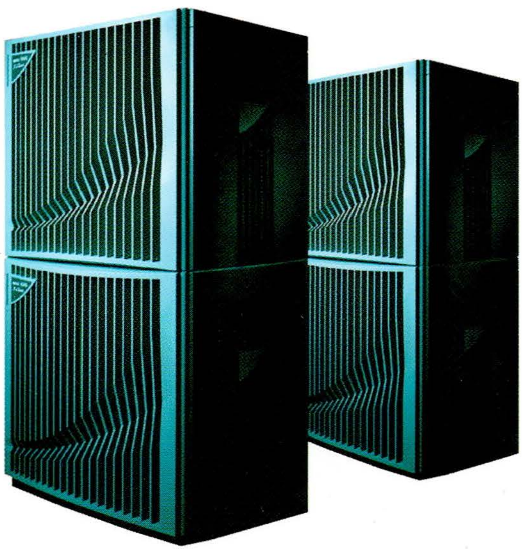
HEWLETT® PACKARD

X-Class Servers



Exemplar Networking Guide

Sixth Edition

Exemplar Networking Guide

X-Class Servers

B5655-90029

Sixth Edition

June 1997

Hewlett-Packard Company
Convex Division
Richardson, Texas
United States of America

Exemplar Networking Guide
X-Class Servers

B5655-90029

Notice

The information contained in this document is subject to change without notice.

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

This entire book is recyclable.

Printed in the United States of America

Revision Information for

Exemplar Networking Guide
X-Class Servers

Edition	Document No.	Description
Sixth	B5655-90029	Released in June 1997 with SPP-UX V5.2. Includes minor updates.
Fifth	B5655-90004	Released in January 1997 with SPP-UX V5.1. Includes updated terminology, file, and path information for the new operating system.
Fourth	710-031130-003	Released in June 1996 with SPP-UX V4.0. Included new chapter on ATM configuration.
Third	710-031130-002	Released in September 1995 with SPP-UX V3.1. Included new chapter on HIPPI configuration.
Second	710-031130-001	Released in June 1995 with SPP-UX V3.1. Included new chapters on automounter, Ethernet configuration, and troubleshooting.
First	710-031130-000	Initial release December 1994 with SPP-UX V2.0.

Contents

Figures

Tables

Preface

This guide focuses on the tasks that you as the system manager perform to configure and troubleshoot interface controllers on Hewlett-Packard Exemplar X-Class Technical Servers.

Using this guide

Use Table 1 to select which portions of the book to read.

Table 1 Book organization

Read	To learn more about
Introduction	SPP-UX interface controllers and various HP-UX utilities, products, and mount options that SPP-UX does not support when setting up a network.
Configuring FDDI	Configuring FDDI into your HP Exemplar Technical Servers.
Configuring Ethernet	Configuring Ethernet into your HP Exemplar Technical Servers.
Configuring HIPPI	Configuring HIPPI into your HP Exemplar Technical Servers.
Configuring ATM	Configuring ATM into your HP Exemplar Technical Servers.
Troubleshooting network interfaces	Flowcharts to help you diagnose and solve network interface problems.
Tuning system parameters	Parameters that you can tune to improve system throughput and performance and streamline network protocols.

Notational conventions

Notational conventions used in this book are described below.

`bold monospace`

In command examples, text shown in bold monospace identifies input that must be typed exactly as shown.

`monospace`

In paragraph text, monospace identifies command names, system calls, and data structures and types. In command examples, monospace identifies command output, including error messages.

Italic

In paragraph text, *italic* identifies titles of documents or highlights the first occurrence of important terms.

In command syntax diagrams, *italic* identifies variables that you must provide. The following command example uses brackets to indicate that the variable *output_file* is optional:

command *input_file [output_file]*

Note

A note highlights important supplemental information.

Associated documents

Associated documents include:

- *HP Network Overview* (B1029-90000)
- *HP Installing & Administering Internet Services* (B1030-90000)
- *HP Installing & Administering NFS Services* (B1031-90000)
- *PCI Fast Ethernet Installation and Service Guide* (A4716-90010)
- *PCI Dual Attach FDDI Installation and Service Guide* (A4716-90012)
- *PCI, ATM and HIPPI Installation and Service Guide* (A4716-90017)
- *PCI Ultra SCSI Installation and Service Guide* (A4716-90011)

Ordering documents

To order additional copies of this document or other documents listed in "Associated documents," send requests to:

Hewlett-Packard Company
Convex Division
Customer Service
P.O. Box 833851
Richardson TX 75083-3851 USA

Please include the order number (xxxxx-9xxxx number) or the exact title of the document.

Technical assistance

If you have questions that are not answered in this book, contact the Hewlett-Packard Convex Technical Assistance Center (TAC) at the following locations:

- Within the continental U.S., call 1 (800) 952-0379.
- From Canada, call 1 (800) 345-2384.
- All other locations, contact your local Hewlett-Packard office.

You can also use the `contact` utility to report any problems you are having.

Introduction 1

This chapter provides a brief introduction to the SPP-UX interface controllers. It also describes the differences between setting up SPP-UX and HP-UX networks. The topics covered include:

- Introducing SPP-UX interface controllers
- Understanding network differences

Introducing SPP-UX interface controllers

SPP-UX supports the following interface controllers:

- Fiber Distributed Data Interface (FDDI)
- Ethernet
- High Performance Parallel Interface (HIPPI)
- Asynchronous Transfer Mode (ATM)

Fiber Distributed Data Interface

Fiber Distributed Data Interface (FDDI) is a PCIbus controller that connects Hewlett-Packard Exemplar Technical Servers directly to FDDI networks. FDDI offers the next level of LAN performance beyond Ethernet, with a peak data rate of 100 megabits per second. FDDI also covers a greater service area than Ethernet, allowing up to 2 kilometers distance between nodes when using multimode cable; and up to 40 kilometers when using single mode cable.

The FDDI standard specifies a fiber transmission medium and a token ring topology. FDDI supports a dual attached Class A FDDI station and connects to both the primary and secondary rings of the network. The secondary ring provides redundancy in case of a failure of the primary ring. Dual ring support provides higher reliability and availability of the FDDI network.

Refer to "Configuring FDDI" on page 5 for more information.

Ethernet

Hewlett-Packard Exemplar Technical Servers interface to an Ethernet network through PCIbus controllers. These controllers support the attachment of transceivers. Ethernet networks have a peak data rate of 100BASE-T (100 megabits per second) over segments of up to 500 meters in length.

Refer to "Configuring Ethernet" on page 17 for more information.

High Performance Parallel Interface

High Performance Parallel Interface (HIPPI) is a copper-based data communications standard capable of transferring data at 800 Mbit/sec over 32 parallel lines or 1.6 Gbit/sec over 64 parallel lines.

Refer to "Configuring HIPPI" on page 31 for more information.

Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) is the technique for transport, multiplexing, and switching that provides the high degree of flexibility required by B-ISDN. ATM is a connection-oriented protocol that employs fixed-size packets with a 5-byte header and 48 bytes of information.

Refer to "Configuring ATM" on page 39 for more information.

Understanding network differences

The documents listed below describe the HP-UX internet utilities, network file system, and network information system. You should refer to these documents when setting up a SPP-UX network.

- *HP Network Overview* (B1029-90000)
- *HP Installing & Administering Internet Services* (B1030-90000)
- *HP Installing & Administering NFS Services* (B1031-90000)

There are certain differences, however, between setting up SPP-UX and HP-UX networks. The following paragraphs describe various HP-UX utilities, products, and mount options that SPP-UX does not support.

SPP-UX does not support the following HP-UX utilities:

- `sam`
- `fuser`

SPP-UX does not support the following HP-UX products:

- X.25
- PPP/SLIP
- VT3K
- OSI (OTS, XTI, FTAM, X.400, and MMS)
- SNA
- DCE
- Secure Internet Services

SPP-UX Internet Services supports TCP/IP with the following exceptions:

- Dynamic Host Configuration Protocol
- Distance-Vector Multicast Routing Protocol
- Internet Group Management Protocol
- Bootstrap Protocol
- Dynamic Maximum Transmission Unit Discovery

SPP-UX Internet Services supports the HP-UX Application Binary Interface with the following exceptions:

- Link Level Access
- Data Link Provider Interface
- STREAMS system calls (`getmsg`, `putmsg`, and `poll`)
- XTI/TLI

SPP-UX Network File System does not support the following mount options:

- `retry`
- `noac`
- `nocto`
- `acdirmax`
- `acdirmin`
- `acregmax`
- `acregmin`
- `actimeo`

Configuring FDDI

2

This chapter describes how to configure a Fiber Distributed Data Interface (FDDI) network device into your SPP-UX network. The topics covered include:

- Defining an FDDI
- Configuring FDDI into your network
- Verifying your network's configuration

Note

This chapter does not explain how to install FDDI. You must install and test the FDDI hardware, and connect the FDDI card to an active ring before you begin configuration. Refer to *PCI Dual Attach FDDI Installation and Service Guide* (A4716-90012) for more information.

FDDI is a PCIbus controller that connects Hewlett-Packard Exemplar Technical Servers directly to FDDI networks. FDDI offers the next level of LAN performance beyond Ethernet, with a peak data rate of 100 megabits per second. FDDI also covers a greater service area than Ethernet, allowing up to 2 kilometers distance between nodes.

The FDDI standard specifies a fiber transmission medium and a token ring topology. FDDI supports a dual attached Class A FDDI station and connects to both the primary and secondary rings of the network. The secondary ring provides redundancy in case of a failure of the primary ring. Dual ring support provides higher reliability and availability of the FDDI network.

Note

You perform initial SPP-UX configuration steps on the test station at SPP-UX boot time. Refer to the SPP-UX Release Notice specific to your version of the operating system for more information. SPP-UX does not support the lanscan(1M) command—use netstat(1) instead.

Defining an FDDI

Before booting the nodes of your system, you must specify a logical unit number for each hypernode's FDDI controller to OpenBoot. You must also determine the hardware configuration of your system by looking at the device tree.

Defining an FDDI with OpenBoot commands

Steps 1 through 7 detail the logical unit number specification process and must be performed for each FDDI adapter on each Exemplar hypernode.

Step 1 Determine the full device name in each HP Exemplar Technical Servers console window. At the OpenBoot `ok` prompt, enter:

`show-devs`

`show-devs` with no argument displays all devices known on the hypernode. The two HIPPI controllers are shown below as HIPPI-A and HIPPI-B; this indicates the separate source (write) and destination (read) controllers, respectively. Each controller must appear in the show-devs output, as exemplified below:

```
[0:1] ok show-devs
/pci@fe,290000
/pci@fe,210000
/pci@fe,90000
/pci@fe,10000
/HP,PA80000@fc,30000
/HP,PA80000@fc,a0000
/HP,PA80000@fc,b0000
/HP,PA80000@fc,120000
/HP,PA80000@fc,130000
/HP,PA80000@fc,1a0000
/HP,PA80000@fc,1b0000
/HP,PA80000@fc,220000
/HP,PA80000@fc,230000
/HP,PA80000@fc,2a0000
/HP,PA80000@fc,2b0000
/HP,PA80000@fc,320000
/HP,PA80000@fc,330000
/HP,PA80000@fc,3a0000
/HP,PA80000@fc,3b0000
/ramdisk
/virtual-memory@0,fffa0000
/memory@0,0
/sim@0,0
/core@f0,f0000000
/pty
```

```
/openprom
/chosen
/aliases
/options
/packages
/pci@fe,290000/qlisp@1000,0
/pci@fe,290000/pci1011,9@800,0          (Ethernet)
/pci@fe,290000/qlisp@1000,0/st
/pci@fe,290000/qlisp@1000,0/sd
/pci@fe,210000/qlisp@1000,0
/pci@fe,210000/pci107e,8@0,0                (ATM)
/pci@fe,210000/qlisp@1000,0/st
/pci@fe,210000/qlisp@1000,0/sd
/pci@fe,90000/pci1077,1020@1000,0
/pci@fe,90000/pci1011,f@0,0                (FDDI)
/pci@fe,10000/pci5555,3@800,0           (HIPPI-A)
/pci@fe,10000/pci5555,3@0,0             (HIPPI-B)
/core@f0,f0000000/sram@0,800000
/core@f0,f0000000/tty@0,d46000
/core@f0,f0000000/lan@0,d30000
/core@f0,f0000000/lcd@0,d4c000
/core@f0,f0000000/nvram@0,d00000
/core@f0,f0000000/rtc@0,d07ff8
/core@f0,f0000000/epuc@0,c00000
/core@f0,f0000000/emuc@0,c10000
/core@f0,f0000000/flash
/packages/firmware
/packages/dns
/packages/inet
/packages/obp-tftp
/packages/deblocker
/packages/tape-label
/packages/disk-label
/packages/lif
/packages/nfs
/packages/hfs
/packages/firmware/isp-fw-v2-10
[0:1] ok
```

Step 2 Find the device path that includes the string **pci1011,f** in the output from the `show-devs` command above. It appears as the output example below:

`/pci@fe,90000/pci1011,f@0,0`

Step 3 Enter the entire line of text as *device_string* in the following command:

`ok` **`mkmap`** *logical_unit_number device_string*

where

Configuring FDDI

`mkmap`

Defines an SPP-UX logical-unit to physical-unit mapping. This mapping is a label for tape and network devices that cannot be labeled like disks.

logical_unit_number

Designates the logical-unit number of each FDDI device.

You may want to use the *interface* unit number you supply when you enable your network with the `ifconfig` command. For instance, if your interface is `fddi0`, the *logical_unit_number* should be `0`.

device_string

Represents the full FDDI device name;
`/pci@fe,90000/pci1011,f@0,0`

Your FDDI device's logical unit number is assigned by OpenBoot at hypernode power on/reset time.

Step 4 Enter **`show-map`** to list current mappings and check your `mkmap` command.

Step 5 Make sure your current directory is root. If it is not, change to the root directory by entering **`cd /`**

Step 6 Add your new logical unit. Enter:

```
ok reset
```

Step 7 Boot the nodes of your system by entering the proper command at the OpenBoot `ok` prompt in each HP Exemplar Technical Servers console window. Refer to the SPP-UX Release Notice specific to your version of operating system for more information.

When SPP-UX boots, the FDDI driver asks OpenBoot for its logical unit number. The driver keeps track of this number.

OpenBoot's device tree

Determine the hardware configuration of your system by looking at the device tree, accessed from your system console, using the `show-devs` command. The device tree describes the hardware devices attached to each hypernode; its organization is similar to an SPP-UX file system tree.

As the device tree diverges, a route (or device path) to individual devices becomes apparent. The device path represents the type of device and where that device is located within the device tree. A full device name is a series of device paths separated by slashes.

Device path syntax is

name@address:[arguments]

where

name

Represents an address on the main system bus. Valid names for systems include:

`sd`

SCSI disk

`st`

SCSI tape

`pci1011,9`

Ethernet

`pci107e,8`

ATM

`pci1011,f`

FDDI

`pci5555,3`

HIPPI

`qlisp`

SCSI

@address

Represents a physical space unique to the device. *address* consists of two 32-bit numbers, usually in hexadecimal format, separated by a comma. The interpretation of these two numbers depends on the location of the device in the device tree.

arguments

(Optional) Passes additional information to the device's software. Valid options are specific to each device package. *argument* usually shows additional device information, such as disk partition.

Configuring FDDI into your network

Before SPP-UX can communicate over a network interface, it needs to know the interface's network address and operating characteristics. This section explains how to use the `ifconfig` command to assign a network address to your FDDI and set parameters that affect its operation.

Step 1 Determine the network address and operating characteristics you need.

Step 2 Log in as the superuser.

Step 3 Use the `ifconfig` command to test your choices from Step 1 as options on the command line.

The `ifconfig` command has the following syntax:

`/usr/sbin/ifconfig` *interface address_family* [*address* [*dest_address*]] [*parameters*]

where

interface

Is a string composed of interface type, or name, and unit number like `fddi0`. Logical unit numbers are assigned at boot time by the OpenBoot command `mkmap`. `netstat` displays the name and unit number of interfaces associated with your system.

address_family

Is the name of the protocol on which your naming scheme is based. An interface can receive transmissions in differing protocols, each of which may require separate naming schemes. Therefore, it is necessary to specify the *address_family*, which may affect interpretation of the remaining parameters on the command line. The only address family currently supported is `inet` (DARPA -Internet family).

address

Is either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

dest_address

Is the address of the destination system. It consists of either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

parameters

Is any of the operating parameters listed in Table 2.

Table 2 `ifconfig` parameter options for FDDI

Option	Description
up	Enables interface after an `ifconfig down`. Occurs automatically when setting the address on an interface. Setting this flag has no effect if the hardware is down.
down	Marks an interface. 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.
broadcast address	Specifies the address used to represent broadcasts to the network. By default, the broadcast address has a host part of all ones. Refer to "Defining the broadcast address" on page 13.
netmask *mask*	Specifies the portion of the internet address to reserve for the combined network and subnet fields. Refer to "Defining the subnet mask" on page 14.
dest_address	Specifies address of correspondent on the other end of a point-to-point link.
arp	Enables use of the Address Resolution Protocol (ARP), a mechanism for dynamically mapping between Internet addresses and physical addresses. Refer to "Using the Address Resolution Protocol (ARP)" on page 14. (Default)
-arp	Disables use of the ARP.
debug	Enables driver-dependent debugging code. This usually turns on extra console error logging.
-debug	Disables driver-dependent debugging code. (Default)
-trailers	Disables the use of a trailer link-level encapsulation. (Default)
metric *n*	Sets the routing metric of the interface to *n*. The default is 0. The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

The example below shows sample `ifconfig` options, to declare the Internet address as 130.150.60.6, the broadcast address as 130.150.60.255, the netmask as 0xffffff00, and marks the interface as up.

```
# ifconfig fddi0 inet 130.150.60.6 broadcast\
address 130.150.60.255 netmask 0xffffff00 up
```

Step 4 Verify that the system accepted the information by entering

```
# ifconfig interface
```

In response, the system displays the network address and operating characteristics of the interface. Sample output is shown below.

```
fddi0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 130.150.60.74 netmask ffffff00 broadcast 130.150.60.255
```

Step 5 Edit your /etc/rc.config.d/netconf file to add or modify the corresponding `ifconfig` variables. Table 3 shows example settings for the `ifconfig` variables in /etc/rc.config.d/netconf.

Table 3 Example settings for `ifconfig` variables

Variable	Default setting
INTERFACE_NAME[0]	fddi0
IP_ADDRESS[0]	150.99.221.6
SUBNET_MASK[0]	255.255.255.0
BROADCAST_ADDRESS[0]	null
IFCONFIG_ARGS[0]	null
LANCONFIG_ARGS[0]	null

Step 6 (Optional) Add routes through this interface using the `route` command. See "Defining gateways with the route command" on page 15 for more information.

You may edit the route variables in /etc/rc.config.d/netconf for permanent routes. Table 4 shows example settings for these `route` variables.

Table 4 Example settings for `route` variables

Variable	Default setting
ROUTE_DESTINATION[0]	default
ROUTE_MASK[0]	null
ROUTE_GATEWAY[0]	15.99.221.254
ROUTE_COUNT[0]	1
ROUTE_MTU[0]	null
ROUTE_SWITCH_ADDR[0]	null
ROUTE_ARGS[0]	null

Defining the broadcast address

All machines that communicate with each other must use the same broadcast address: either all zeros or all ones. The current standard is a broadcast host address of all ones, as in broadcast 128.194.*255.255*.

The default broadcast address contains a host part of all ones. `ifconfig` enables you to change an interface's broadcast address. Hewlett-Packard Exemplar Technical Server Family systems can accept broadcasts with a host part of all zeros (for compatibility with systems that use BSD 4.2 broadcasts) because it transmits and receives broadcasts with the address set using the mask in `ifconfig`.

If a machine on your network does not understand the broadcast address you select, some utilities, such as `rwho`, fail. If this happens, use `ifconfig` to set the broadcast address to the same broadcast address for all machines on the network.

Defining the subnet mask

Specify the mask as a single hexadecimal number with a leading 0x, or in dot notation. Ones in the subnet mask indicate bit positions to use for the combined network and subnet fields; zeros mark the positions of bits in the host field. To avoid confusion with broadcast addresses, do not use subnet numbers of all zeros or all ones. For example, specifying

```
netmask 0xffffff00
```

or

```
netmask 255.255.255.0
```

both indicate that you want 24 bits of combined network and subnet fields, and 8 bits of host number. For a class B network, this mask logically partitions your 16 bits of host number into an 8-bit subnet field and an 8-bit host field. If you do not supply a `netmask`, the mask is set according to the network class (A, B, or C with 8, 16, or 24 bits of network part, respectively) or your chosen IP address.

Using the Address Resolution Protocol (ARP)

ARP maps logical internet addresses to physical addresses by caching the logical mappings between dot notation addresses (as in the /etc/hosts file) and physical addresses. If an interface requests mapping for an address not cached, ARP queues the message that requires the mapping, then broadcasts a message on the associated network to request the address. If a response is received, the new mapping is cached, and the queued message is transmitted.

If you want to communicate with a host on the network that does not use the ARP protocol, you must use the `arp` program to manually add address mapping information to the local ARP table. The `arp` program forces caching of an ARP table entry for a specific host, so that the ARP protocol does not transmit a packet to a host to get its hardware address.

`arp` has the following syntax:

`arp -s` *hostname address* `[temp] [pub]`

where

`-s`

Creates an ARP entry for the host called *hostname*.

hostname

Specifies the remote host as listed in the /etc/hosts file. If an ARP entry already exists for hostname, the existing entry is updated with the new information.

address

Specifies the physical hardware stations address. This address is displayed by most systems at boot time and is usually printed on the network controller. You specify it as six hexadecimal numbers separated by colons, as in

```
08:00:20:06:dd:42
```

The entry will be permanent unless you also specify `temp`.

`temp`

Specifies that the *address* entry is not permanent.

`pub`

Specifies the entry is published, which means that this system will act as an ARP server responding to requests for *hostname* even though the host address is not its own.

You can check the current status of the ARP table by entering

```
# arp -a
```

`arp` has more options than are presented here. For a complete summary, refer to the arp(1M) man page.

Defining gateways with the route command

Modify the `route` variables in /etc/rc.config.d/netconf if your hypernode is used as a gateway or will use gateways.

The following shows a common use example of the `route` command where 130.168.85.254 represents the gateway:

```
# route add default 130.168.85.254
```

You may use `netstat -r` to look at the route table contents.

`route` has the following syntax:

`route add [net|host]` *destination gateway metric*

where

`add`

Adds a route.

`net|host`

Specifies the type of destination address.

Configuring FDDI

destination

Specifies the destination host system where packets will be routed. Can be a host name (official host name or alias), an Internet address in dot notation, or the keyword `default`, which represents the wildcard gateway routing. See the route(1m) man page for more information about the keyword, `default`.

gateway

Specifies the gateway through which the destination is reached. Can be a host name (official host name or alias) or an Internet address in dot notation.

metric

The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

`route` has more options than are presented here. For a complete summary, refer to the route(1M) man page.

Verifying your network's configuration

Successfully booting your system in multiuser mode indicates that you have correctly configured network interfaces. In most cases, the machine will not run in multiuser mode if you make a mistake during the configuration process. You should test the network after the system is up and running in multiuser mode.

You can verify network configuration by entering

```
# netstat -i
```

If the network is properly configured, the system displays output similar to that shown in the example below.

```
Name   Mtu   Network        Address      Ipkts  Ierrs  Opkts Oerrs  Coll
fddi0  4332  15.99.221.16   acmes        125295     0  64580     0     0
lo0    4056  loopback       localhost    4808       0  4808      0     0
```

The displayed host name and network name are specific to your installation. The name of the interface you just configured should appear. If it does not, you have a problem with your installation or your configuration of the network.

Configuring Ethernet

3

This chapter describes how to integrate a newly installed or reconfigured Ethernet interface into your SPP-UX network. The topics covered include:

- Defining an Ethernet interface
- Configuring Ethernet into your network
- Verifying your network's configuration

Note

This chapter does not explain how to install Ethernet. You must install and test the Ethernet hardware, and connect the Ethernet card to an active LAN before you begin configuration. Refer to *PCI Fast Ethernet Installation and Service Guide* for more information.

Hewlett-Packard Exemplar Technical Servers interface to an Ethernet network through PCIbus controllers. These controllers support the attachment of transceivers. Ethernet networks have a peak data rate of 100BASE-T (100 megabits per second) over segments of up to 500 meters in length.

Defining an Ethernet interface

Before booting the nodes of your HP Exemplar Technical Servers you must specify a logical unit number for each hypernode's Ethernet controller to OpenBoot. You must also determine the hardware configuration of your system by looking at the device tree.

Defining an Ethernet interface with OpenBoot

Steps 1 through 7 detail the logical unit number specification process and must be performed for each Ethernet adapter on each Exemplar hypernode.

Step 1 Determine the full device name in each HP Exemplar technical server console window. At the OpenBoot `ok` prompt, enter:

`show-devs`

`show-devs` with no argument displays all devices known on the hypernode. The two HIPPI controllers are shown below as HIPPI-A and HIPPI-B; this indicates the separate source (write) and destination (read) controllers, respectively. Each controller must appear in the `show-devs` output , as exemplified below:

```
[0:1] ok show-devs
/pci@fe,290000
/pci@fe,210000
/pci@fe,90000
/pci@fe,10000
/HP,PA80000@fc,30000
/HP,PA80000@fc,a0000
/HP,PA80000@fc,b0000
/HP,PA80000@fc,120000
/HP,PA80000@fc,130000
/HP,PA80000@fc,1a0000
/HP,PA80000@fc,1b0000
/HP,PA80000@fc,220000
/HP,PA80000@fc,230000
/HP,PA80000@fc,2a0000
/HP,PA80000@fc,2b0000
/HP,PA80000@fc,320000
/HP,PA80000@fc,330000
/HP,PA80000@fc,3a0000
/HP,PA80000@fc,3b0000
/ramdisk
/virtual-memory@0,fffa0000
/memory@0,0
/sim@0,0
/core@f0,f0000000
```

```
/pty
/openprom
/chosen
/aliases
/options
/packages
/pci@fe,290000/qlisp@1000,0
/pci@fe,290000/pci1011,9@800,0              (Ethernet)
/pci@fe,290000/qlisp@1000,0/st
/pci@fe,290000/qlisp@1000,0/sd
/pci@fe,210000/qlisp@1000,0
/pci@fe,210000/pci107e,8@0,0                     (ATM)
/pci@fe,210000/qlisp@1000,0/st
/pci@fe,210000/qlisp@1000,0/sd
/pci@fe,90000/pci1077,1020@1000,0
/pci@fe,90000/pci1011,f@0,0                     (FDDI)
/pci@fe,10000/pci5555,3@800,0                (HIPPI-A)
/pci@fe,10000/pci5555,3@0,0                  (HIPPI-B)
/core@f0,f0000000/sram@0,800000
/core@f0,f0000000/tty@0,d46000
/core@f0,f0000000/lan@0,d30000
/core@f0,f0000000/lcd@0,d4c000
/core@f0,f0000000/nvram@0,d00000
/core@f0,f0000000/rtc@0,d07ff8
/core@f0,f0000000/epuc@0,c00000
/core@f0,f0000000/emuc@0,c10000
/core@f0,f0000000/flash
/packages/firmware
/packages/dns
/packages/inet
/packages/obp-tftp
/packages/deblocker
/packages/tape-label
/packages/disk-label
/packages/lif
/packages/nfs
/packages/hfs
/packages/firmware/isp-fw-v2-10
[0:1] ok
```

Step 2 Find the device path that includes the string **pci1011,9** in the output from the show-devs command above. It appears as the output example below:

```
/pci@fe,290000/pci1011,9@800,0
```

Step 3 Enter the entire line of text for *device_string* in the following command:

ok **mkmap** *logical_unit_number device_string*

Configuring Ethernet

where

`mkmap`

Defines an SPP-UX logical-unit to physical-unit mapping. This mapping is a label for tape and network devices that cannot be labeled like disks.

logical_unit_number

Designates the logical-unit number of each Ethernet device.

You may want to use the *interface* unit number you supply when you enable your network with the `ifconfig` command. For instance, if your interface is `le0` the *logical_unit_number* should be `0`.

device _string

Represents the full Ethernet device name; `/pci@fe,290000/pci1011,9@800,0`

Your Ethernet device's logical unit number is assigned by OpenBoot at hypernode power on/reset time.

Step 4 Enter **`show-map`** to list current mappings and check your `mkmap` command.

Step 5 Make sure your current directory is root. If it is not, change to the root directory by entering **`cd /`**

Step 6 Add your new logical unit. Enter:

```
ok reset
```

Step 7 Boot the nodes of your system by entering the proper command at the OpenBoot `ok` prompt in each HP Exemplar Technical Server console window. Refer to the SPP-UX Release Notice specific to your version of operating system for more information.

When SPP-UX boots, the Ethernet driver asks OpenBoot for its logical unit number. The driver keeps track of this number.

OpenBoot's device tree

Determine the hardware configuration of your system by looking at the device tree, accessed from your system console, using the `show-devs` command. The device tree describes the hardware devices attached to each hypernode; its organization is similar to an SPP-UX file system tree.

As the device tree diverges, a route (or device path) to individual devices becomes apparent. The device path represents the type of device and where that device is located within the device tree. A full device name is a series of device paths separated by slashes.

Device path syntax is

name@address:arguments

where

name

Represents an address on the main system bus. Valid names for HP Exemplar Technical Servers include:

`sd`

SCSI disk

`st`

SCSI tape

`pci1011,9`

Ethernet

`pci107e,8`

ATM

`pci1011,f`

FDDI

`pci5555,3`

HIPPI

`qlisp`

SCSI

@address

Represents a physical space unique to the device. *address* consists of two 32-bit numbers, usually in hexadecimal format, separated by a comma. The interpretation of these two numbers depends on the location of the device in the device tree.

:arguments

(Optional) Passes additional information to the device's software. Valid options are specific to each device package. *argument* usually shows additional device information, such as disk partition.

Configuring Ethernet into your network

Before SPP-UX can communicate over a network interface, it needs to know the interface's network address and operating characteristics. This section explains how to use the `ifconfig` command to assign a network address to an Ethernet and set parameters that affect its operation.

Step 1 Determine the network address and operating characteristics you need.

Step 2 Log in as the superuser.

Step 3 Use the `ifconfig` command to test your choices from step one as options on the command line.

The `ifconfig` command has the following syntax:

`/usr/sbin/ifconfig` *interface address_family [address [dest_address]] [parameters]*

where

interface

Is a string composed of interface type, or name, and unit number like `le0`. Logical unit numbers are assigned at boot time by the OpenBoot command `mkmap`. `netstat` displays the name and unit number of interfaces associated with your HP Exemplar technical servers.

address_family

Is the name of the protocol on which your naming scheme is based. An interface can receive transmissions in differing protocols, each of which may require separate naming schemes. Therefore, it is necessary to specify the *address_family*, which may affect interpretation of the remaining parameters on the command line. The only address family currently supported is `inet` (DARPA -Internet family).

address

Is either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

dest_address

Is the address of destination system. It consists of either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

parameters

Is any of the operating parameters listed in Table 5.

Table 5 `ifconfig` parameter options for Ethernet

Option	Description
`up`	Enables interface after an `ifconfig down`. Occurs automatically when setting the address on an interface. Setting this flag has no effect if the hardware is `down`.
`down`	Marks an interface. The system does 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.
`broadcast address`	Specifies the address used to represent broadcasts to the network. By default, the broadcast address has a host part of all ones. Refer to "Defining the broadcast address" on page 26.
`netmask` *mask*	Specifies the portion of the internet address to reserve for the combined network and subnet fields. Refer to "Defining the subnet mask" on page 26.
`dest_address`	Specifies address of correspondent on the other end of a point-to-point link.
`arp`	Enables use of the Address Resolution Protocol (ARP), a mechanism for dynamically mapping between Internet addresses and physical addresses. Refer to "Using the Address Resolution Protocol (ARP)" on page 27. (Default)
`-arp`	Disables use of the ARP.
`debug`	Enables driver-dependent debugging code. This usually turns on extra console error logging.
`-debug`	Disables driver-dependent debugging code. (Default)
`-trailers`	Disables the use of a trailer link-level encapsulation. (Default)
`metric` *n*	Sets the routing metric of the interface to *n*. The default is 0. The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

Step 4 Verify that the system accepted the information.

Enter

```
# ifconfig interface
```

In response, the system displays the network address and operating characteristics of the interface. Sample output is shown below.

```
le0: flags=43<UP,BROADCAST,RUNNING>
        inet 130.150.70.3 netmask ffffff00 broadcast 130.150.60.255
        hardware address aa.00.04.00.2e.28
```

Step 5 Edit your /etc/rc.config.d/netconf file to add or modify the corresponding `ifconfig` variables. Table 6 shows example settings for the `ifconfig` variables in /etc/rc.config.d/netconf.

Table 6 Example settings for `ifconfig` variables

Variable	Default setting
INTERFACE_NAME[1]	le0
IP_ADDRESS[1]	15.99.102.124
SUBNET_MASK[1]	255.255.255.0
BROADCAST_ADDRESS[1]	null
IFCONFIG_ARGS[1]	null
LANCONFIG_ARGS[1]	null

Step 6 (Optional) Add routes through this interface using the `route` command. See "Defining gateways with the route command" on page 28 for more information.

You may edit the route variables in /etc/rc.config.d/netconf for permanent routes. Table 7 shows example settings for these `route` variables.

Table 7 Example settings for `route` variables

Variable	**Default setting**
ROUTE_DESTINATION[0]	default
ROUTE_MASK[0]	null
ROUTE_GATEWAY[0]	15.99.102.254
ROUTE_COUNT[0]	1
ROUTE_MTU[0]	null
ROUTE_SWITCH_ADDR[0]	null

Defining the broadcast address

The default broadcast address contains a host part of all ones. `ifconfig` enables you to change an interface's broadcast address. An HP Exemplar technical server can accept broadcasts with a host part of all zeros (for compatibility with systems that use BSD 4.2 broadcasts), because it transmits and receives broadcasts with the address set using the mask in `ifconfig`.

If a machine on your network does not understand the broadcast address you select, some utilities, such as `rwho`, fail. If this happens, use `ifconfig` to set the broadcast address to the same broadcast address for all machines on the network.

All machines that communicate with each other must use the same broadcast address, either all zeros or all ones. The current standard is a broadcast address of all ones, as in broadcast 128.194.255.255.

Defining the subnet mask

Ones in the subnet mask indicate bit positions to use for the combined network and subnet fields; zeros mark the positions of bits in the host field. You specify the mask as a single hexadecimal number with a leading 0x, or in dot notation. For example, specifying

```
netmask 0xffffff00
```

or

```
netmask 255.255.255.0
```

both indicate that you want 24 bits of combined network and subnet fields, and 8 bits of host number. For a class B network, this mask logically partitions your 16 bits of host number into an 8-bit subnet field and an 8-bit host field. If you do not supply a `netmask`, the mask is set according to the network class (A, B, or C with 8, 16, or 24 bits of network part, respectively) of your chosen IP address.

To avoid confusion with broadcast addresses, do not use subnet numbers of all zeros or all ones.

Using the Address Resolution Protocol (ARP)

ARP maps logical internet addresses to physical Ethernet addresses by caching the logical mappings between dot notation addresses (as in the /etc/hosts file) and physical Ethernet addresses. If an interface requests mapping for an address not cached, ARP queues the message that requires the mapping, then broadcasts a message on the associated network to request the address. If a response is received, the new mapping is cached, and the queued message is transmitted.

If you want to communicate with a network host that does not use the ARP protocol, you must use the `arp` program to manually add address mapping information to the local ARP table. The `arp` program forces caching of an ARP table entry for a specific host, so that the ARP protocol does not transmit a packet to a host to get its Ethernet address.

`arp` has the following syntax:

`arp -s` *hostname address* `[temp] [pub]`

where

`-s`

Creates an ARP entry for the host called *hostname*.

hostname

Specifies the remote host as listed in the /etc/hosts file. If an ARP entry already exists for hostname, the existing entry is updated with the new information.

address

Specifies the physical hardware stations address. This address is displayed by most systems at boot time and is usually printed on the network controller. You specify it as six hexadecimal numbers separated by colons, as in

`08:00:20:06:dd:42`

The entry will be permanent unless you also specify `temp`.

`temp`

Specifies that the *address* entry is not permanent.

`pub`

Specifies the entry is published, which means that this system will act as an ARP server responding to requests for *hostname* even though the host address is not its own.

You can check the current status of the ARP table by entering

```
# arp -a
```

arp has more options than are presented here. For a complete summary, refer to the arp(1M) man page.

Defining gateways with the route command

Modify the route variables in /etc/rc.netconfig.d/netconf if your hypernode is used as a gateway or will use gateways.

The following shows a common use example of the route command where 130.168.85.254 represents the gateway.

```
# route add default 130.168.85.254
```

You may use netstat -r to look at the route table contents.

route has the following syntax:

route add [net|host] *destination gateway metric*

where

add

Adds a route.

net|host

Specifies the type of destination address.

destination

Specifies the destination host system where packets will be routed. Can be a host name (official host name or alias), an Internet address in dot notation, or the keyword default, which represents the wildcard gateway routing. See the route(1m) man page for more information about the keyword, default.

gateway

Specifies the gateway through which the destination is reached. Can be a host name (official host name or alias) or an Internet address in dot notation.

metric

The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

route has more options than are presented here. For a complete summary, refer to the route(1M) man page.

Verifying your network's configuration

Successfully booting your HP Exemplar Technical Servers in multiuser mode indicates that you have correctly configured network interfaces. In most cases, the machine simply does not run in multiuser mode if you make a mistake during the configuration process. Of course, you should test the network after the system is up and running in multiuser mode.

To verify network configuration, enter

```
# netstat -i
```

If the network is properly configured, the system displays output similar to that shown in the following example.

```
Name    Mtu    Network      Address    Ipkts     Ierrs    Opkts     Oerrs   Coll
le0     1500   acme-net     acmes      1520512   0        1327049   0       0
```

The displayed host name and network name are specific to your installation. The name of the interface you just configured should appear. If it does not, you have a problem with your installation or your configuration of the network.

The espeed(1m) utility can be used to display or modify the controller settings. The controller negotiates to the highest possible speed during boot.

Configuring HIPPI

4

This chapter describes how to integrate a newly installed or reconfigured High Performance Parallel Interface (HIPPI) into SPP-UX. The topics covered include:

- Configuring HIPPI into your network
- Defining the subnet mask
- Verifying your network's configuration
- Tuning network performance

HIPPI is a high-performance parallel interface. HIPPI is a copper-based data communications standard capable of transferring data at 800 Mbit/sec over 32 parallel lines or 1.6 Gbit/sec over 64 parallel lines.

Note

This chapter does not explain how to install HIPPI. You must install and test the HIPPI hardware before you begin configuration. For additional information on installing HIPPI refer to the *PCI ATM and HIPPI Installation and Service Guide: Exemplar S-Class and X-Class Servers.* (A4716-90017)

Contrary to FDDI, Ethernet and ATM; HIPPI does not require an OpenBoot Program (OBP) `mkmap` command. The unit number is determined by the hardware jumpers. This difference is reflected in the following configuration section.

Configuring HIPPI into your network

Before SPP-UX can communicate over a network interface, it needs to know the interface's network address and operating characteristics. This section explains how to use the `ifconfig` command to assign a network address to a HIPPI and set parameters that affect its operation.

Step 1 Determine the network address and operating characteristics you need.

Step 2 Log in as the superuser.

Step 3 Use the `ifconfig` command to test your choices from Step 1 as options on the command line.

The `ifconfig` command has the following syntax:

`/usr/sbin/ifconfig` *interface address_family [address [dest_address]] [parameters]*

where

interface

Is a string composed of interface type, or name, and unit number like `hippi0`. Logical unit numbers are determined by hardware jumpers. `netstat` displays the name and unit number of interfaces.

address_family

Is the name of the protocol on which your naming scheme is based. An interface can receive transmissions in differing protocols, each of which may require separate naming schemes. Therefore, it is necessary to specify the *address_family*, which may affect interpretation of the remaining parameters on the command line. The only address family currently supported is `inet` (DARPA -Internet family).

address

Is either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

dest_address

Is the address of the destination system. It consists of either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

parameters

Is any of the operating parameters listed in Table 8. You must specify the `-arp` parameter option for HIPPI.

Table 8 `ifconfig` parameter options for HIPPI

Option	Description
`up`	Enables interface after an `ifconfig down`. Occurs automatically when setting the address on an interface. Setting this flag has no effect if the hardware is `down`.
`down`	Marks an interface. 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.
`broadcast address`	Specifies the address used to represent broadcasts to the network. By default, the broadcast address has a host part of all ones.
`netmask` *mask*	Specifies the portion of the internet address to reserve for the combined network and subnet fields. Refer to "Defining the subnet mask" on page 36.
`dest_address`	Specifies address of correspondent on the other end of a point-to-point link.
`arp`	Enables use of the Address Resolution Protocol (ARP), a mechanism for dynamically mapping between Internet addresses and physical addresses.
`-arp`	Disables use of the ARP.
`debug`	Enables driver-dependent debugging code. This usually turns on extra console error logging.
`-debug`	Disables driver-dependent debugging code. (Default)
`-trailers`	Disables the use of a trailer link-level encapsulation. This is the default.
`metric` *n*	Sets the routing metric of the interface to *n*. The default is 0. The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

Step 4 Verify that the system accepted the information.

Enter

```
# ifconfig hippi0
```

In response, the system displays the network address and operating characteristics of the interface. Sample output is shown below.

```
hippi0: flags=43<UP,BROADCAST NOTRAILERS,RUNNING NOARP>
      inet 130.168.39.1 netmask ffffff00 broadcast 130.168.39.255
```

Step 5 Edit your /etc/rc.config.d/netconf file to add or modify the corresponding `ifconfig` variables. Table 9 shows example settings for the `ifconfig` variables in the /etc/rc.config.d/netconf file.

Table 9 Example settings for `ifconfig` variables

Variable	Default setting
INTERFACE_NAME[2]	hippi0
IP_ADDRESS[2]	130.168.39.1
SUBNET_MASK[2]	255.255.255.0
BROADCAST_ADDRESS[2]	null
IFCONFIG_ARGS[2]	-arp
LANCONFIG_ARGS[2]	null

Step 6 (Optional) Add routes through this interface using the `route` command. Refer to "Defining gateways with the route command" on page 36 for more information.

You can also edit the route variables in /etc/rc.config.d/netconf manually. Table 10 shows example settings for the `route` variables in /etc/rc.config.d/netconf.

Table 10 Example settings for `route` variables

Variable	Default setting
ROUTE_DESTINATION[1]	batman-h
ROUTE_MASK[1]	null
ROUTE_GATEWAY[1]	batman-h
ROUTE_COUNT[1]	0
ROUTE_MTU[1]	65280
ROUTE_SWITCH_ADDR[1]	0x1
ROUTE_ARGS[1]	null
ROUTE_DESTINATION[2]	zonk-h
ROUTE_MASK[2]	null
ROUTE_GATEWAY[2]	batman-h
ROUTE_COUNT[2]	null
ROUTE_MTU[2]	65280
ROUTE_SWITCH_ADDR[2]	0x2
ROUTE_ARGS[2]	null

Defining the subnet mask

Ones in the subnet mask indicate bit positions to use for the combined network and subnet fields; zeros mark the positions of bits in the host field. To avoid confusion with broadcast addresses, do not use subnet numbers of all zeros or all ones. You specify the mask as a single hexadecimal number with a leading 0x, or in dot notation. For example, specifying

```
netmask 0xffffff00
```

or

```
netmask 255.255.255.0
```

both indicate that you want 24 bits of combined network and subnet fields, and 8 bits of host number. For a class B network, this mask logically partitions your 16 bits of host number into an 8-bit subnet field and an 8-bit host field. If you do not supply a `netmask`, the mask is set according to the network class (A, B, or C with 8, 16, or 24 bits of network part, respectively) of your chosen IP address.

Defining gateways with the route command

If HIPPI is used for point-to-point connections, you do not need to use the `route` command. When a switch is involved, however, the `route` command indicates how to get to each destination. The format of the `route` command for HIPPI is:

\# **route addhippi host** *dest gateway metric mtu ifield*

where

`addhippi`

Adds a route.

dest

Is the destination host

gateway

Is the address of the host on which the HIPPI interface is installed

metric

A count that indicates the number of hops to the destination

mtu

Is maximum transmission unit (default is 65280) *ifield*

Is a 32–bit entity (I-field) that is defined in RFC 1374. Bits relevant to the `route` command are as follows:

Path selection (bits 26, 25) can be 00, 01, or 11 (binary). 00 (source route mode) indicates that the I-field bits 23-00 contain a 24–bit source route; 01 or 11 (logical address mode) indicate that bits 23-00 contain 12–bit source and destination Addresses. The value 11 is meaningful when more than one route exists from a source to a destination; it allows the switch to choose the route. Use of 01 forces the switch to always use the same route for the same source/destination pair.

If logical address mode is used, the source address field (bits 23-12) is not used, and the destination address field (bits 11-0) contains the switch address of the destination. If source route mode is used, the routing control field (bits 23-00) contains the route to the destination.

Source routing examples

If you assume host A and host B are connected to port 0 and port 1 of a switch respectively, the path selection bits should be 00 for source routing. The `route` commands for host A would look similar to the following:

```
# route addhippi host addr_A addr_A 0 65280 0
```

```
# route addhippi host addr_B addr_A 0 65280 1
```

The following is an example of the `route` command and shows output from a follow-up `netstat -r` command.

```
# route addhippi host zap-h zap-h 0 65280 0x01000006

# netstat -r

Routing tables
Destination     Gateway                    Flags     Refs     Use     Interface
localhost       localhost                  UH        3        4415    lo0
zap-h           zap-h                      UG        678      0       hippi0
15.99.221       zap-h                      U         3620     0       hippi0
```

Configuring HIPPI

Logical routing examples

If more than one switch occurs between host A and host B, each port is used for the path selection bits. Assume host A is on port 0 of switch 1, switch 2 is on port 1 of switch 1 and host B is on port 4 of switch 2. The `route` command from host A to host B would look similar to the following:

```
# route addhippi host addr_B addr_A 1 65280 0x000041
```

As the I-field is passed from switch 1, the routing control bits are shifted right by 4 bits.

Verifying your network's configuration

Successfully booting your system in multiuser mode indicates that you have correctly configured network interfaces. In most cases, the machine will not run in multiuser mode if you make a mistake during the configuration process. You should test the network after the system is up and running in multiuser mode.

To verify network configuration, enter

```
# netstat -i
```

If the network is properly configured, the system displays output similar to that shown in the following example.

```
Name    Mtu    Network         Address   Ipkts  Ierrs    Opkts   Oerrs  Coll
hippi0  65280  130.168.39.4    zap-h     538    0        1       0      0
```

The displayed host name and network name are specific to your installation. The name of the interface you just configured should appear. If it does not, you have a problem with your installation or the configuration of your network. The troubleshooting chapter contains more complete instructions for using `netstat`.

Configuring ATM

5

This chapter describes how to integrate a newly installed or reconfigured Asynchronous Transfer Mode (ATM) interface into SPP-UX. The topics covered include:

- Defining an ATM interface
- Configuring ATM into your network
- Defining the subnet mask
- Verifying your network's configuration

Note

This chapter does not explain how to install ATM. You must install and test the ATM hardware, and connect the ATM card to an active switch before you begin configuration. For additional information on installing HIPPI refer to the *PCI ATM and HIPPI Installation and Service Guide: Exemplar S-Class and X-Class Servers. (A4716-90017)*

ATM is the technique for transport, multiplexing, and switching that provides the high degree of flexibility required by B-ISDN. ATM is a connection-oriented protocol that employs fixed-size packets with a 5-byte header and 48 bytes of information.

Defining an ATM interface

Before booting the nodes of your HP Exemplar Technical Servers, you must specify a logical unit number for each hypernode's ATM controller to OpenBoot. You must also determine the hardware configuration of your system by looking at the device tree.

Defining an ATM interface with OpenBoot commands

Steps 1 through 7 detail the logical unit number specification process and must be performed for each ATM adapter on each Exemplar hypernode.

Step 1 Determine the full device name in each HP Exemplar Technical Server console window. At the OpenBoot `ok` prompt, enter:

```
show-devs
```

`show-devs` with no argument displays all devices known on the hypernode. The two HIPPI controllers are shown below as HIPPI-A and HIPPI-B; this indicates the separate source (write) and destination (read) controllers, respectively. Each controller must appear in the `show-devs` output, as exemplified below:

```
[0:1] ok show-devs
/pci@fe,290000
/pci@fe,210000
/pci@fe,90000
/pci@fe,10000
/HP,PA80000@fc,30000
/HP,PA80000@fc,a0000
/HP,PA80000@fc,b0000
/HP,PA80000@fc,120000
/HP,PA80000@fc,130000
/HP,PA80000@fc,1a0000
/HP,PA80000@fc,1b0000
/HP,PA80000@fc,220000
/HP,PA80000@fc,230000
/HP,PA80000@fc,2a0000
/HP,PA80000@fc,2b0000
/HP,PA80000@fc,320000
/HP,PA80000@fc,330000
/HP,PA80000@fc,3a0000
/HP,PA80000@fc,3b0000
/ramdisk
/virtual-memory@0,fffa0000
/memory@0,0
/sim@0,0
/core@f0,f0000000
```

```
/pty
/openprom
/chosen
/aliases
/options
/packages
/pci@fe,290000/qlisp@1000,0
/pci@fe,290000/pci1011,9@800,0           (Ethernet)
/pci@fe,290000/qlisp@1000,0/st
/pci@fe,290000/qlisp@1000,0/sd
/pci@fe,210000/qlisp@1000,0
/pci@fe,210000/pci107e,8@0,0                  (ATM)
/pci@fe,210000/qlisp@1000,0/st
/pci@fe,210000/qlisp@1000,0/sd
/pci@fe,90000/pci1077,1020@1000,0
/pci@fe,90000/pci1011,f@0,0                  (FDDI)
/pci@fe,10000/pci5555,3@800,0            (HIPPI-A)
/pci@fe,10000/pci5555,3@0,0              (HIPPI-B)
/core@f0,f0000000/sram@0,800000
/core@f0,f0000000/tty@0,d46000
/core@f0,f0000000/lan@0,d30000
/core@f0,f0000000/lcd@0,d4c000
/core@f0,f0000000/nvram@0,d00000
/core@f0,f0000000/rtc@0,d07ff8
/core@f0,f0000000/epuc@0,c00000
/core@f0,f0000000/emuc@0,c10000
/core@f0,f0000000/flash
/packages/firmware
/packages/dns
/packages/inet
/packages/obp-tftp
/packages/deblocker
/packages/tape-label
/packages/disk-label
/packages/lif
/packages/nfs
/packages/hfs
/packages/firmware/isp-fw-v2-10
[0:1] ok
```

Step 2 Find the device path that includes the string **pci107e,8** in the output from the show-devs command above. It appears as the output example below:

```
/pci@fe,210000/pci107e,8@0,0
```

Step 3 Enter the entire line of text as *device_string* in the following command:

ok **mkmap** *logical_unit_number device_string* where

`mkmap`

Defines an SPP-UX logical-unit to physical-unit mapping. This mapping is a label for tape and network devices that cannot be labeled like disks.

logical_unit_number

Designates the logical-unit number of each ATM device.

You may want to use the *interface* unit number you supply when you enable your network with the `ifconfig` command. For instance, if your interface is `ia0`, the *logical_unit_number* should be `0`.

device_string

Represents the full ATM device name; `/pci@fe,210000/pci107e,8@0,0`

Your ATM device's logical unit number is assigned by OpenBoot at hypernode power on/reset time.

Step 4 Enter **`show-map`** to list current mappings and check your `mkmap` command.

Step 5 Make sure your current directory is root. If it is not, change to the root directory by entering **`cd /`**

Step 6 Add your new logical unit. Enter:

```
ok reset
```

Step 7 Boot the nodes of your system by entering the proper command at the OpenBoot `ok` prompt in each HP Exemplar Technical Server console window. Refer to the SPP-UX Release Notice specific to your version of operating system for more information.

When SPP-UX boots, the ATM driver asks OpenBoot for its logical unit number. The driver keeps track of this number.

OpenBoot's device tree

Determine the hardware configuration of your system by looking at the device tree, accessed from your system console, using the `show-devs` command. The device tree describes the hardware devices attached to each hypernode; its organization is similar to an SPP-UX file system tree.

As the device tree diverges, a route (or device path) to individual devices becomes apparent. The device path represents the type of device and where that device is located within the device tree. A full device name is a series of device paths separated by slashes.

Device path syntax is

name@address : *arguments*

where

name

Represents an address on the main system bus. Valid names for HP Exemplar Technical Servers include:

`sd`

SCSI disk

`st`

SCSI tape

`pci1011,9`

Ethernet

`pci107e,8`

ATM

`pci1011,f`

FDDI

`pci5555,3`

HIPPI

`qlisp`

SCSI

@address

Represents a physical space unique to the device. *address* consists of two 32-bit numbers, usually in hexadecimal format, separated by a comma. The interpretation of these two numbers depends on the location of the device in the device tree.

: *arguments*

(Optional) Passes additional information—such as disk partition—to the device's software. Valid options are specific to each device package.

Configuring ATM into your network

Before SPP-UX can communicate over a network interface, it needs to know the interface's network address and operating characteristics. This section explains how to use the `ifconfig` command to assign a network address to an ATM and set parameters that affect its operation.

Step 1 Boot to single user mode by entering:

`ok` **`boot -s`**

Step 2 Determine the network address and operating characteristics you need.

Step 3 Edit the /etc/hosts file:

`%` **`cd /etc`**

`%` **`vi hosts`**

and add a unique host name for the ATM interface as shown in the following example:

Internet address atmhostname

Step 4 Test the `ifconfig` command with your choices from Step two as options on the command line.

The `ifconfig` command has the following syntax:

`/usr/sbin/ifconfig` *interface address_family* [*address* [*dest_address*]] [*parameters*]

where

interface

Is a string composed of interface type, or name, and unit number like `ia0`. Logical unit numbers are assigned at boot time by the OpenBoot command `mkmap`.
`netstat` displays the name and unit number of interfaces associated with your system.

address_family

Is the name of the protocol on which your naming scheme is based. An interface can receive transmissions in differing protocols, each of which may require separate naming schemes. Therefore, it is necessary to specify the *address_family*, which may affect interpretation of the remaining parameters on the command line. The only address family currently supported is `inet` (DARPA -Internet family).

address

Is either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

dest_address

Is the address of the destination system. It consists of either a host name present in the host name database or a DARPA Internet address expressed in Internet standard dot notation.

parameters

Is any of the operating parameters listed in Table 11. You must specify the `-arp` parameter option for ATM.

Table 11 `ifconfig` parameter options for ATM

Option	Description
`up`	Enables interface after an `ifconfig down`. Occurs automatically when setting the address on an interface. Setting this flag has no effect if the hardware is `down`.
`down`	Marks an interface. 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.
`broadcast address`	Specifies the address used to represent broadcasts to the network. By default, the broadcast address has a host part of all ones.
`netmask` *mask*	Specifies the portion of the internet address to reserve for the combined network and subnet fields. Refer to "Defining the subnet mask" on page 48.
`dest_address`	Specifies address of correspondent on the other end of a point-to-point link.
`arp`	Enables use of the Address Resolution Protocol (ARP), a mechanism for dynamically mapping between Internet addresses and physical addresses.
`-arp`	Disables use of the ARP.
`debug`	Enables driver-dependent debugging code. This usually turns on extra console error logging.
`-debug`	Disables driver-dependent debugging code. (Default)

Table 11 `ifconfig` parameter options for ATM **(continued)**

Option	Description
`-trailers`	Disables the use of a trailer link-level encapsulation. (Default).
`metric` *n*	Sets the routing metric of the interface to *n*. The default is 0. The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as additional hops to the destination network or host.

Step 5 Verify that the system accepted the information by entering the following:

`#` **`ifconfig`** *interface*

In response, the system displays the network address and operating characteristics of the interface as shown in the following example:

```
ia0: flags=43<UP,BROADCAST,RUNNING,NOARP>
            inet 130.150.70.3 netmask ffffff00 broadcast 130.150.70.255
            hardware address aa.00.04.00.2e.28
```

Step 6 Edit your /etc/rc.config.d/netconf file to add or modify the corresponding `ifconfig` variables. Table 12 shows example settings for the `ifconfig` variables in the /etc/rc.config.d/netconf file.

Table 12 Example settings for `ifconfig` variables

Variable	Default setting
INTERFACE_NAME[3]	ia,8
IP_ADDRESS[3]	130.150.70.3
SUBNET_MASK[3]	255.255.255.0
BROADCAST_ADDRESS[3]	null
IFCONFIG_ARGS[3]	-arp
LANCONFIG_ARGS[3]	null

Step 7 Edit the /etc/rc.config.d/netconf file and set the PVCARP_FILE variable.

Step 8 The value of the PVCARP_FILE variable is the definition file where each line is in the format of the `/usr/sbin/pvcarp -s` version of the `pvcarp` command. The recommended setting for PVCARP_FILE is /etc/atmarps.

`pvcarp` has more options than are presented here, -s for instance. For a complete summary, refer to the pvcarp(1M) man page.

Step 9 Edit or create the /etc/atmarps file with the following command:

```
% vi atmarps
```

Add any other hosts, link numbers, and VCI's that should be included to the file:

otheratmhost atm_addr

Defining the subnet mask

Ones in the subnet mask indicate bit positions to use for the combined network and subnet fields; zeros mark the positions of bits in the host field. You specify the mask as a single hexadecimal number with a leading 0x, or in dot notation. For example, specifying

```
netmask 0xffffff00
```

or

```
netmask 255.255.255.0
```

both indicate that you want 24 bits of combined network and subnet fields, and 8 bits of host number. For a class B network, this mask logically partitions your 16 bits of host number into an 8-bit subnet field and an 8-bit host field. If you do not supply a `netmask`, the mask is set according to the network class (A, B, or C with 8, 16, or 24 bits of network part, respectively) of your chosen IP address.

To avoid confusion with broadcast addresses, do not use subnet numbers of all zeros or all ones.

Verifying your network's configuration

Successfully booting your system in multiuser mode indicates that you have correctly configured network interfaces. In most cases, the machine will not run in multiuser mode if you make a mistake during the configuration process.

Step 1 Enter the `pvcarp` command with the `-a` option. In response, the system displays the network address and operating characteristics of the interface as shown in the following example:

```
# pvcarp -a
Device                 IP  Address         Link    VC         Rate(mbits)  Encap
--------------------  -----------------  ----  -------     -----------  -----
otheratmhost          200.200.200.1      0     100        BR           ipllc
```

Step 2 Enter

```
# netstat -r
```

If the network is properly configured, the system displays output similar to that shown in following example:

```
Routing  tables
Destination         Gateway          Flags           Refs          Use  Interface
otheratmhost        atmhostname      UH               0             0    ia0
default             xxxxxxxxx        UG               17            0    fddi0
???.???.???         hostname         U                0             0    fddi0
200.200.200         atmhostname      U               241            0    ia0
```

Step 3 Test the connections to other hosts using the `ping` command.

Enter

```
# ping otheratmhost
```

If the network is working properly, the system displays output similar to that shown in following example:

```
PING  otheratmhost:  64  byte  packets
        64  bytes  from  130.168.190.1:  icmp_seq=0.  time=4.  ms
        64  bytes  from  130.168.190.1:  icmp_seq=1.  time=1.  ms
        64  bytes  from  130.168.190.1:  icmp_seq=2.  time=1.  ms
        64  bytes  from  130.168.190.1:  icmp_seq=3.  time=2.  ms

              ----need-a  PING  Statistics----
        4  packets  transmitted,  4  packets  received,  0%  packet  loss
        round-trip  (ms)    min/avg/max  =  1/2/4 ^C
```

`ping` sends one datagram per second, and prints one line of output for every ECHO_RESPONSE returned. No output is produced if the remote host fails to respond. `ping` also reports round-trip times and packet loss statistics.

Troubleshooting network interfaces

6

This chapter provides flowcharts to help you diagnose and solve network interface problems. Network interface problems can include runaway jobs, missing daemons, lack of kernel memory, bad connections, and unplugged transceivers.

The topics covered in this chapter include:

- Network interface connections
- Network interface configuration
- Network level loopback
- Transport level loopback

Table 13 summarizes the types of symbols used in the flowcharts.

Table 13 Flowchart symbol summary

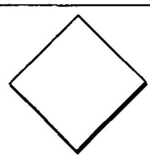
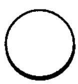
Symbol	Description
	Contains a decision point in the process flow.
	Contains a command or action you need to perform.
	Contains a reference to a specific step in the flowchart (if alphabetic) or another flowchart (if numeric).

Table 13 Flowchart symbol summary **(continued)**

Symbol	Description
	Suggests diagnostic procedures.
	Suggests contacting HP Convex TAC.

Network hardware connections

Use flowchart 1 to check that the hardware connections between your system and the network interface are connected and operational.

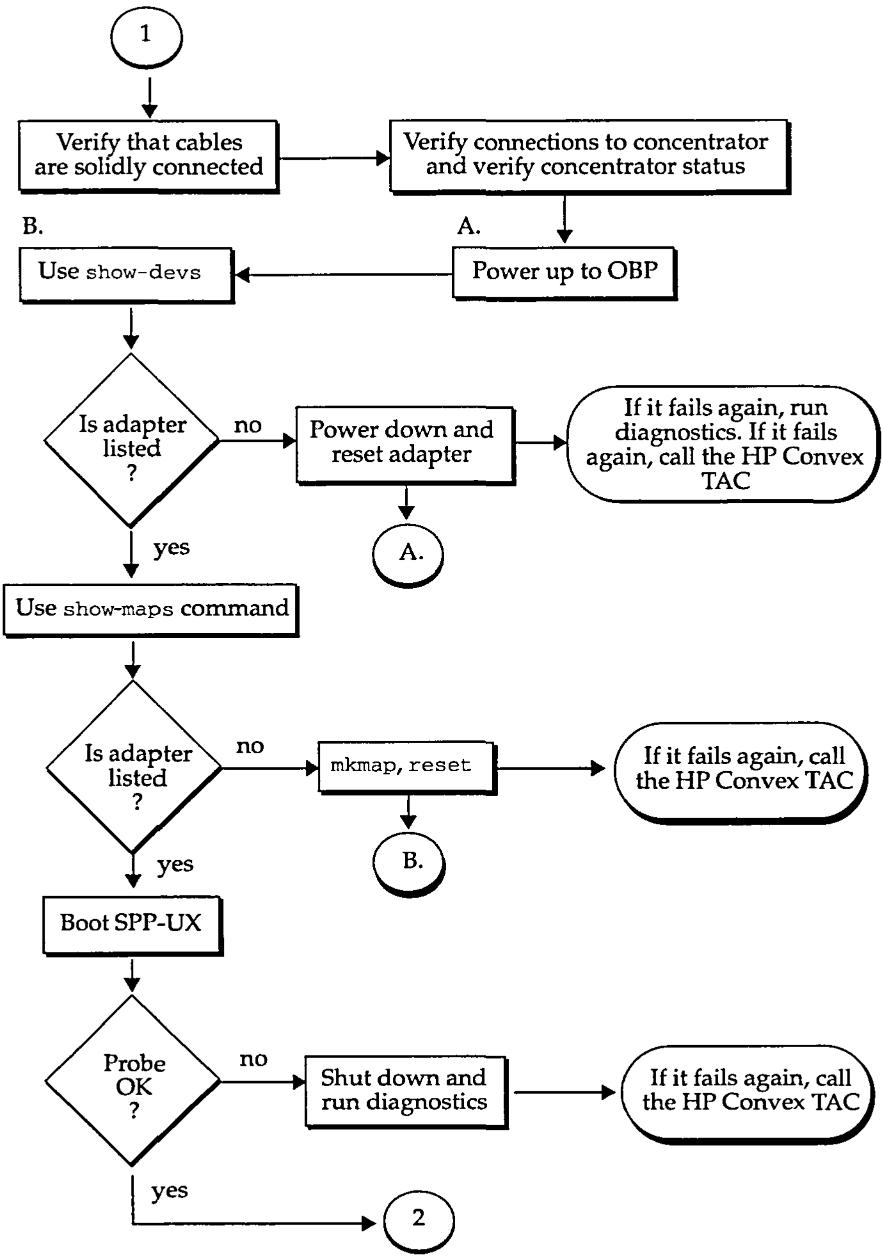


Figure 1 Flowchart 1: Network hardware connections

Network interface configuration

Use flowcharts 2 and 3 to verify the configuration of the network interface using the `ifconfig` command.

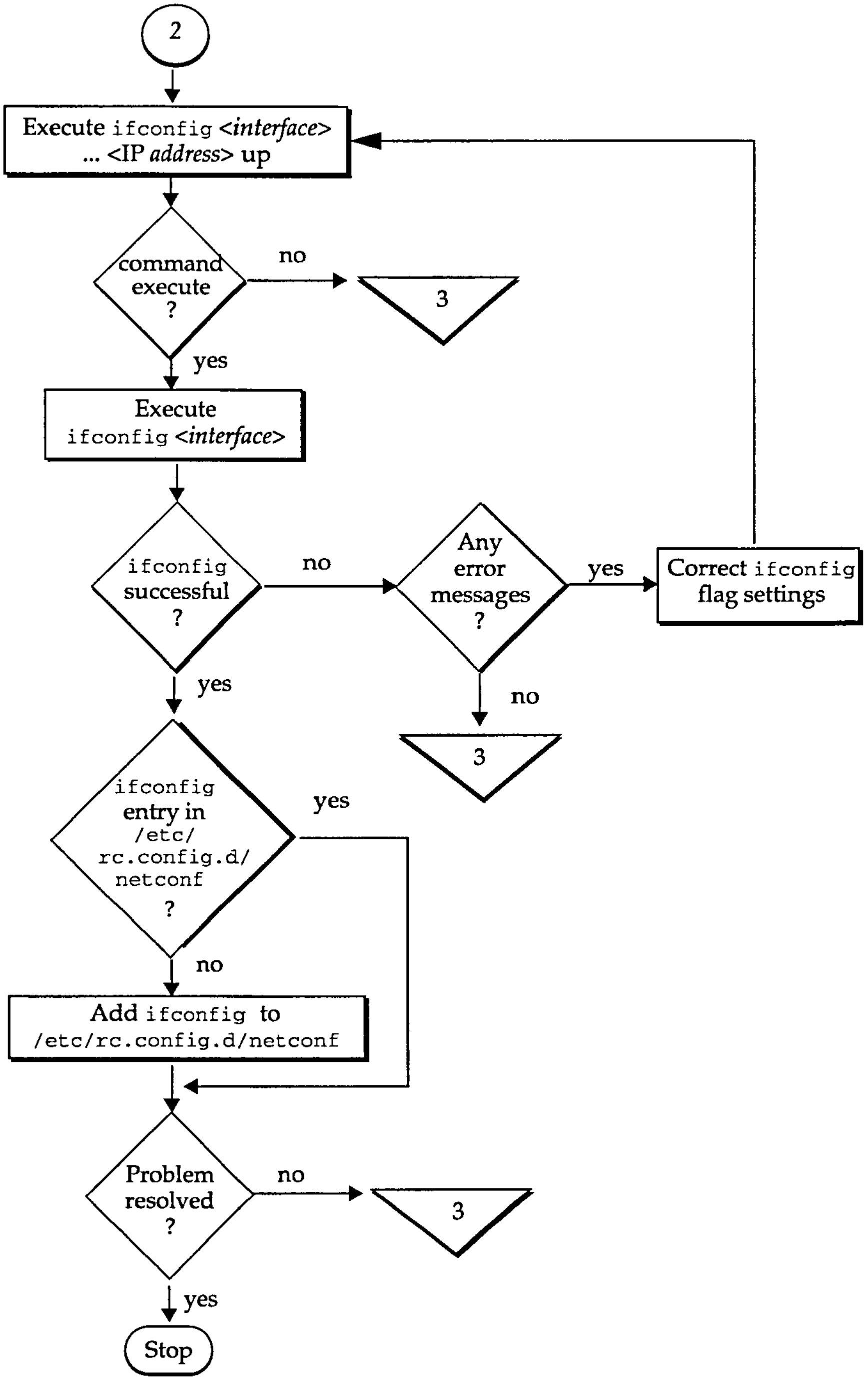


Figure 2 Flowchart 2: Network interface configurations

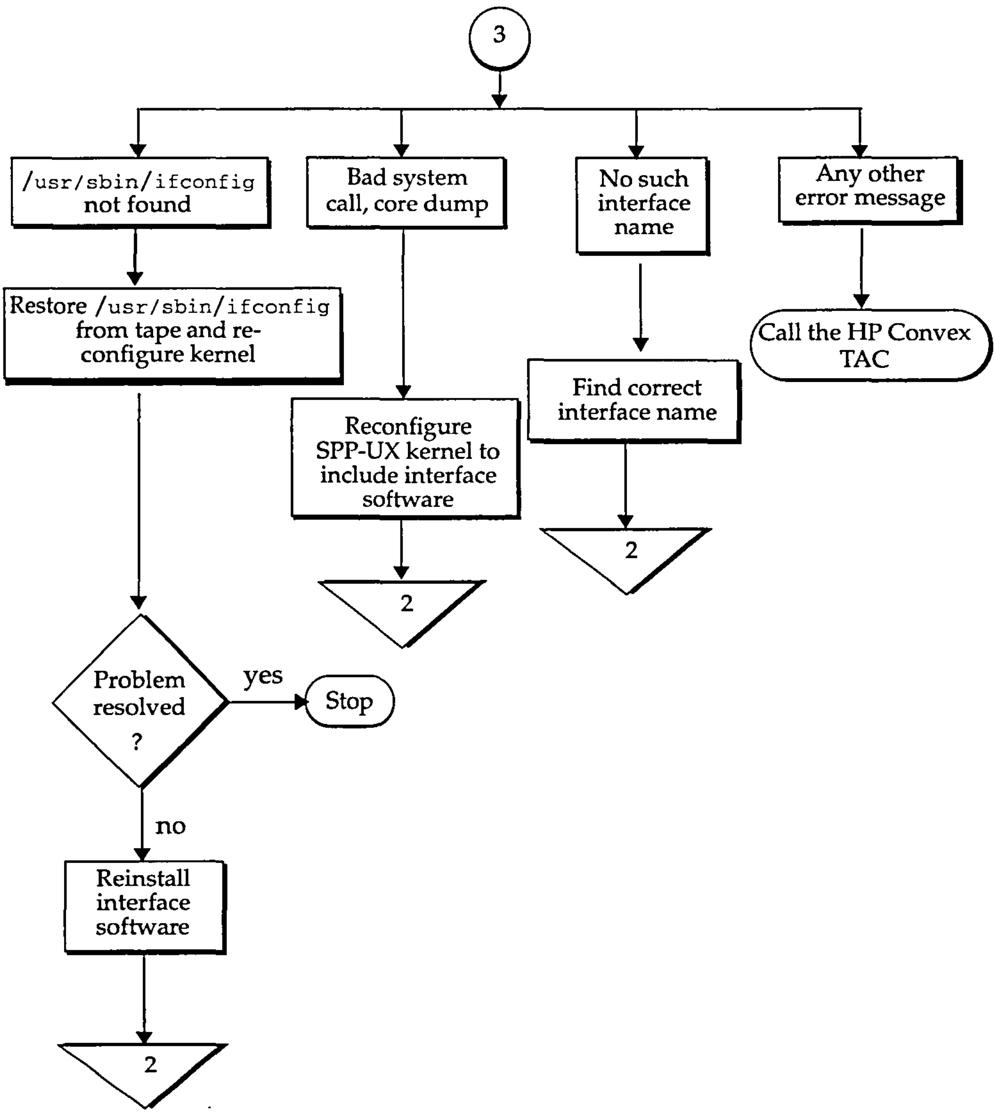


Figure 3 Flowchart 3: Network interface configurations

Network level loopback

Use flowcharts 4 and 5 to check roundtrip communication between network layers on the source and target host using the `ping` command.

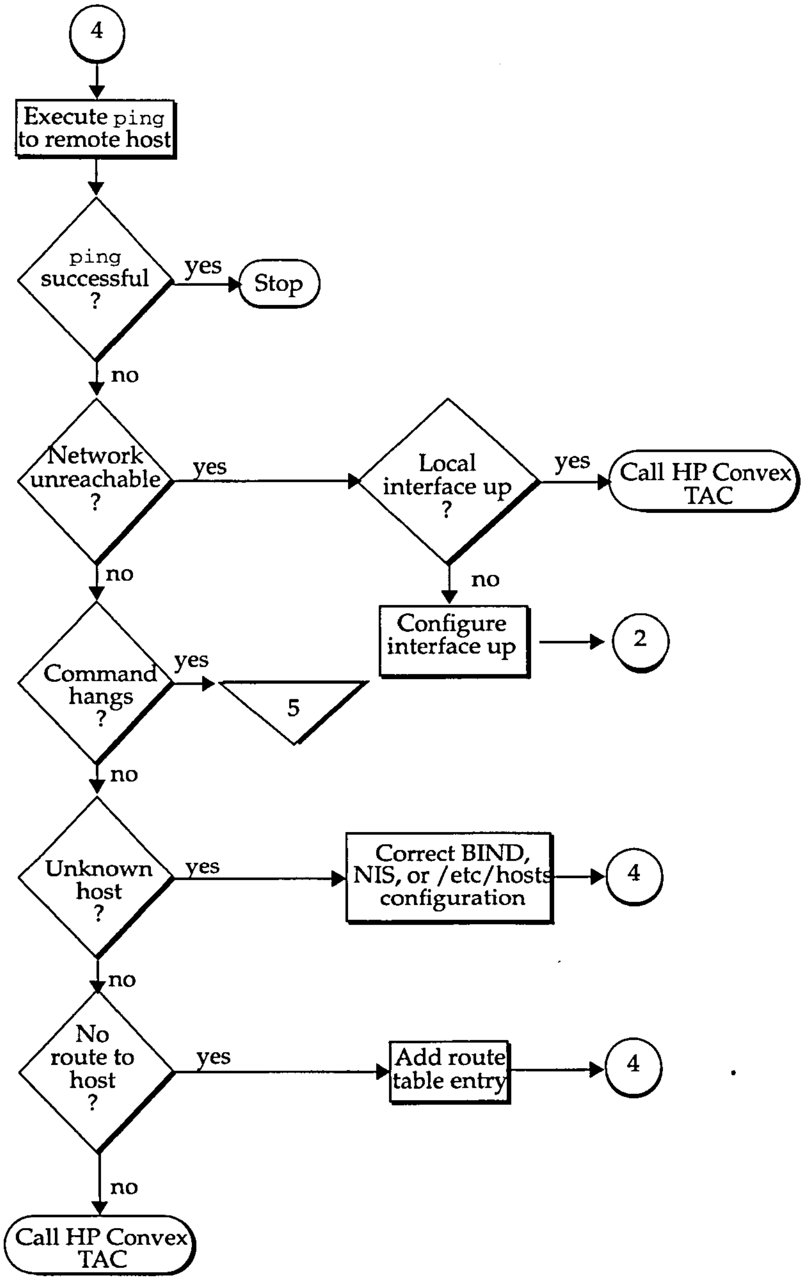


Figure 4 Flowchart 4: Network level loopback

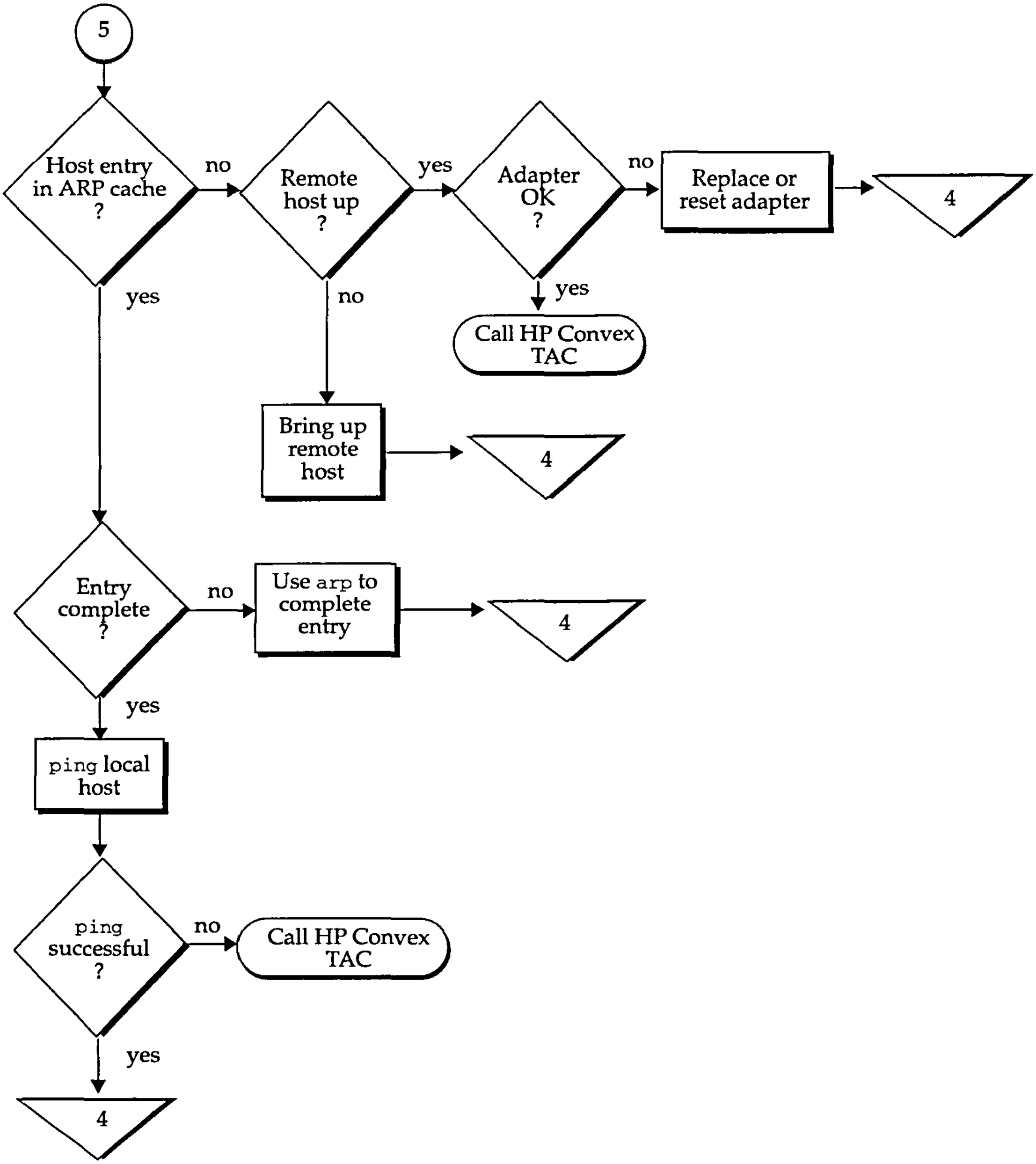


Figure 5 Flowchart 5: Network level loopback

Transport level loopback

Use flowchart 6 to check roundtrip communication between transport layers on the source and target host using the `telnet` and `ftp` commands.

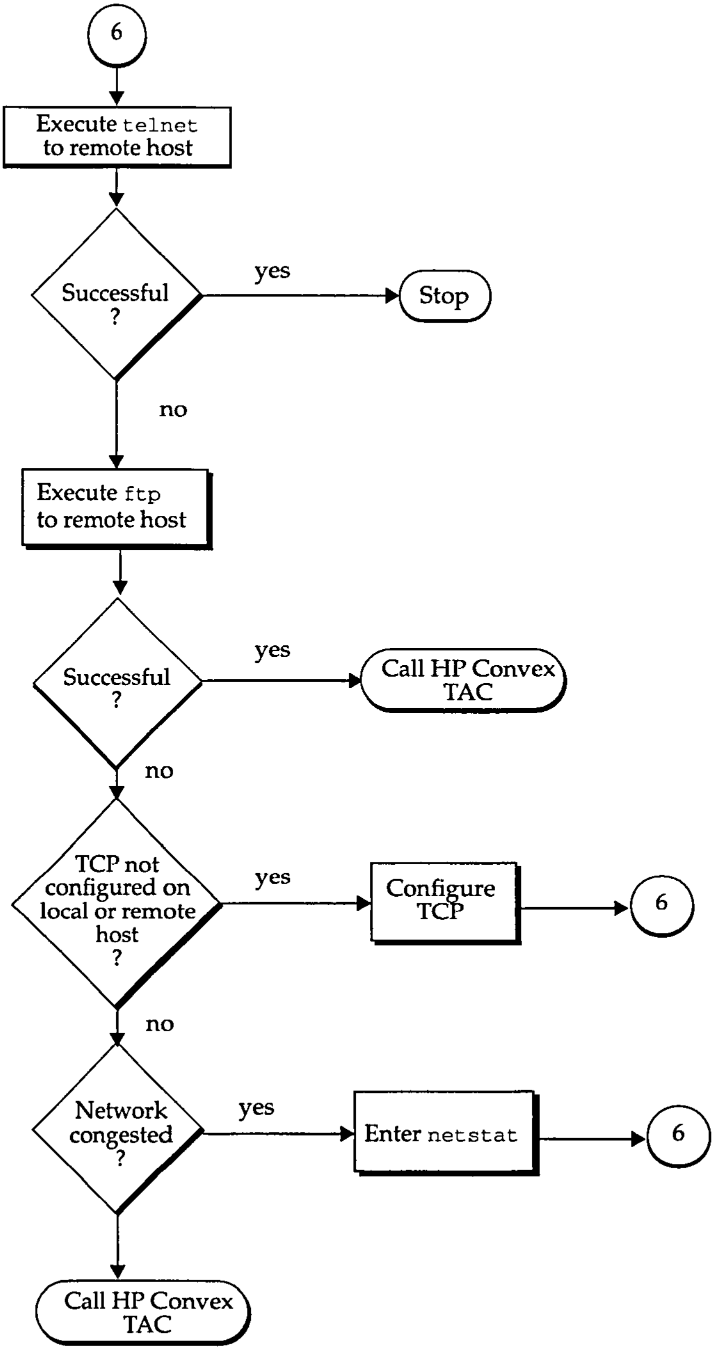


Figure 6 Flowchart 6: Transport level loopback

Tuning system parameters

A

This appendix describes parameters that you can tune to improve system throughput and performance and streamline network protocols. These parameters are stored in /stand/spp3/tunables. If you change the values for these parameters, the new values take effect the next time you boot the system.

When the system boots, the initialization process reads a file containing parameters that affect the way SPP-UX handles CPUs and peripheral devices. Among other uses, these boot-time parameters enable you to optimize network performance.

Most network-related boot-time parameters apply to networking in general, rather than to a specific type of network interface; a few are interface specific.

Note

Tailoring parameters to improve performance for one interface controller may degrade performance in another. You should consider all changes carefully to prevent over subscription of system resources (for example, I/O space).

Each of the following tables describes the parameters controlling a particular component of SPP-UX. The components include:

- Microkernel
- Server
- Emulator
- Route/ARP server

Microkernel parameters

Table 14 shows the microkernel parameters that are tunable.

Table 14 Microkernel parameters

Parameter	Range	Default	Description
HSnet, node complex data allocation	0 | 16	16	Per node contribution of global-shared memory in Mbytes[1].
HSnet, node complex data requirement	0 - 16	1	Per node global-shared memory used for receives in Mbytes. If interface controllers reside on the node, add 1 Mbyte per Ethernet controller, 1 Mbyte per FDDI controller, 2 Mbytes per ATM controller, and 3 Mbytes per HIPPI controller.
HSnet, node local data allocation	1-0x800000	0x400000	Per node memory used for transmits in bytes.
HSnet, loopback_mtu	1024 - 64512	4056	Maximum transmission unit for loopback in bytes.
HSnet, protect_io_buf	0 | 1	0	Enable noncoherent I/O debugging.
HSnet, loopback node	0 - 15	0	Node that the loopback pseudo-driver resides on.
fddi<n>, Max Receive	1 - 256	256	Maximum number of outstanding receives.
fddi<n>, Max Transmit	1 - 256	256	Maximum number of outstanding transmits.
fddi<n>, Mac_t_req	50000 - 2162688	100000	Requested token rotation time in milliseconds.
fddi<n>, Full duplex	0 | 1	0	Full duplex.
hippi<n>, Max Receive	1-256	256	Maximum number of outstanding receives.
hippi<n>, Max Transmit	1-256	256	Maximum number of outstanding transmits.
hippi<n>, Direct Connection	0 | 1	0	Point-to-point versus switch.
ia<n>, Packet Buffer Size	10240 - 65568	10240	Maximum product data unit.
ia<n>, Max I/Os	100 - 4096	300	Maximum number of controller I/Os.
ia<n>, Reassembly Descriptor Percentage	1-99	75	Percentage of descriptors used by receives

Table 14 Microkernel parameters **(continued)**

Parameter	Range	Default	Description
le<n>, Auto_negotiation_poll	0 - 900	30	Poll adapter to see if negotiation has occurred. Frequency in seconds. Zero is never polled.
le<n>, Max_receive_ descriptors	2 - 256	191	Maximum number of outstanding receives on adapter.
le<n>, Max_transmit_ descriptors	2 - 256	64	Maximum number of outstanding transmits on adapter.
Sbus Parameters			
fddi<n>, Fifo Size	1 - 16	1	Number of pages per I/O channel. (Sbus)
fddi<n>, Recieve	1 - 99	75	Percentage of channels used by receives. (Sbus)
hippi<n>, Fifo Size	1 - 16	4	Number of pages per I/O channel. (Sbus)
hippi<n>, Direct Connection	0 | 1	0	Point-to-point versus switch. (Sbus)
ia<n>, Fifo Size	1 - 16	2	Number of pages per I/O channel. (Sbus)
ia<n>, Recieve	1 - 99	75	Percentage of channels used by receives. (Sbus)
ia<n>, Packet Buffer Size	10240 - 65568	10240	Maximum product data unit. (Sbus)
ia<n>, Max I/Os	100 - 4096	700	Maximum number of controller I/Os (Sbus)

[1] The node complex data allocation and node complex data requirement parameters are interrelated. The sum of the data requirements should be less than or equal to the sum of the data allocations. Table 15 shows the relationships between these two parameters.

Table 15 Data allocation and requirements

Allocation specified	Requirement specified	Memory allocated	Memory usage
None	None	16 Mbytes per node	1 Mbyte + 1 Mbyte as needed
Yes	None	As specified	1 Mbyte + 1 Mbyte as needed
None	Yes	16 Mbytes per node	As specified
Yes	Yes	As specified	As specified

Tuning parameters

Server parameters

Table 16 shows the server parameters that are tunable.

Table 16 Server parameters

Parameter	Range	Default	Description
Server, do_udpcksum	0 | 1	1	Enables checksums of user datagram protocol headers by the Network File System

Emulator parameters

Table 17 shows the emulator parameters that are tunable.

Table 17 Emulator parameters

Parameter	Range	Default	Description
Emulator, print_emul_tunables	0 | 1	0	If set, prints emulator specific tunables to boot log
Emulator, adv_WS_option	0 | 1	1	If set, enables advertisement of the transmission control protocol windowsize
Emulator, adv_TS_option	0 | 1	0	If set, enables timestamp advertising
Emulator, disable_loopback_csums	0 | 1	0	Disables checksums on packets sent over the loopback interface
Emulator, do_udpcksum	0 | 1	1	Enables checksums of user datagram protocol packet headers
Emulator, tcp_loopback_mtu	1000 - 64512	16384	Maximum transmit unit for the transmission control protocol over the loopback interface
Emulator, raw_default_socketspace	4096 - 262144	32768	Amount of buffer space set aside for raw socket data
Emulator, sendmsg_access_rights	0 | 1	1	Enables (1) or disables (0) the sendmsg system call to pass access rights
Emulator, str_n_event	0 - 512	25	Number of system bufcall structures to allocate
Emulator, str_n_tevent	0 - 512	128	Number of system timeout structures to allocate
Emulator, str_n_mblk	0 - 8192	1200	Initial number of memory blocks allocated
Emulator, str_lo_pct	0 - 100	60	Percent of memory blocks to be low priority
Emulator, str_med_pct	0 - 100	80	Percent of memory blocks to be medium priority
Emulator, str_msg_sz	16K - 10M	2M	Maximum message size
Emulator, str_ctl_sz	16K - 64K	16K	Maximum I/O control message size
Emulator, inherit_listeners	0 | 1	1	Inherit listening sockets during a fork(2)

Tuning parameters

Route/ARP parameters

Table 18 shows the route/ARP parameters that are tunable.

Table 18 Route/ARP server parameters

Parameter	Range	Default	Description
Routearpserv, print_ras_tunables	0 | 1	0	If set, routearpserver- specific tunables are logged
Routearpserv, subnetsarelocal	0 | 1	1	If set, an address is local even if it belongs to another subnet
Routearpserv, do_rarp	0 | 1	0	If set, the Exemplar Technical Server System acts as a reverse-ARP server
Routearpserv, do_ipf	0 | 1	0	If set, the Exemplar Technical Server System acts as an IP gateway, if it has multiple interfaces
Routearpserv, do_ipsendredirects	0 | 1	1	If set, the Exemplar Technical Server System sends Internet Control Message Protocol route redirect packets to improve route performance

Glossary

A

abortive release
Occurs when a transport user issues a disconnect request to terminate a connection, possibly resulting in loss of data.

active user
A user who initiates a connection-mode service.

address
A unique number or character string that identifies a particular network node. Also called a *network address, host address,* and *internet address.*

address class
An attribute of a DARPA Internet network that indicates the size of the network and limits the network's name space. For example, class C networks are limited to a maximum of 254 hosts.

address resolution
Mechanism for mapping host names to network addresses, or network addresses to physical addresses.

Address Resolution Protocol (ARP)
TCP/IP protocol that maps internet addresses to physical addresses.

American National Standards Institute (ANSI)
A repository and coordinating agency for standards implemented in the U.S. Its activities include the production of Federal Information Processing (FIPS) standards for the Department of Defense (DoD).

ANSI
A repository and coordinating agency for standards implemented in the U.S. Its activities include the production of Federal Information Processing (FIPS) standards for the Department of

Glossary

Defense (DoD).

application layer
Protocol layer that provides system-independent services for end-user applications, such as electronic mail and file transfers.

application-level services
Service that enable application programs to take advantage of network facilities. Also called *end-user services.*

Application Program Interface (API)
A set of system calls and library routines that provide programmers with access to network services.

ARP
TCP/IP protocol that maps internet addresses to physical addresses.

ARPANET
The Advanced Research Project Agency Network, now called the Defense Advanced Research Projects Agency (DARPA) Internet. ARPANET was funded by the U.S. government to serve as a testbed for internetworking technology. TCP/IP protocols were developed as part of this project.

asynchronous mode
Execution mode in which control returns to a user immediately following a library function call even if the corresponding asynchronous event has not occurred.

Asynchronous Transfer Mode (ATM)
Asynchronous Transfer Mode (ATM) is the technique for transport, multiplexing, and switching that provides a high degree of flexibility required by B-ISDN. ATM is a connection-oriented protocol that employs fixed-size packets with a 5-byte header and 48 bytes of information.

B

backbone
A central network used to interconnect LANs.

(BCUG)
Similar to a CUG (Closed User Group), but restricts access between pairs of DTEs.

Berkeley Internet Name Domain (BIND)
A service that enables clients to name objects and services on the network and share those names with other clients. In effect, BIND is a distributed database system for objects in a network.

Berkeley sockets
Interprocess Communication (IPC) facilities provided by a set of system calls and library routines for use in writing applications involving more than one task.

Bilateral Closed User Group (BCUG)
Similar to a CUG (Closed User Group), but restricts access between pairs of DTEs.

BIND
A service that enables clients to name objects and services on the network and share those names with other clients. In effect, BIND is a distributed database system for objects in a network.

bridge
A node used to transparently connect networks, usually of the same type, at the physical layer.

C

called address
Address of the destination host.

calling address
Address of the source host.

Call User Data (CUD)
Network service parameter carried with a call request. This parameter allows data to be sent with the call request.

circuit
A virtual communication path between nodes. Sometimes used to refer to a physical communication path.

client
The user of a service.

client/server model
The structure by which services are implemented. A client process on one host makes a request that a server process on another hosts fulfills.

client stub
Used in implementing remote procedure call (RPC) facilities, the client stub takes the place of the actual called procedure in the client to abstract the details of passing messages over the network.

(CCITT)
An international organization that sponsors standards for data networks, telephone switching, digital systems, and terminals. You often see this organization referred to as the Consultative Committee for International Telephony and Telegraphy, to match the acronym for its French name, Comité Consultatif International de Télégraphique et Téléphonique.

communication line
A physical communication path, such as coaxial or fiber-optic cable.

communication link
A logical communication path consisting of the hardware and software needed to establish a connection and transfer data between nodes. Also called a *logical link.*

computer network
A system of interconnected computers that enables machines and their users to exchange information and share resources.

confirmation
An event that usually results when a peer entity issues a response, such as the acknowledgment of a connection request.

connection establishment
The connection-mode service phase in which two peer processes set up a communication link.

connectionless mode
A transport service mode in which independent units of data are transferred between peer entities.

connectionless service
A service that does not require a continuous communication link between end nodes. Instead, data is sent in independent units addressed to the destination node.

connection-oriented mode
A service mode in which two peer entities establish a communication link prior to exchanging data. Also called *connection mode.*

connection-oriented service
A service that requires two peer entities to establish a communication link prior to exchanging data.

connection release
The connection-mode service phase in which two processes terminate their connection.

Corporation for Open Systems (COS)

COS is an organization composed of major suppliers of data processing and data communications products founded to advance the use of international standards. COS was instrumental in the development of procedures for certifying compliance of communication systems with international standards.

CUD

Network service parameter carried with a call request. This parameter allows data to be sent with the call request.

D

daemon

A process that executes continuously to provide a service on an as-requested basis.

DARPA

The U.S. government agency that funded research on internetworking that led to the development of the TCP/IP protocol suite. See ARPANET.

DARPA Internet

A group of networks interconnected through the use of TCP/IP protocols.

DARPA Internet protocols

TCP/IP protocol suite developed during research on internetworking funded by the U.S. government.

Data Circuit-Terminating Equipment (DCE)

One of two systems that form a connection (the other is DTE). The DCE, or network interface, conveys the data received from one DTE to another.

datagram

In connectionless-mode service, the independent unit of data exchanged between two peer transport users.

datagram socket

IPC facility that provides a bidirectional flow of data that is not promised to be sequenced, reliable, or unduplicated. Applications send and receive data from many different sockets without establishing connections first.

data link layer

Layer of the OSI model responsible for transmitting data over a communication link, including error detection, correction, and recovery functions. Often divided into two sublayers: medium access control (MAC) and logical link control (LLC).

Data Network Identification Code (DNIC)
Part of a DTE address that identifies a specific data network.

Data Terminal Equipment (DTE)
One of two systems that comprise an connection (the other is DCE). The DTE, or host, establishes and terminates a network connection.

data transfer
A phase in both connection-mode and connectionless-mode services in which two peer transport users exchange data.

DCE
One of two systems that form a connection (the other is DTE). The DCE, or network interface, conveys the data received from one DTE to another.

DECnet
A network used to interconnect computers that run DNA protocols.

Defense Advanced Research Projects Agency (DARPA)
The U.S. government agency that funded research on internetworking that led to the development of the TCP/IP protocol suite. See ARPANET.

device path
The OpenBoot term analogous to the SPP-UX term "file system tree," describing a type of hardware device.

device tree
The OpenBoot term analogous to the SPP-UX term "file," describing a hierarchical data structure that OBP creates and maintains.

Destination Service Access Point (DSAP)
The link service access point of the destination link service user.

Digital Network Architecture (DNA)
Digital Equipment Corporation's proprietary network architecture.

DNA
Digital Equipment Corporation's proprietary network architecture.

DNIC
Part of a DTE address that identifies a specific data network.

DNS
A distributed database service that enables clients to name objects and services on the network and share those names with other clients. BIND is an implementation of DNS. See Berkeley Internet Name Domain (BIND).

domain
A portion of name space that can be identified by a label, such as .edu in the host name foobar.edu.

Domain Name System (DNS)
A distributed database service that enables clients to name objects and services on the network and share those names with other clients. BIND is an implementation of DNS. See Berkeley Internet Name Domain (BIND).

dot notation
Defines a network address as a series of four (usually) 8-bit decimal numbers separated by periods, as in 128.50.10.1.

downstream
Refers to the direction from a stream head toward a streams driver. This is the direction of data flow that results from a `write` or `putmsg` operation.

driver
In general, a driver is the software that controls a physical device, such as a network interface. In OSI terminology, a driver is an entity in a streams protocol stack used to multiplex data between protocols. Streams protocol stacks have at least one device driver, which is located at the stream's end and is the interface to the hardware. Drivers are accessed through a node (or nodes) in the file system.

DSAP
The link service access point of the destination link service user.

DTE
One of two systems that comprise an connection (the other is DCE). The DTE, or host, establishes and terminates a network connection.

E

EIA
A national trade association concerned primarily with the development of hardware-level standards. EIA standards are identified by the letters EIA, followed by a hyphen and a number, such as EIA-232, a standard used for connecting terminals to computers.

EIA-232
EIA specification for a widely-used physical interface. It describes the functions for a 25-pin connector.

EIA-422A
EIA specification for a balanced electrical interface.

EIA-449
EIA specification for a physical interface having a greater range and higher data rate than EIA RS-232. It describes the functions for a 37-pin connector.

EIA-423A
EIA specification for a unbalanced electrical interface.

Electronics Industry Association (EIA)
A national trade association concerned primarily with the development of hardware-level standards. EIA standards are identified by the letters EIA, followed by a hyphen and a number, such as EIA-232, a standard used for connecting terminals to computers.

email (electronic mail)
A facility that allows users to communicate information across a network in a way similar to the traditional exchange of memos and correspondence through interoffice mail or the postal system.

end node
Terminals or workstations from which users request network services and hosts that process those requests.

endpoint
The point of communication between a service user and a connection. Each connection has two endpoints.

end system
Terminals or workstations from which users request network services and hosts that process those requests.

end user
The person or program that requests a service.

end-user services
Services that enable people or application programs to take advantage of network facilities.

entity
An addressable unit of software or hardware that provides a service to or makes use of a service provided by another addressable unit.

/etc/biod
Starts NFS asynchronous block I/O daemons.

/etc/hosts
Contains the database of host names and addresses for all hosts on your local network and on networks connected with yours.

/etc/networks
Contains logical names, addresses, and aliases for each network with which your local area network (LAN) communicates.

/etc/nfsd
Starts NFS filesystem requests daemons.

Ethernet
A local area network that interconnects nodes via coaxial cable. It uses the CSMA/CD (Carrier Sense Multiple Access/Collision DeFDDItection) access method and transmits at 10 megabits per second. Ethernet has evolved into the IEEE 802.3 standard.

European Computer Manufacturers' Association (ECMA)
ECMA works closely with ISO and CCITT toward developing standards for data processing and data communication. ECMA includes all European computer manufacturers.

event
In OSI terminology, the transfer of data from a service provider to a service user.

executor
Node at which Network Control Program (NCP) commands are executed. The executor can be the local node or a remote node.

expedited data
Data considered significant and not subject to normal flow control for the connection. Specific handling is defined by individual transport providers.

eXternal Data Representation (XDR)
A facility consisting of a set of library routines that provides a common way of representing data types over a network. XDR allows applications to transfer data between diverse machines, such as Sun Workstations, VAX, IBM-PC, and HP Exemplar Technical servers.

F

Fiber Distributed Data Interface (FDDI)
A network interface that connects HP Exemplar Technical servers directly to FDDI networks. FDDI offers the next level of LAN performance beyond Ethernet, with a peak data rate of 100

megabits per second. The FDDI standard specifies a fiber transmission medium and a token ring topology.

file server
Software that makes local file systems available to remote clients.

File Transfer, Access, and Management (FTAM)
An ISO application-level service that allows users to transfer and manipulate files between hosts.

File Transfer Protocol (FTP)
TCP/IP protocol that allows users to transfer files between hosts.

frame
Data and link-level control information that is transmitted over a network. A packet is carried within the data portion of the frame.

FTAM
An ISO application-level service that allows users to transfer and manipulate files between hosts.

FTP
TCP/IP protocol that allows users to transfer files between hosts.

full device name
The OpenBoot term analogous to the SPP-UX term "absolute path name," describing a series of device paths separated by slashes that represent where a device is located within the hierarchical structure of the device tree.

G

gateway
Device used to interconnect networks with incompatible architectures, protocols, and addressing schemes.

GOSIP
Identifies a set of standard OSI protocols with which networking systems used by U.S. government agencies must conform.

Government OSI Profile (GOSIP)
Identifies a set of standard OSI protocols with which networking systems used by U.S. government agencies must conform.

H

hardware address
The device-dependent physical address of a node attached to a communication line.

High Performance Parallel Interface (HIPPI)
Currently the fastest industry standard for connecting high-performance computers. HIPPI provides a data rate of 800 megabits per second over distances of to 25 meters.

HIPPI
Currently the fastest industry standard for connecting high-performance computers. HIPPI provides a data rate of 800 megabits per second over distances of to 25 meters.

host
A computer system that supports network applications. See node.

I

ICMP
TCP/IP protocol responsible for relaying error messages detected by gateways back to the source node.

IEEE
An international professional organization and a member of ANSI and ISO. IEEE created Project 802, the committee that developed a set of widely-used LAN standards known as the 802 standard.

IEEE 802.2
IEEE LAN standard that specifies the data link layer for the CSMA/CD (Carrier Sense Multiple Access/Collision Detection), token passing bus, and token passing ring access methods.

IEEE 802.3
IEEE LAN standard that specifies the physical layer for the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) access method. See Ethernet.

initiator
In connection-mode service, a transport user that requests a connection to a peer user. See active user, client.

inode
A data structure containing information about a file, such as ownership, permissions, and the file's location on disk. An inode exists for every file accessible to the Hewlett-Packard Exemplar Technical Server Family system.

Institute for Electrical and Electronic Engineers (IEEE)
An international professional organization and a member of ANSI and ISO. IEEE created Project 802, the committee that developed a set of widely-used LAN standards known as the 802 standard.

interface

(1) A logical data path between adjacent layers of the OSI model.
(2) A logical path between any two modules or systems. (See network interface.)
(3) As a verb, interface means to interconnect to or interoperate with.

intermediate node

A networked machine responsible for transferring data along the path between end nodes. See end node.

International Organization for Standardization (ISO)

An international regulatory body for information processing and communication systems. Among other achievements, ISO is responsible for the design of the OSI model. This organization is often referred to as the International Standards Organization.

International Standards Organization

See *International Organization for Standardization (ISO).*

International Telegraph & Telephone Consultative Committee (CCITT)

An international organization that sponsors standards for data networks, telephone switching, digital systems, and terminals. You will often see this organization referred to as the Consultative Committee for International Telephony and Telegraphy, to match the acronym for its French name, Comité Consultatif International de Télégraphique et Téléphonique.

internet

When shown in lowercase, any interconnection of autonomous networks. When written with an initial capital letter, usually refers specifically to the DARPA Internet.

internet address

A 32-bit number that identifies a specific node on a TCP/IP network. Usually written in dot notation, as in 128.40.10.1.

Internet Control Message Protocol (ICMP)

TCP/IP protocol responsible for relaying error messages detected by gateways back to the source node.

Internet Protocol (IP)

TCP/IP protocol that encapsulates packets into datagrams and delivers them from one machine to another. IP also provides addressing and routing services.

Internet Services
HP Exemplar Technical server networking product that provides TCP/IP-based services over Ethernet, FDDI, HIPPI, and ATM interfaces.

internetwork
See internet.

Internetworking
The art and science of interconnecting diverse networks.

interoperability
The ability of network components to communicate.

Interprocess Communication (IPC)
A set of system calls and library routines that gives application programmers access to the full power and functionality of the TCP/IP protocol suite. Through UNIX domain sockets and shared memory, IPC also enables programs to communicate with other programs running on the same machine.

IP
TCP/IP protocol that encapsulates packets into datagrams and delivers them from one machine to another. IP also provides addressing and routing services.

IPC
A set of system calls and library routines that gives application programmers access to the full power and functionality of the TCP/IP protocol suite. Through UNIX domain sockets and shared memory, IPC also enables programs to communicate with other programs running on the same machine.

ISO
An international regulatory body for information processing and communication systems. Among other achievements, ISO is responsible for the design of the OSI model. This organization is often referred to as the International Standards Organization.

ISO reference model
Also known as the *OSI Reference Model* or the *ISO reference model,* the OSI model is an architectural framework for the development of standardized networking systems.

J

Joint Networking Team (JANET)
United Kingdom regulatory body for data communication standards.

K

kernel

The core of the operating system where basic system facilities, such as file access and memory management functions, are performed.

L

LAN

A network that serves several users at relatively high speeds within a small geographic area.

line

A physical communication path between nodes, such as that provided by an Ethernet cable. Also called a *communication line.*

link

A virtual communication path between processes running on different nodes. Also called a *communication link* or *logical link.*

Link Service Access Point (LSAP)

An address or identifier through which a service user accesses data link layer services.

listener

Generally, a user application that manages multiple transport endpoints by waiting on (or listening to) each port for incoming connection requests.

listening endpoint

A transport endpoint that has been bound for listening.

LLC

Data link layer protocols for operation over a LAN.

Local Area Network (LAN)

A network that serves several users at relatively high speeds within a small geographic area.

logical channel number

In packet-switched networks, a number assigned to a virtual connection.

logical link

See link.

Logical Link Control (LLC)

Data link layer protocols for operation over a LAN.

logical unit

Defines a unique path to physical devices (such as disks, tape drives, or networks).

long haul network

A network whose size and service area is larger than a single site. WANs usually include telephone system trunk lines or satellite links. Also called Wide Area Network (WAN).

LSAP

An address or identifier through which a service user accesses data link layer services.

M

MAC

Component of the data link layer that controls access to the physical medium.

MAN

A medium-scale network capable of providing large corporate customers with the ability to transfer massive amounts of data within a service area roughly the size of a city. MANs use LAN technology to provide data rates faster than wide area networks, which rely on regular telephone switching technology.

master server

Contains the master set of NIS maps. When the master server's maps are updated, it distributes the updates to each NIS slave server.

Media Access Control (MAC)

Component of the data link layer that controls access to the physical medium.

message

An arbitrarily long unit of data, such as the contents of a file, that can be transmitted over a network. The size of a message is not limited by the size of buffers used within the protocol stack to transmit the message.

Message Transfer Agent (MTA)

A component of the X.400 Message-Handling System, the MTA is responsible for relaying messages initiated by the user agent (UA).

Metropolitan Area Network (MAN)
A medium-scale network capable of providing large corporate customers with the ability to transfer massive amounts of data within a service area roughly the size of a city. MANs use LAN technology to provide data rates faster than wide area networks, which rely on regular telephone switching technology.

modem eliminator
A device used to connect a local terminal and a computer port without the pair of modems normally used with synchronous links.

module
A STREAMS component that performs intermediate transformations on messages flowing between the Stream head and the driver.

MTA
A component of the X.400 Message-Handling System, the MTA is responsible for relaying messages initiated by the user agent (UA).

multi-homed host
A host attached to multiple network interfaces. Such hosts are assigned one network address per interface. Multi-homed hosts are often used as gateways through which networks are interconnected.

N

named pipe
Pipes that exist permanently in the file system with directory entries and path names. Because you can access these pipes by name, you can use them for a variety of applications that you cannot accomplish with ordinary pipes. Typically, named pipes are used to allow a number of processes to communicate with a daemon.

name server
A system that provides distributed database facilities for hosts in a network. See Berkeley Internet Name Domain (BIND), Domain Name System (DNS), and Network Information Service (NIS).

name space
The set of possible host names within a domain or within an address class.

National Institute of Standards and Technology (NIST)
Formerly known as the National Bureau of Standards (NBS), NIST is responsible for defining the set of standard protocols required for systems used by U.S. government agencies. NIST activities produced the Government OSI Profile (GOSIP).

network
A system of interconnected computers that enables machines and their users to exchange information and share resources.

network architecture
The logical organization of a networking system.

Network File System (NFS)
A system that provides transparent access to networked files over a TCP/IP-based network. NFS links together heterogeneous systems to share resources and files over local area and wide area networks.

Network Information Center (NIC)
The central authority responsible for assigning and maintaining addresses of networks and hosts on the DARPA Internet.

Network Information Service (NIS)
NIS is a distributed network lookup service that eases the job of administering networked machines. By using NIS, password, group, and host information for an entire network may be maintained in a single database.

network interface
A logical path between an application and a physical network. A device driver and network interface board provide the network interface for a Hewlett-Packard Exemplar Technical Server Family host.

network layer
The OSI layer responsible for routing and relaying data from one node to another on the same network or across multiple networks.

Network Service Access Point (NSAP)
An address or identifier through which a Network Service User (NSU) may access network layer services.

Network Service User (NSU)
A software entity that uses the services of the network layer.

network topology
A description of the organization of a network in terms of its components, interconnections, and geography.

NFS

A system that provides transparent access to networked files over a TCP/IP-based network. NFS links together heterogeneous systems to share resources and files over local area and wide area networks.

NIC

The central authority responsible for assigning and maintaining addresses of networks and hosts on the DARPA Internet.

NIS

NIS is a distributed network lookup service that eases the job of administering networked machines. By using NIS, password, group, and host information for an entire network may be maintained in a single database.

NIS client

Requests data from maps on NIS servers.

NIS map

Contains Non-ASCII administrative files, usually derived from ASCII files found in the /etc directory. Each NIS map has a map name used by programs to access it.

NIS server

Fulfills requests made by NIS clients. Also called a *service provider*.

node

A computer attached to a network that initiates or facilitates the flow of data across the network.

NSAP

An address or identifier through which a Network Service User (NSU) may access network layer services.

NSAP address

An address used to identify a specific Network Service Access Point (NSAP) and a particular Network Service User (NSU).

NSU

A software entity that uses the services of the network layer.

O

OpenBoot PROM (OBP)

Firmware (programmable read-only memory) responsible for system control prior to primary operating system execution. OBP first queries; then tests and initializes the system hardware and loads the operating system.

OpenConnect
A gateway product that allows HP Exemplar Technical servers to communicate with nodes in an SNA network.

open systems
Systems that conform to any non-proprietary, publicly-available standards. In recent years, however, the term has come to mean only those systems that use the international standards for network architecture, as specified by the OSI model.

Open Systems Interconnection (OSI)
The ISO definition of a system that provides reliable, data-transparent, host-independent services.

orderly release
A transport service feature in which two cooperating transport users gracefully terminate a connection to avoid data loss.

OSI
The ISO definition of a system that provides reliable, data-transparent, host-independent services.

OSI model
Also known as the *OSI Reference Model* or the ISO reference model, the OSI model is an architectural framework for the development of standardized networking systems.

OSI reference model
See OSI model.

outstanding connect indication
A connect indication that a transport user has received but has not yet responded to.

Glossary

P

packet
Data carried within the frame as the information field. In DARPA Internet terminology, packets are also called *datagrams*. See datagram.

Packet Level Protocol (PLP)
A network layer protocol used for connection-oriented operation.

passive user
In connection-mode service, a transport user that waits for another user to initiate establishment of a connection.

peer entities
Entities running in different nodes at the same layer of the OSI model between which virtual communication takes place. Also called *peer transport users*.

peer processes
Entities running in different nodes at the same layer of the OSI model between which virtual communication takes place. Also called *peer transport users*.

peer protocols
Protocols running at the same OSI layer on different machines.

peer transport users
Entities running in different nodes at the same layer of the OSI model between which virtual communication takes place.

physical address
The device-dependent physical address of a node attached to a communication line.

physical layer
OSI layer responsible for transmitting data bits over a specific physical medium. Physical layer protocols include EIA-232 and V.35.

pipe
A pair of file descriptors that provide the mechanism for a one-way flow of data.

PLP
A network layer protocol used for connection-oriented operation.

port
The logical point through which data flows between a node and a network. A given port is often associated with a particular service.

presentation layer
The OSI layer concerned with the syntax of transmitted information. The presentation layer is responsible for the order and format of data, and for services such as data encryption.

profile
(1) An agreed-upon subset of standards that identifies services members of a group must implement to ensure interoperability with other members' systems. (2) A set of default settings for PAD parameters.

protocol
A set of data and message formats, and a set of procedures governing transmission that when complied with enable network components to interoperate. Protocols define a common way in which network components must transmit and interpret information.

protocol address
Within the context of an entire network, the identifier of a specific transport endpoint.

protocol stack
A layered set of protocols. Entities at each layer provide services to entities at the next higher layer. Also called a *protocol suite.*

protocol suite
See protocol stack.

Public Data Network (PDN)
A network established by a Post, Telephone, and Telegraph (PTT) authority, common carrier, or private operating company for the specific purpose of providing data communication services to the public.

R

raw socket
IPC facility that provides users with access to the underlying network protocols.

Remote Procedure Call (RPC)
A facility provided by the NFS product that allows a client process to have another process execute a procedure call, as if the caller had executed the procedure call in its own address space.

repeater
A node that amplifies the electrical signal on a segment of communication line without interpreting the data, effectively extending the network beyond the limitations of its physical media.

request
An action that initiates the transfer of data between peer entities. For example, an initiator may request that a connection be established. Requests are made by a service user, or *client*, to a service provider, or *server*.

Requests for Comments (RFCs)
A set of notes and technical papers that discuss various subjects related to TCP/IP protocols, including proposed and accepted standards.

responder
A system that responds to a request from an initiator. See server, passive user.

response
An action that an entity may issue to respond to a request. For example, when a peer entity issues a connection request, the responding entity issues a connection response that either accepts or rejects the connection.

RFC
A set of notes and technical papers that discuss various subjects related to TCP/IP protocols, including proposed and accepted standards.

ring
A network topology in which each node is connected to adjacent nodes to complete a circle.

RIP
TCP/IP protocol implemented by the `routed` program to provide routing services for the local network or subnet.

rlogin
Connects your terminal on the current host to a designated remote host.

router
Operates as an intermediate node whose purpose is to direct messages to the appropriate network. Routers use the destination address included in a packet to determine where to send it. If more than one route to the destination exists, the router tries to choose the most efficient one.

routing
The task of finding the most efficient path over which to send packets to their destination.

Routing Information Protocol (RIP)
TCP/IP protocol implemented by the `routed` program to provide routing services for the local network or subnet.

RPC
A facility provided by the NFS product that allows a client process to have another process execute a procedure call, as if the caller had executed the procedure call in its own address space.

S

SAP
An address or identifier through which a service user can communicate with a service provider.

server
A process that fulfills a request issued by a client process, and transmits a response back to the client. Also called a *service provider.*

server stub
Used in implementing remote procedure call (RPC) facilities, the server stub waits for an RPC request from a client, executes a local procedure call on behalf of the client, and returns results to the client. Server stubs are used to abstract the details of passing messages over the network.

service
A function or set of functions provided by a network *entity.*

Service Access Point (SAP)
An address or identifier through which a service user can communicate with a service provider.

service provider
An entity that provides service for the next higher entity on a protocol stack. For example, the physical layer is a service provider to the data link layer. Also called a *server.*

service user
An entity that uses the service of the next lower entity in a protocol stack. For example, a transport entity is the service user of a network entity. Also called a *client.*

session layer
The OSI layer that establishes, manages, and terminates a period of communication between two end users. It is also responsible for synchronizing the exchange of data and controlling traffic over the connection.

Simple Mail Transfer Protocol (SMTP)
TCP/IP protocol used to relay electronic mail messages between networked machines.

slave server
Contains complete copies of the master server's set of NIS maps.

SMTP
TCP/IP protocol used to relay electronic mail messages between networked machines.

SNA
Proprietary network architecture developed by IBM to connect IBM computers. Hewlett-Packard Exemplar Technical Server Family computers can be interconnected with SNA networks through the gateway product, *OpenConnect*.

SNPA
An address that identifies the physical attachment of a system to a network. For a WAN, the SNPA is the DTE address of the system; for an Ethernet LAN, the SNPA is the combined Ethernet address and Link SAP.

socket
Endpoint used for interprocess communication. See Berkeley sockets.

socket pair
Bidirectional pipes that enable application programs to set up two-way communication between processes that share a common ancestor.

Source Service Access Point (SSAP)
Link service access point of the source link service user.

SSAP
Link service access point of the source link service user.

star
A network topology in which a central node serves as the point of connection for all other nodes.

Stream
Data flow path between a Stream head and driver in a STREAMS protocol stack.

Stream head
The entity in a STREAMS protocol stack that interfaces with a user process, providing the interface between the Stream in kernel space and the user application in user space. The Stream head processes STREAMS system calls from the user application and allows data to be transferred between the user application and the Stream in both directions.

STREAMS
A combination of system calls, kernel routines, and kernel utilities that provide an efficient means of implementing a dynamic network stack. STREAMS defines a standard interface between a STREAMS-based user application and the STREAMS protocol stack with which it communicates.

stream socket
IPC facility that provides for the bidirectional, reliable, sequenced, and unduplicated flow of data without record boundaries.

subnet
A network accessed via a single physical link or a single Ethernet interface.

subnetting
Partitions network address space defined by a single network number into multiple virtual networks called *subnets* or *subnetworks*.

subnetwork
A network accessed via a single physical link or a single Ethernet interface.

Subnetwork Point of Attachment (SNPA)
An address that identifies the physical attachment of a system to a network. For a WAN, the SNPA is the DTE address of the system; for an Ethernet LAN, the SNPA is the combined Ethernet address and Link SAP.

synchronous mode
Execution mode in which a user calls a library function and control does not return until after the corresponding asynchronous event occurs.

System Network Architecture (SNA)
Proprietary network architecture developed by IBM to connect IBM computers. Hewlett-Packard Exemplar Technical Server Family computers can be interconnected with SNA networks through the gateway product, *OpenConnect*.

T

TCP
A connection-oriented TCP/IP protocol used to transfer a stream of data from a process running in one machine to its peer process running in a remote machine.

TCP/IP
A layered set of internetworking protocols named after its two major protocols, Transmission Control Protocol (TCP) and Internet Protocol (IP).

TCP/IP protocols
TCP/IP protocol suite developed during research on internetworking funded by the U.S. government.

TELNET Protocol
TCP/IP protocol that enables a user to log in to a remote host that runs basic TCP/IP protocols, but not necessarily BSD networking protocols.

throughput class
QOS parameter that selects throughput rate in bits per second. Throughput describes the maximum amount of data that can be sent through the network when the network is operating at saturation.

topology
A description of the organization of a network in terms of its components, interconnections, and geography.

TP0/2/4
Transport Protocol (TP), Classes 0, 2, and 4. Each class provides a different set of transport layer services, such as packet sequencing and retransmission.

Transmission Control Protocol (TCP)
A connection-oriented TCP/IP protocol used to transfer a stream of data from a process running in one machine to its peer process running in a remote machine.

Transport address
An address consisting of a TSAP selector and a network address that uniquely identifies a TSU.

transport connection
In connection-mode service, the communication circuit between two peer transport users.

transport endpoint
Within the context of a single system, the path of communication between a transport user and an underlying transport provider. The path is identified by a local file descriptor.

transport layer
The OSI layer that provides a reliable end-to-end or host-to-host data transfer service that shields upper layers from the details of the underlying network. It is responsible for ordered delivery of data, flow control, and error recovery.

Transport Layer Interface (TLI)
The collection of operations, states, and events that define the interface between transport users and transport providers.

transport provider
A protocol that provides transport-level services.

transport provider identifier
When initializing a transport endpoint, the character string that identifies the device file that corresponds to a transport provider.

Transport Provider Interface (TPI)
Interface used by application programs to access transport services, such as TCP, UDP, and TP0/2/4. Transport drivers communicate with application programs through a stream head.

Transport Service Access Point (TSAP)
On a specific end system, uniquely identifies the context of a transport provider for a transport user. In connection-mode service, a single TSAP may be connected to multiple remote TSAPs with each connection uniquely identified by the peer transport endpoints.

Transport Service User (TSU)
An OSI software entity that uses the services of one or more transport providers.

transport user
An OSI software entity that uses the services of one or more transport providers.

Trivial File Transfer Protocol (TFTP)
TCP/IP protocol that enables users to transfer files to and from remote hosts. It is normally used to transfer files, such as boot files, from the HP Exemplar Technical servers to remote workstations. Unlike File Transfer Protocol (FTP), TFTP does not require a user account or password on the remote host.

TSAP
On a specific end system, uniquely identifies the context of a transport provider for a transport user. In connection-mode service, a single TSAP may be connected to multiple remote TSAPs with each connection uniquely identified by the peer transport endpoints.

TSAP selector
The name of a TSAP relative to an NSAP. When combined with a network address, it provides a globally unique name for a transport address.

U

UA
Entity that provides the interface between an end user and a Message Transfer Agent (MTA) in an X.400 Message-Handling System (MHS).

UDP
Connectionless TCP/IP protocol used to transfer datagrams from a process running in one machine to its peer process running in a remote machine.

UNIX-to-UNIX Copy (UUCP)
A communication system that can run over direct serial lines, network connections, or ordinary telephone lines. UUCP is used for file copying and remote command execution.

unreliable data delivery service
Connectionless service that does not guarantee delivery of a packet to its destination. Other entities must perform error detection and correction functions.

upstream
Refers to the direction from a STREAMS driver to the Stream head. This is the direction of data flow that results from a `read` or `getmsg` operation.

User Agent (UA)
Entity that provides the interface between an end user and a Message Transfer Agent (MTA) in an X.400 Message-Handling System (MHS).

user data
Data that originates from a service user.

User Datagram Protocol (UDP)
Connectionless TCP/IP protocol used to transfer datagrams from a process running in one machine to its peer process running in a remote machine.

UUCP
A communication system that can run over direct serial lines, network connections, or ordinary telephone lines. UUCP is used for file copying and remote command execution.

W

Wide Area Network (WAN)
A network whose size and service area is larger than a single site. WANs usually include telephone system trunk lines or satellite links. Also called a *long haul network*.

Index

O

R

S

T

HEWLETT® PACKARD

CONVEX PRESS

B5655-90029