective internet addresses as found in /etc/hosts.
#
# The ifconfig command line below is sufficient to initialize the network
# interface for any node that has one LAN interface card and whose
# hostname and Internet address are present in the hosts(4) file.
#
```

```
# NOTE:  If the ifconfig command line does not specify a subnet mask,
#        the subnet mask defaults to the network mask.
#        It is not necessary for both encapsulation methods to be turned on
#        for the LAN Interface.  For further explanation see lanconfig(1m)
#
# The loopback interface must be explicitly configured for each address
# family of interest.  The following command assumes that the hostname
# has already been set and is mapped to an IP Address in /etc/hosts.
#
# SEE ALSO: ifconfig(1m), lanconfig(1m)

case $NODENAME in
```




```
     STATUS=$?
     if [ ! $STATUS -eq 0 ]
     then
        net_init=1
     fi
```




```
     STATUS=$?
     if [ ! $STATUS -eq 0 ]
     then
        net_init=1
     fi
     ;;
esac
/etc/ifconfig lo0 inet 127.0.0.1 up
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
   net_init=1
fi
# The x25init(1m) command configures X.25 network interface parameters.  The
# "case $NODENAME" construct below allows each node in a diskless cluster
# to execute node specific x25init calls if necessary.  Add entries to
# the case construct for specific nodes in the diskless cluster only if
# the nodes have X.25 interfaces. The nodes must execute separate x25init
# commands for each of the interfaces. The STATUS checking is for Instant
# Ignition.
#
# For example:
#
#   case $NODENAME in
#          NODEA ) /etc/x25init -c /etc/x25/config_filename1
#                  STATUS=$?
#                  if [ ! $STATUS -eq 0 ]
#                  then
#                     net_init=1
#                  fi
```

```
#                   /etc/x25init -c /etc/x25/config_filename2
#                   STATUS=$?
#                   if [ ! $STATUS -eq 0 ]
#                   then
#                      net_init=1
#                   fi
#                   /etc/x25init -a /etc/x25/ip_x25_mapfile
#                   STATUS=$?
#                   if [ ! $STATUS -eq 0 ]
#                   then
#                      net_init=1
#                   fi
#                   ;;
#           NODEB ) /etc/x25init -c /etc/x25/config_file_nodea
#                   STATUS=$?
#                   if [ ! $STATUS -eq 0 ]
#                   then
#                      net_init=1
#                   fi
#                   ;;
#     esac
#
# initializes two X.25 interfaces on NODEA and one interface on NODEB.
# For nodes which have IP configured over X.25, the x25init -a command
# provides the mapping of IP Addresses to X.121 addresses.  It is recommended
# to put the configuration and ipmap files in the /etc/x25 directory.
#
# In the above example, at least one of NODEA's X.25 Cards supports IP
# since IP-to-X.25 Map table is initialized on NODEA.
#
# SEE ALSO: x25init(1m)
```

(D) If installed, add X.25 commands here

```
#
# Initialize network routing.
#
# (STEP 2) (OPTIONAL, FOR NETWORKS WITH GATEWAYS ONLY)
#
# The route(1m) command manipulates the network routing tables.
# The "case $NODENAME" construct below allows each node in a diskless
# cluster to execute node specific route calls if necessary.  Add entries
# to the case construct for specific nodes in the diskless cluster if needed.
# The STATUS checking is for Instant Ignition.
#
# For example,
#
#   case $NODENAME in
#     $ROOTSERVER) /etc/route add 192.0.2 gatenode 1
#                  STATUS=$?
#                  if [ ! $STATUS -eq 0 ]
#                  then
#                     net_init=1
```

```
#                 fi
#                 ;;
#              *) /etc/route add default 15.2.104.69 1
#                 STATUS=$?
#                 if [ ! $STATUS -eq 0 ]
#                 then
#                    net_init=1
#                 fi
#                 ;;
#   esac
#
# adds network destination "192.0.2" to the rootserver's routing tables,
# indicating a correspondence between that destination and the gateway
# "gatenode", and specifying the number of hops to the gateway as 1.  For
# all other nodes (* is the wildcard), the default gateway is set to
# 15.2.104.69.
#
# The route command should be invoked once per gateway.
#
# SEE ALSO: route(1m), routing(7)

case $NODENAME in
```

(E)

```
          *) # /etc/route add default 'hostname'
             ;;
esac

#
# Initialize the network node name.
#
# (STEP 3)
#
# The nodename(1m) command assigns an NS node name to the node.
# Nodename takes an option of the form "nodename.domainname.orgname" where,
#
#   nodename            is the name of the local node
#   domainname          is the name of the domain
#   orgname             is the name of the organization
#
# Each name must start with an alphabetic character.
#
# It is strongly recommended that the string used for "nodename" above be
# identical to the string used as an argument to the hostname(1) command,
# which is typically invoked from the system initialization shell script
# file "/etc/rc".  The NS nodename used on each node in your network needs
# to be unique within that network. The "case $NODENAME" construct below
# allows each node in a diskless cluster to execute a node specific
# nodename(1) call if necessary.  Add entries to the case construct for
# specific nodes in the diskless cluster only if needed.
#
```

```
# For example,
#
#   case $NODENAME in
#     * )  /bin/nodename '/bin/uname -n'.mydomain.myorg
#          ;;
#   esac
#
# sets the NS nodename for all nodes (* is the wildcard) in domain
# "mydomain" and organization "myorg".
#
# The nodename command line below sets the nodename field to the system
# hostname, the domainname field to the rootserver's name, and the orgname
# field to "diskless".
#
# SEE ALSO: nodename(1)

if [ -x /bin/nodename ]
then
   case $NODENAME in
           *) /bin/nodename '/bin/uname -n'.$DOMAIN.$ORGANIZATION
              STATUS=$?
              if [ ! $STATUS -eq 0 ]
              then
                 net_init=1
              fi
              ;;
   esac
fi

#
# Start remote loop back daemon
#
if [ -f /usr/adm/rld.log ]
then
        /bin/mv /usr/adm/rld.log /usr/adm/OLDrld.log
fi
[ -x /etc/rlbdaemon ]  &&  (/etc/rlbdaemon 2&1 )  /usr/adm/rld.log
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
   net_init=1
fi

/bin/echo "Network Link started"
```

```
#
# Start NFS.  This requires installation of the NFS product.
#
if [ -x /etc/netnfsrc ]
then
```

(H)

```
   /etc/netnfsrc
```

```
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
       net_init=1
    fi
fi

/bin/echo "ARPA/Berkeley daemons started: \c"

#
# Start the Internet daemon.
#
```

(I)

```
[ -x /etc/inetd ]  &&  /etc/inetd  &&  /bin/echo "inetd  \c"
```

```
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
   net_init=1
fi

#
# Start ARPA/BSD networking services.
#
```

(J)

```
if [ -x /etc/netbsdsrc ]
then
        /etc/netbsdsrc
```

```
        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

/bin/echo
```

```
#
# Do nfs mounts after inetd is running
#
```

(K)

```
if [ -x /etc/netnfsrc2  -a  -f /etc/nfs.up ]
then
    /etc/netnfsrc2

    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
       net_init=1
    fi
fi

#
# Start NS networking services.
#
```

(L)

```
if [ -x /etc/netnssrc ]
then
        /etc/netnssrc

        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

#
# Start HP Network Management Agent
#
```

(M)

```
if [ -x /etc/netnmrc ]
then
        /etc/netnmrc

        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

#
# Start HP LAN Manager/X.
#
```

(N)

```
if [ -x /etc/netlmrc ]
then
        /etc/netlmrc
```

L-8 Sample netlinkrc File

```
        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

#
# Start NCS.  This requires installation of the NCS product.
# NCS must be started before any other NCS products are started.
#
```

(O)

```
if [ -x /etc/netncsrc ]
then
        /etc/netncsrc
```

```
        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

#
# Start NetLS.  This requires installation of the NetLS product.
# NCS must be started before NetLS is started.
#
```

(P)

```
if [ -x /etc/netlsrc ]
then
        /etc/netlsrc
```

```
        STATUS=$?
        if [ ! $STATUS -eq 0 ]
        then
           net_init=1
        fi
fi

# return exit code for Instant Ignition
if [ $net_init -eq 0 ]
then
  exit 0
else
  exit 1
fi
```

M

Related Manuals, Protocols, and Standards

This section provides lists of related networking and system documentation along with lists of the protocols and standards on which the LAN/ARPA products are based.

Reference Manual Guide

Table M-1. List of Networking Manuals

For Information on:	Read:
General Networking	*Networking Overview*
Installing LAN Hardware	*LAN Interface Controller (LANIC) Installation and Reference Manual* *LAN Cable and Accessories Manual*
Troubleshooting LAN	*Installing and Administering LAN/9000*
Link Level Access	*LLA Programmer's Guide*
BSD Interprocess Communication	*Berkeley IPC Programmer's Guide*
Network Interprocess Communication (NetIPC)	*NetIPC Programmer's Guide*
General ARPA/Berkeley Services Information	*Using ARPA Services*
Troubleshooting ARPA Services	*Administering ARPA Services*
General NFS Services Information	*Installing Administering NFS Services* *Programming and Protocols for NFS Services*
General NS/9000 Information	*Using Network Services*
Troubleshooting NS	*Installing and Administering NS*
EISA Configuration Utility	*Installing Peripherals*

Table M-1. List of Networking Manuals (Cont'd)

For Information on:	Read:
Installing Token Ring/9000 Software	*Installing and Administering Token Ring/9000 Software*
Installing FDDI/9000 Software	*Installing and Administering FDDI/9000 Software*
Using Serial Line IP Protocols with HP 9000	*Using Serial Line IP Protocols*

Table M-2. List of Related HP-UX System and Programmer's Manual

For Information on:	Read:
System Administration	*Installing and Updating HP-UX* *System Administration Tasks*
HP-UX Reference Manuals	*HP-UX Reference*
HP-UX Operating System	*HP-UX User's Guide* *HP-UX Concepts and Tutorials*
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*
Cluster Management	*Managing Clusters of HP 9000 Computers: Sharing the HP-UX File System* (Series 700) *Managing Clusters of HP 9000 Computers: Sharing the HP-UX File System* (Series 300/400)

Table M-3. List of Networking Protocols and Standards

For Information on:	Read:
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*
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
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

Index

!

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

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

Edition 3 . . . . . July 1992

HEWLETT PACKARD

Copyright © 1992
Hewlett-Packard Company
Printed in USA 07/92

Customer Order No.
98194-60530

Manufacturing No.
98194-90035
Mfg. number is for HP internal use only

98194-90035