ULTRIX

Introduction to Networking and Distributed System Services

Order Number: AA–ME97B–TE

ULTRIX

Introduction to Networking and Distributed System Services

Order Number: AA-ME97B-TE
June 1990

Product Version: ULTRIX Version 4.0 or higher

This guide provides an overview of basic TCP/IP networking concepts and the ULTRIX software's distributed system services. It also briefly discusses the Simple Network Management Protocol (SNMP) and the ULTRIX implementation of the SNMP Agent.

digital equipment corporation
maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital
CDA
DDIF
DDIS
DEC
DECnet
DECstation
DECUS
DECwindows
DTIF
MASSBUS
MicroVAX
Q-bus
ULTRIX
ULTRIX Mail Connection
ULTRIX Worksystem Software
VAX
VAXstation
VMS
VMS/ULTRIX Connection
VT
XUI

Ethernet is a registered trademark of Xerox Corporation.

UNIX is a registered trademark of AT&T in the USA and other countries.

Contents

About This Manual

1 Networking Concepts

2 Distributed System Services

3 Managing a Network with SNMP

A Local Area Network Files

Figures

Tables

About This Manual

This guide discusses the following topics:

- Internet addressing, network classes, and routing on TCP/IP local area networks (LANs)
- Distributed system services
- The Simple Network Management Protocol (SNMP)

The ULTRIX software includes support for the TCP/IP suite of networking protocols. Whenever LANs or networking services are discussed, a TCP/IP LAN is assumed.

Audience

This guide is intended for the person who maintains networks on a system running the ULTRIX operating system. This person is often the system administrator of a timesharing system, but could be the system adminstrator of a workstation in a networked environment.

To use this guide, you should be familiar with ULTRIX system commands, the system configuration, file naming conventions, and an editor, such as `vi` or `ed`. You should also know the names and addresses of the other systems on the network if a LAN already exists.

Organization

This manual has three chapters, three appendixes, and an index:

Chapter 1: Networking Concepts
This chapter describes the Internet addressing scheme, network classes, and network routing.

Chapter 2: Distributed System Services
This chapter describes the distributed system services that are supported by the ULTRIX software, and discusses how to plan a distributed environment. It also discusses the relationship between the distributed system services that you configure on your network and your network's security level.

Chapter 3: Managing a Network with SNMP
This chapter provides an overview of the Simple Network Management Protocol (SNMP), and describes the ULTRIX implementation of the SNMP Agent and Extended Agent.

Appendix A: Local Area Network Files
This appendix describes the system files involved in setting up and maintaining a LAN.

Appendix B: The Management Information Base
This appendix describes the structure of the Management Information Base

(MIB) that is used by the SNMP Agent.

Appendix C: Network Maintenance Commands
This appendix describes commands that are useful for monitoring and maintaining your network.

Related Documents

The following RFCs are referenced in the text and are available from the Network Information Center (NIC):

DDN Network Information Center
SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025
(800) 235-3155 or (415) 859-3695

E-mail: nic@ddn.mil

- RFC 1066—*Management Information Base for Network Management of TCP/IP-based internets*
- RFC 1098—*A Simple Network Management Protocol*
- RFC 1118—*The Hitchhiker's Guide to the Internet*
- RFC 1129—*Internet time synchronization: the Network Time Protocol*

The following text book is also a valuable reference: Internetworking with TCP/IP: Principals, Protocols and Architecture, by Douglas Comer, Prentice-Hall, Inc., 1988. (ISBN 0-13-470154-2)

You should have the hardware documentation for your system available, as well as the following ULTRIX manuals:

- *Guide to the BIND/Hesiod Service*
- *Guide to the Yellow Pages Service*
- *Guide to Kerberos*
- *Security Guide for Administrators*

Conventions

% The default user prompt is your system name followed by a right angle bracket. In this manual, a percent sign (%) is used to represent this prompt.

A number sign is the default superuser prompt.

user input This bold typeface is used in interactive examples to indicate typed user input.

`system output` This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.

UPPERCASE
lowercase

The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.

rlogin

In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.

macro

In text, bold type is used to introduce new terms.

[ ]

In syntax descriptions and function definitions, brackets indicate items that are optional.

cat(1)

Cross-references to the *ULTRIX Reference Pages* include the appropriate section number in parentheses. For example, a reference to cat(1) indicates that you can find the material on the cat command in Section 1 of the reference pages.

CTRL/*x*

This symbol is used in examples to indicate that you must hold down the CTRL key while pressing the key *x* that follows the slash. When you use this key combination, the system sometimes echoes the resulting character, using a circumflex (^) to represent the CTRL key (for example, ^C for CTRL/C). Sometimes the sequence is not echoed.

New and Changed Information

This manual is a revision. New and changed information is as follows:

- The name of the manual has changed to reflect its contents more accurately. The new name is *Introduction to Networking and Distributed System Services.*
- All network setup information has been removed from this manual. For information on using the netsetup command to establish a node on a LAN, see the *Guide to System and Network Setup*. For information on using the netsetup command to update the /etc/hosts and /etc/hosts.equiv files, see the netsetup(8) reference page.
- For information on setting up routers (commonly called gateways) or subnetworks, see the *Guide to System and Network Setup.*
- Chapter 2, Distributed System Services—This new chapter describes the network naming, time, and authentication services supported by the ULTRIX software. It also describes how to plan a distributed environment.
- Chapter 3, Managing a Network with SNMP—This new chapter provides an overview of the Simple Network Management Protocol (SNMP). It also describes the ULTRIX implementation of the SNMP Agent.
- Appendix A: Local Area Network Files—The material in this appendix was previously located in Chapter 1 of the *Guide to Networking.* It describes the system files involved in setting up and maintaining a LAN.
- Appendix B: The Management Information Base—This new appendix includes several figures that depict the structure of the Management Information Base (MIB) used by the ULTRIX SNMP Agent.

- Appendix C: Network Maintenance Commands—This new appendix describes commands that are useful for monitoring and maintaining your network.

In addition, there are artistic, editorial, formatting, and typographical changes.

Networking Concepts 1

This chapter provides an overview of the following concepts:

- Internet addresses
- Routing decisions
- Subnetworks and Netmasks
- Internet Protocol broadcast addresses

In addition, Section 1.6 contains examples and explanations that integrate information on Internet addresses, subnetworks, and Internet broadcast addresses.

1.1 Overview

A **local area network** (LAN) is a group of two or more computer systems, or hosts, connected by a transmission medium, such as an Ethernet cable. Each host is connected to the transmission medium by a hardware interface.

Every LAN should be assigned a unique network number by the Network Information Center (NIC). The local network administrator then assigns each host connected to the LAN an Internet address that includes both the LAN's network number and a unique host number.

Note

To obtain an Internet address for your network, contact:

DDN Network Information Center
SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025
(800) 235-3155 or (415) 859-3695

E-mail: nic@nic.ddn.mil

Digital recommends that you register your LAN with the NIC even if you do not intend to connect to the Internet. Then, if you decide to connect to the Internet later, you will not have to change all the host addresses on your LAN.

An Internet network consists of two or more LANs interconnected by a host that acts as a **router** (commonly called a gateway).

Routers have a network interface for each LAN to which they are connected. Each network interface is assigned a unique Internet address. Because they are connected to multiple LANs, routers allow data to be transferred between systems on the LANs to which they are connected.

Table 1-1 shows some of the common interface names, the controllers that they correspond to, and the processors for which they provide a network interface.

Note that the ln, sl, and xna interfaces are used on both VAX and RISC processors.

Table 1-1: Common Interface Names

Name	Controller	Machine
Ethernet		
de	DEUNA or DELUA	VAX
ln	DESVA	VAXstation, or DECstation
ni	DEBNA	VAX
qe	DEQNA or DELQA	MicroVAX
xna	DEBNI or DEMNA	VAXstation, or DECstation
Point-to-Point		
dmc	DMR-11 or DMC-11	VAX
dmv	Q-bus	MicroVAX
sl	Serial line IP	VAXstation, or DECstation

1.2 Internet Addresses

An Internet address has a total of 32 bits (four bytes), and is composed of the network number and the host number. The network number is assigned by the NIC; the host number is assigned locally. From the Internet address of a particular host, you can determine the network's class and number, as well as the individual host's number.

When you apply to the NIC for an Internet address, it assigns your network a unique number based on the information you provide. Your entire network is known to the Internet by that address, regardless of how the hosts are configured locally.

There are two ways of notating the Internet address:

- The common notation
- The alternate notation

1.2.1 Common Internet Address Notation

The common Internet address notation uses four fields separated by dots (.) to describe the 32 bits:

fieldl.field2.field3.field4

Each field ranges from 0 to 255 decimal. The significance of each field and the bits within each field depend on the network class.

The network number always includes *field1*. The first (high-order bits) bit or bits in *field1* indicate the network class of the Internet address. Depending on the network class, the last one, two, or three fields contain the host number.

The following is an example of a network number expressed in common notation:

```
98.15.12.62
```

1.2.2 Alternate Internet Address Notation

In the alternate Internet address notation, the network part is notated in the same way as the common Internet address notation. However, the alternate Internet notation expresses the entire host part of the Internet address as one field.

An example of an Internet address on a class A network expressed in the common notation is 98.15.12.62. The same Internet address expressed in the alternate notation is 98.986174.

Use base 256 arithmetic to convert the host portion of the address (15.12.62) to the alternate notation. Do this by multiplying the host fields of the address by increasing powers of 256. Multiply field 4 by 256**0, which is the same as multiplying field 4 by 1. Multiply field 3 by 256**1, which is the same as multiplying field 3 by 256. Multiply field 2 by 256**2, which is the same as multiplying field 2 by 65536. Add the resulting numbers together to get the correct number for the host portion of the address.

The arithmetic for calculating the host number 15.12.62 is as follows:

```
62 x 256**0 =        62
12 x 256**1 =      3072
15 x 256**2 =    983040
--------------------------
                 986174
```

See the `inet`(3n) reference page for a detailed discussion of the Internet address.

1.2.3 Network Classes

The NIC categorizes networks into classes based on the total number of hosts attached to a network. The three main classes are A, B, and C. The network numbers for each class fall within a specific range.

Table 1-2 shows the binary and decimal ranges of the network numbers for the three network classes.

Table 1-2: Binary and Decimal Ranges for the Three Network Classes

Class	Binary Range	Decimal Range
A	00000000–01111111	0–126
B	10000000–10111111	128–191.254
C	11000000–11011111	192–223.254.254

Each of the four bytes in the Internet address is interpreted slightly differently depending on the network class.

Class A networks are the largest, with more than 65,536 hosts. In a Class A network, the first field of the Internet address specifies the network number and class. The first field can be from 1 to 126. The high-order bit of the network number (as expressed in binary) is always 0. By convention, 127 is reserved as the local loopback address, which is defined in the `/etc/networks` and `/etc/hosts` files. Loopback is used for testing and for local connections (requests that are handled locally).

The remaining three fields specify the host number (and a subnet address, if subnetworks are being used; see Section 1.4).

Very few networks are large enough to meet the criteria for a Class A Internet address. Most are either Class B or Class C.

Class B networks are mid-size networks, with from 256 to 65,536 hosts. In a Class B network, the first two fields of the Internet address specify network number and class. The first field can be from 128 to 191, and the second field can be from 1 to 254. The first two high-order bits of the network number (as expressed in binary) are always 10.

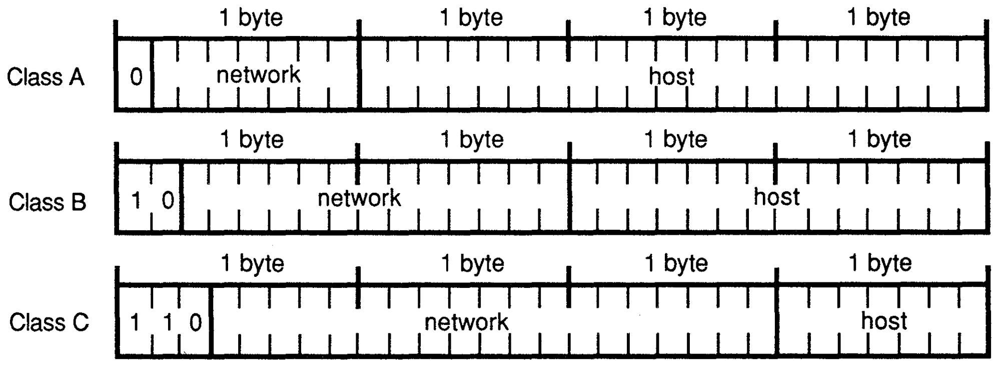
The remaining two fields specify the host number (and a subnet address, if subnetworks are being used).

Class C networks are the smallest networks, with up to 256 hosts. The first three fields of the Internet address specify the network number and class. The first field can be from 192 to 223, the second field from 0 to 255, and the third field from 1 to 254. The first three high-order bits of the network number (as expressed in binary) are always 110.

The remaining field specifies the host number. Subnet routing is not generally used with a Class C network because the eight bits in the host field only allow for 255 hosts on the network. If subnet routing were used, there would be even fewer host bits available.

Figure 1-1 shows the bit positions and decimal ranges of the network numbers for the three main network classes. Note that for a Class A network the first high-order bit is 0; for a Class B network the first two high-order bits are 10; and for a Class C network the first three high-order bits are 110.

Figure 1-1: Internet Network Classes



ZK-0089U-R

1.3 Routing Decisions

Two kinds of routing occur: direct and indirect. All routing is based on the network number of the Internet address and the use of Internet Protocol (IP) routing tables.

All hosts, including routers, use routing tables to route and deliver data. For each destination network, routing tables list the network number and the Internet address of the router to use when sending data to that network. If the destination network is connected to the host, the host sends the information directly. If the destination network is not directly attached to the host, the host consults its routing table to find the internet address of the router to use to forward the data.

1.3.1 Direct Routing

Direct routing occurs between hosts on the same physical network. The host checks the network number on the packet it is sending. If the network number on the packet matches a network number to which the host is directly connected, it sends the packet over the appropriate connection to the destination host.

If the network number on the packet differs from that of the sending host, the host uses indirect routing to send it.

1.3.2 Indirect Routing

When a packet is destined for a host on another network, the sending host sends the packet to a router (commonly called a gateway, or IP gateway) on its own physical network. The router does the following when it receives a packet destined for another network:

1. Checks whether there is a host-specific route specified for the destination host.

   If there is a host-specific route the router uses that route to send the packet. If there is no host-specific route, it continues with step 2.

2. Checks whether there is a network-specific route specified for the destination network.

   If there is a network-specific route the router uses that route to send the packet. If there is no network-specific route, it continues with step 3.

3. Checks whether there is a default route specified.

   The default route indicates the router that should receive data for which no route is explicitly specified. If there is a default route specified, the router forwards the packet. If there is no default route, it returns the following error message:

   ```
   Network unreachable
   ```

1.3.3 Adding Routes

You can update your host's routing tables automatically with the `/etc/routed` daemon, or manually with the `/etc/route` command.

1.3.3.1 The /etc/routed Daemon

The /etc/routed Daemon – The routed daemon updates your host's internal routing tables automatically. If your host is running routed any directly connected hosts or networks that are also running routed periodically supply your host with updates from their routing tables. If any new routes have been added, or if any old routes are deleted, your host receives that information.

To run the routed daemon, remove the comment characters (#) from in front of the following lines in the /etc/rc.local file:

```
#[ -f /etc/routed ] && {
#       /etc/routed & echo 'routed'          >/dev/console
#}
```

The routed daemon is invoked from the /etc/rc.local file when the system reboots. To start the daemon immediately without rebooting, issue the following command at the superuser prompt:

```
# /etc/routed
```

See the routed(8c) reference page for more information on the routed daemon.

1.3.3.2 The /etc/route Command

The /etc/route Command – You can use the /etc/route command to add routes to your host's routing tables manually. You need to specify default routes and routes to particular hosts or networks manually.

You can specify one of three kinds of routes:

- Host-specific
- Network-specific
- Default

A host-specific route explicitly states the route that data destined for a certain host should take. The following command routes data destined for the host host2 through the router host1:

```
# /etc/route add host2 host1 1
```

A network-specific route explicitly states the route that data destined for a certain network should take. The following command routes data destined for the network 16.1.10 through the router host1:

```
# /etc/route add net 16.1.10 host1 1
```

The net indicates that the route is to a network, and prevents the network number that you specify from being mistakenly interpreted as a host.

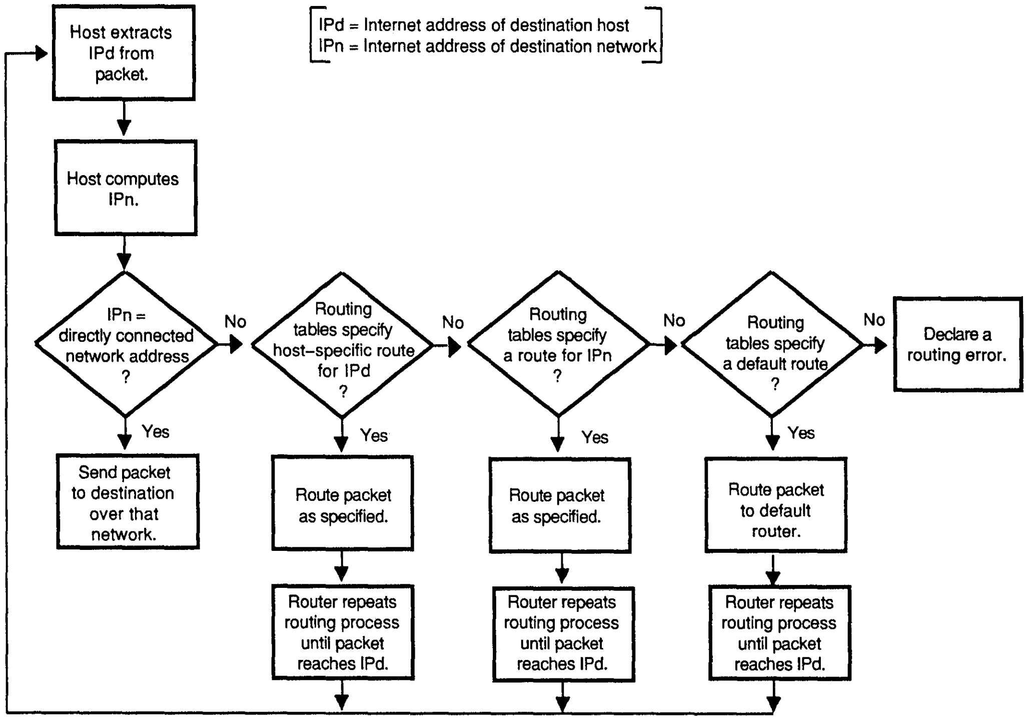
A default route is one that the host uses to route data to networks not listed in its routing table. The following command specifies that all default data be routed through host1:

```
# /etc/route add default host1 1
```

See the route(8c) reference page for more information on the /etc/route command.

The flow diagram in Figure 1-2 illustrates the process a host goes through when deciding how to route data to another host.

Figure 1-2: Internet Routing Process


IPd = Internet address of destination host
IPn = Internet address of destination network
Host extracts IPd from packet.
Host computes IPn.
IPn = directly connected network address ?
No
Routing tables specify host-specific route for IPd ?
No
Routing tables specify a route for IPn ?
No
Routing tables specify a default route ?
No
Declare a routing error.
Yes
Send packet to destination over that network.
Yes
Route packet as specified.
Router repeats routing process until packet reaches IPd.
Yes
Route packet as specified.
Router repeats routing process until packet reaches IPd.
Yes
Route packet to default router.
Router repeats routing process until packet reaches IPd.
ZK-0174U-R

See the *Guide to System and Network Setup* for information on setting up a router.

1.4 Subnetworks and Netmasks

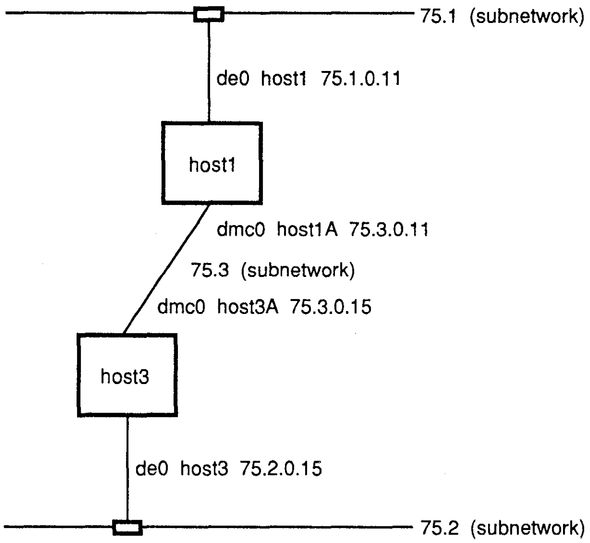
Subnetworks allow a network to be known by one address to the Internet, while being known locally by a set of addresses. They allow you to organize hosts on your network into logical groups, while simplifying and distributing administration and communications on the Internet.

The NIC specifies only the network portion of the Internet address. How the bits in the host portion of the Internet address are used is at the discretion of each site. You can tell the systems on the LAN how many bits from the host portion of the Internet address to use to define subnetworks by defining a network mask, or **netmask.**

If your network uses subnetworks you interprete the Internet address slightly differently. The bits in the host field of the Internet address are divided into two groups: subnetwork and host. Therefore, in networks that use subnet routing, the Internet address consists of three fields: network, subnetwork, and host.

Figure 1-3 shows three subnetworks of network 75, connected by point-to-point links. The systems `host1` and `host3` each have two network interfaces: `host1` is connected to subnetwork 75.1 by a DEUNA or DELUA interface (de0) whose Internet address is 75.1.0.11, and to subnetwork 75.3 by a DMR–11 or DMC–11 interface (dmc0) whose Internet address is 75.3.0.11. The system `host3` is connected to subnetwork 75.2 by a DEUNA or DELUA interface (de0) whose Internet address is 75.2.0.15, and to subnetwork 75.3 by a DMR–11 or DMC–11 interface (dmc0) whose Internet address is 75.3.0.15.

Figure 1-3: Subnetwork Configuration with Point-to-Point Links



ZK-0158U-R

A host on another network does not know that a particular network uses subnetworks. It accesses all of the hosts on that network by going through the router that connects the two networks. From there, the data is routed to the appropriate subnetwork, where the destination host recognizes its own address and picks up its data.

The netmask, which is defined in the `/etc/rc.local` file of each system, tells the system which bits of the Internet address to interpret as the network number, subnetwork number, and host number. The netmask entry is similar to the following:

```
/etc/ifconfig qe0 '/bin/hostname' broadcast 133.185.7.255 \
                                  netmask 255.255.252.0
```

Like the Internet address, a netmask is a 32-bit number with the following format:

field1.field2.field3.field4

There is a one-to-one correspondence between the 32 bits in the netmask and the 32 bits in the Internet address.

When a bit in the netmask is turned on (binary 1), the corresponding bit position in the Internet address is interpreted as part of the network or subnetwork address. When a bit in the netmask is turned off (binary 0), the corresponding bit position in the Internet address is interpreted as part of the host address. For example, the decimal number 255 is 11111111 in binary notation, and means the field should be interpreted as part of the network or subnetwork address.

The first field of the netmask is always 255 because it is interpreted as the network address, regardless of whether there are subnetworks.

The fourth field of the netmask is usually 0, so the system can interpret the host address. The second and third fields of the netmask are usually 255 or 0. How fields 2 and 3 of the netmask are interpreted depends on the class of network.

For example, a netmask of 255.255.0.0 could be one of the following:

- The default netmask for a Class B network with *field1* and *field2* defining the network number, and *field3* and *field4* defining the host number
- A Class A network with *field1* defining the network number, *field2* defining the subnetwork number, and *field3* and *field4* defining the host number

Remember, only bits in the host portion of the Internet address are used to designate subnetworks. Therefore, in a Class A network you can use bits in *field2*, *field3*, or *field4* to designate subnetworks, and bits in *field3* or *field4* of a Class B network. Class C networks rarely use subnetworks, but if they did, only bits in *field4* would be available.

In general, an entire 8-bit field is either turned on (255) or off (0). This makes it easier for users to distinguish between the network, subnetwork, and host portions of the address. However, values other than 255 and 0 can be used also. Following are the valid values in decimal and binary notation for the two middle fields of the netmask:

Decimal	Binary
255	11111111
254	11111110
252	11111100
248	11111000
240	11110000
224	11100000
192	11000000
128	10000000
0	00000000

1.5 Internet Protocol Broadcast Addresses

The broadcast mask, which is defined in the `/etc/rc.local` file, interprets an Internet address as a broadcast address. The Internet Protocol broadcast address allows messages to be sent to all the hosts on the network at the same time. Therefore, all hosts on a network must have the same broadcast address.

The default format of an IP broadcast address consists of the network number followed by all 1s.

For example, a broadcast address for a host on a Class B network could be:

```
129.39.255.255
```

The Class B network number is 129.39. Fields 3 and 4, which specify the host number in a Class B network, are set to 255. The decimal number 255 is 11111111 in binary.

Note

The industry standard default broadcast address is all 1s, and the ULTRIX operating system follows this standard. However, some operating systems, such as 4.2BSD and the ULTRIX operating system prior to Version 1.2, used all zeros for their broadcast address.

While operating systems that use all 1s can broadcast to and receive from systems that broadcast using all zeros, systems that broadcast using all zeros cannot broadcast to and receive from systems using all 1s. Ideally, you should segregate older machines using zeros on a separate network, and place a router from this network to the networks where 1s are being used. However, if your LAN must include systems that use all zeros as their broadcast address, and you want to preserve backward compatibility, you must change the IP broadcast address on all machines to all zeros.

See the `ifconfig`(8) reference page for more information.

1.6 Examples

This section contains examples that demonstrate the relationship between Internet addresses, subnetworks, and Internet broadcast addresses.

1.6.1 Class A Network Examples

The following examples describe Class A networks.

Class A Network Without Subnetworks – Assume you have an Internet address of 73.16.0.56 and a netmask of 255.0.0.0. In binary notation 73 is 01001001. You know that the address is on a Class A network because 73 is in the range 1 through 126 (inclusive). You can verify that this is a class A network because the first high-order bit is a zero.

In this example, the first field of the Internet address (73) is interpreted as the network part. Fields 2, 3 and 4 of a Class A network address represent portions of the host address. The netmask 255.0.0.0 tells us that subnets are not being used on this network. All of the bits in fields 2, 3 and 4 designate host numbers.

Class A Network With Subnetworks – Assume you have an Internet address of 98.0.0.65. and a netmask of 255.255.255.0. In binary notation 98 is 01100010. You know that the address is on a Class A network because 98 is in the range 1 through 126 (inclusive). You can verify that this is a class A network because the first high-order bit is a zero.

In this example, the first field of the Internet address (98) is interpreted as the network part. The netmask 255.255.255.0 tells us that subnetworks are being used on this network. 16 bits (fields 2 and 3) are being used to designate subnetworks, and only the bits in field 4 designate the host number.

1.6.2 Class B Network Examples

The following examples describe Class B networks.

Class B Network Without Subnetworks – Assume you have an Internet address of 128.0.2.42 and a netmask of 255.255.0.0. In binary notation 128 is 10000000. You know that the address is on a Class B network because 128 is in the range 128 through 191 (inclusive). You can verify that this is a class B network because the first two high-order bits are 10.

In this example, the first two fields of the Internet address (128.0) are interpreted as the network part. Fields 3 and 4 of a Class B network address represent portions of the host address. The netmask 255.255.0.0 tells us that subnetworks are not being used on this network. All of the bits in fields 3 and 4 designate host numbers.

Class B Network With Subnetworks – Assume you have an Internet address of 136.121.18.177 and a subnetmask of 255.255.255.0. In binary notation 136 is 10001000. You know that the address is on a Class B network because 136 is in the range 128 through 191 (inclusive). You can verify that this is a class B network because the first two high-order bits are 10.

In this example, the first two fields of the Internet address (136.121) are interpreted as the network part. The netmask 255.255.255.0 tells us that subnetworks are being used on this network. Eight bits (field 3) are being used to designate subnetworks, and only the bits in field 4 are designating the host number.

Table 1-3 lists more examples of Internet addresses, netmasks, and corresponding broadcast addresses.

Table 1-3: Internet Addresses, Netmasks, and Broadcast Addresses

Internet Address	Class	Network Number	Host Number	Subnet	Broadcast Address	Netmask
3.0.0.10	A	3.	10	No	3.255.255.255 or 3.0.0.0	255.0.0.0
11.1.0.12	A	11.1	12	Yes, 8 bits	11.1.255.255 or 11.1.0.0	255.255.0.0
129.39.0.15	B	129.39	15	No	129.39.255.255 or 129.39.0.0	255.255.0.0
128.45.4.8	B	128.45	8	Yes, 6 bits	128.45.7.255 or 128.45.7.0	255.255.252.0
192.0.1.8	C	192.0.1	8	No	192.0.1.255 or 192.0.1.0	255.255.255.0

Distributed System Services 2

This chapter describes the distributed system services that are available to coordinate the distribution of information throughout your network, synchronize network time, and enhance your network's security.

The ULTRIX software provides the following services for configuring a distributed environment:

- Naming services
- Time services
- Authentication services

It also offers the following configurable security levels: BSD (the default), UPGRADE, and ENHANCED. There are dependencies, described in Section 2.4, between a network's security level and the distributed system services it is running.

2.1 Naming Services

The ULTRIX software supports the following naming services:

- The Berkeley Internet Name Domain (BIND) service, which supports the Hesiod name service
- The Yellow Pages (YP) service

The library routines in `/usr/lib/libc.a` allow transparent access to BIND/Hesiod, YP, and local `/etc` files. The name services configuration file, `/etc/svc.conf`, dictates which naming services are queried, and in what order, for a particular database.

The ULTRIX software allows you to convert from a YP distributed environment to a BIND/Hesiod distributed environment, or to run both services in the same environment. Because the source files for both BIND/Hesiod and YP can be `/etc`-style files, a distributed BSD source area can be shared between the two services by means of symbolic links.

2.1.1 BIND/Hesiod Service

Using the BIND/Hesiod service you can distribute the following databases:

```
aliases
auth
group
hosts
networks
passwd
protocols
rpc
services
```

If the hosts database is located in /var/dss/namedb/src, the bindsetup command automatically appends the bindmaster alias to the hosts database entry for the BIND/Hesiod primary server. This entry allows the BIND/Hesiod primary server to run the Hesiod password update daemon, /usr/etc/hesupd. The passwd command communicates password changes to the BIND hesupd daemon, providing a transparent distributed password program.

Note

> Depending on which naming services your LAN is running, the hosts file can be located in /etc, /var/yp/src, or /var/dss/namedb/src.

The following is a sample BIND/Hesiod primary server's hosts entry after bindsetup appends the bindmaster alias to it:

```
128.11.22.33 chicago.cities.dec.com chicago bindmaster
```

The original entry read as follows:

```
128.11.22.33 chicago.cities.dec.com chicago
```

See the *Guide to System and Network Setup* and the *Guide to the BIND/Hesiod Service* for more information on BIND/Hesiod.

2.1.2 Yellow Pages Service

Using YP you can distribute the following databases:

aliases
group
hosts
netgroup
networks
passwd
protocols
rpc
services

If you use YP to distribute the passwd database, your network must run at the BSD security level. Also, you must use YP to distribute the passwd database in a heterogeneous (multivendor) environment. See the *Guide to System and Network Setup* and the *Guide to the Yellow Pages Service* for more information on YP.

2.2 Time Services

The ULTRIX software supports the following time services:

- Network Time Protocol (NTP)
- Time Synchronization Protocol (TSP)

Both NTP and TSP are based on a client/server model, and can be used individually or together on your network. If TSP and NTP are used together, the TSP master must run the NTP daemon, ntpd. The TSP master then just acts as a distribution service for distributing NTP time to the network hosts running the TSP daemon, timed. It does not compute or distribute the average network time, as it would if TSP alone were running on your network.

For information on setting up the network time services, see the *Guide to System and Network Setup*.

2.2.1 Network Time Protocol

The Network Time Protocol (NTP) provides accurate, dependable, and synchronized time for hosts on both wide area networks (like the Internet) and local area networks. In particular, NTP provides synchronization traceable to clocks of high absolute accuracy, and avoids synchronization to clocks keeping bad time.

Hosts running NTP periodically exchange datagrams querying each other about their current estimate of the time. Using the round-trip time of the packet, a host can estimate the one-way delay to the other. (The assumption is that the delay is roughly equal in both directions.)

The one-way delay to the other host, in addition to the timestamps that are returned with the NTP packet, allow a host to compute the actual difference in its clock time and the time of the clock on the host that it queried.

A host queries a remote host several times over a period and feeds the results from the multiple samples to a digital-filtering algorithm. The algorithm provides a more accurate estimate of the delay, clock offset, and clock stability than could be obtained with a single sample.

NTP messages also contain information about the accuracy and reliability of the time sources. An NTP host connected directly to a highly accurate time source, such as a radio receiver tuned to a time code signal broadcast by a government agency, is called a stratum 1 server. Every other NTP host adopts a stratum number that is 1 higher than the host from which it sets its own time. For example, a host synchronized to a stratum 1 server becomes a stratum 2 host. Stratum determination is done automatically, and the stratum of a host can vary as its connectivity changes.

Hosts running NTP combine a variety of information, including the output of the digital-filtering algorithm and the stratum numbers of the hosts that it queried, to decide which of the several hosts provides the best time. By communicating with several other hosts, an NTP host can usually detect those hosts that are keeping bad time, and is able to stay synchronized even if some of the other hosts become unavailable for long periods.

In practice, NTP is able to synchronize clocks to within a few tens of milliseconds even over wide area networks spanning thousands of miles.

For detailed information on NTP, see RFC 1129—*Internet time synchronization: the Network Time Protocol.*

2.2.2 Time Synchronization Protocol

The Time Synchronization Protocol (TSP) is the protocol used by the `/etc/timed` daemon. In its simplest application, the TSP servers on a broadcast network (for example, an Ethernet) periodically broadcast TSP packets. The hosts on the network elect one of the hosts on the network running TSP as a master. The master then controls the further operation of the protocol until it fails and a new master is elected. The master collects time values from the other hosts (using the ICMP Timestamp protocol) and computes the average of all the times reported. It then sets its own clock to this average, and tells the other hosts to synchronize their clocks with it.

TSP quickly synchronizes all participating hosts. However, because TSP does not trace time back to sources of known accuracy, it is unable to correct for systematic errors. If a clock drifts significantly or a mistake is made in setting the time on a participating host, the average time calculated and distributed by the master can be affected significantly.

2.3 Authentication Services

Kerberos authentication is integrated into the BIND/Hesiod `named` daemon. The Kerberos-authenticated `named` guarantees that all Hesiod information that it distributes comes from a correct, specified source. You must run the Kerberos-authenticated `named` if you are configuring a distributed environment that is running at the UPGRADE or ENHANCED level of security.

Note

> To configure Kerberos on your system, you must have the optional ULTKERB400 software subset installed on a VAX machine, or the UDTKERB400 software subset installed on a RISC machine.
>
> If you did not install it when you first brought your system up, you must do so before attempting to configure Kerberos.

The Kerberos-authenticated `named` daemon and the `kerberos`(8) and `kprop`(8) commands depend on time being synchronized. If the time of day differs by more than five minutes between two systems running Kerberos, the Kerberos-authenticated `named` could fail. Therefore, if you intend to run Kerberos you should set up the network time services first.

2.4 Configurable Security Levels

The ULTRIX software can be configured in different security levels: BSD (the default), UPGRADE, and ENHANCED. The UPGRADE level is transitional between the BSD and ENHANCED levels.

Note

> To run your distributed environment at the UPGRADE or ENHANCED level you must have the ULTSEC400 and ULTKERB400 software subsets installed on a VAX machine, or the UDTSEC400 and UDTKERB400 software subsets installed on a RISC machine.
>
> If you did not install one or both subsets when you first brought your system up, you must do so before attempting to change your system's authentication status.

In the default BSD level, you can use either BIND/Hesiod or YP to distribute the various network databases. In the UPGRADE or ENHANCED level, however, the `passwd` and `auth` databases are used together, and must be distributed by BIND/Hesiod with Kerberos configured. (Other databases can still be distributed by either BIND/Hesiod or YP.) If you are running at the UPGRADE or ENHANCED level, but are not distributing the `passwd` and `auth` databases, your system can run with local `/etc/passwd` and `/etc/auth` files.

The `auth` database, which is accessible to only a select group of programs, contains security-related user information. Programs such as `passwd`, `login`, `su`, `dxsession`, and `named` use the Kerberos-authenticated `named` daemon to access the `auth` database. See the `auth`(5) reference page and the *Security Guide for Administrators* for more information.

2.5 Planning a Distributed Environment

This section briefly describes the processes that should run on each system in a distributed environment that is running at the UPGRADE or ENHANCED security level.

View your distributed environment as a set of interrelated processes that work in a coordinated manner. Although it is a good idea to designate the same machine or machines as the master for the time, naming, and authentication services, it is not required.

For information on the specific steps involved in making the transition from the BSD security level to the ENHANCED security level see the *Guide to Kerberos*.

The following list summarizes the processes that should run on the systems in a distributed environment:

- Each system should be a `timed` master or slave, or a `ntpd` master or slave, or both. Select the most secure and best-administered machines to run the `ntpd` daemon. Because a WWV radio clock is the most accurate, the master `ntpd` should reference one, if possible.

  The master time servers should also run `timed` as masters. The remainder of the systems in the distributed environment run `timed` as slaves, receiving the distributed time from the masters. The resulting combined time service is less vulnerable to the drifting nature of `timed` alone, and is easier to administer than NTP alone.

  See the *Guide to System and Network Setup* for information on setting up the network time services.

- It is strongly recommended that each BIND/Hesiod secondary server also be a Kerberos slave server. If your services are not configured in this way, establish at least one backup server to run as both a BIND/Hesiod secondary server and a Kerberos slave server.

  As the number of BIND/Kerberos servers increases, the propagation delay of new Kerberos or BIND/Hesiod databases increases. Consider creating separate domains or realms of management if the number of servers is more than 25.

  See the *Guide to the BIND/Hesiod Service* and the *Guide to System and Network Setup* for information on setting up the BIND/Hesiod service. See the *Guide to Kerberos* for information on setting up a Kerberos master and slaves.

- Each time you write a program that will use `kerberos`, you must add an entry to the `/etc/services` file, create a principal entry in the Kerberos database, create a new `srvtab` file for each system on which you are going to run the program, and copy `/etc/krb.conf` to each system on which you are going to run the program.

  See the *Guide to Network Programming* for information on writing applications that use Kerberos.

Table 2-1 illustrates a typical configuration for an authenticated environment. In this table, the BIND/Hesiod primary server and the Kerberos master are the same system. The table indicates the role of each system, and what processes and specific system files each requires to perform its role.

Table 2-1: Processes in an Authenticated Environment

Kerberos	BIND	Daemons and Data
Kerberos master	BIND primary server	Kerberos master database BIND master database Authenticated named kerberos ntpd Master timed
Kerberos slave server	BIND secondary server	Propagated Kerberos database Propagated BIND database Authenticated named kerberos Master timed ntpd
Kerberos client	BIND slave server	Authenticated named timed

Managing a Network with SNMP 3

The Simple Network Management Protocol (SNMP) is the de facto industry standard for managing TCP/IP networks. The protocol defines the role of a Network Management Station (NMS) and an SNMP Agent, allowing remote users on an NMS to monitor and manage TCP/IP network entities. The entities that the NMS manages and controls are defined in the Management Information Base (MIB).

For information on the structure of the MIB and the network entities that are globally defined, see Appendix B and RFC 1066—*Management Information Base for Network Management of TCP/IP-based internets.*

On ULTRIX systems users can also define site-specific network entities in a private MIB by means of an Extended SNMP Agent. The Extended Agents are also managed by the NMS. For information on defining entries in a private MIB, see the *Guide to Network Programming.*

This chapter provides an overview of SNMP and information on how to set up the SNMP Agent and Extended Agent.

Note

> The ULTRIX operating system supports an implementation of the SNMP Agent. It does not implement the NMS software.
>
> Systems running the ULTRIX operating system can be remotely managed by an NMS running other vendors' NMS software.

3.1 Overview

This section provides an overview of the following components of the Simple Network Management Protocol (SNMP):

- The ULTRIX SNMP Agent
- The ULTRIX Extended SNMP Agent
- SNMP communities
- The Management Information Base

3.1.1 ULTRIX SNMP Agent

The ULTRIX SNMP Agent, `/etc/snmpd`, is a daemon that runs on the host that is being managed by the NMS. The Agent software is installed locally.

Information passed between the Agent and the NMS is obtained from the MIB, which contains a collection of data that represents those objects being managed. Appendix B shows the structure of the MIB.

The Agent is responsible for the following tasks:

- Answering remote requests for information from the NMS
- Setting MIB variables as requested by the NMS
- Generating the appropriate traps
- Checking that the community name of the NMS requesting information matches the community name defined in its `snmpd.conf` file

3.1.2 ULTRIX Extended SNMP Agent

The ULTRIX Extended SNMP Agent is a logical extension of the ULTRIX SNMP Agent. The Agent and Extended Agent appear as a single entity to the NMS. However, the Extended Agent allows users to define site-specific network entities for the NMS to manage. The site-specific entities are defined in a private MIB that is specific to the Extended Agent.

For information on setting up and managing a private MIB, see the *Guide to Network Programming.*

3.1.3 SNMP Communities

SNMP uses **communities** to achieve access authentication between the NMS and the Agent. The `community` variable is located in the Agent's `/etc/snmpd.conf` file and defines the following:

- Community name

  The community name can be any string of up to 127 characters that is known to both the Agent and the NMS.

- Internet address of the NMS from which the Agent accept requests

  The Internet address can be either 0.0.0.0 or the Internet address of the NMS. If you specify 0.0.0.0 as the Internet address, the SNMP Agent honors the request from any NMS that knows the community name.

- Community type

  Communities can be read-only, read-write, or traps:

  - Read-only communities are monitored, but not managed, by an NMS.
  - Read-write communities are managed by an NMS.
  - Traps are unsolicited messages generated by the Agent that guide the polling from the NMS.

Caution

Use caution when defining the `community` parameter in the Agent's `/etc/snmpd.conf` file. If the Internet address is specified as 0.0.0.0, and the community type is read-write, any NMS knowing the community name can view and change parameters on the Agent.

The Agent verifies that the NMS performing the request is allowed access to the contents of the Agent's MIB by checking whether the community name in its `/etc/snmpd.conf` file matches the community name in the incoming Protocol

Data Unit (PDU). It also checks that the Internet address of the NMS in the incoming PDU matches that of the Internet address in the Agent's `/etc/snmpd.conf` file.

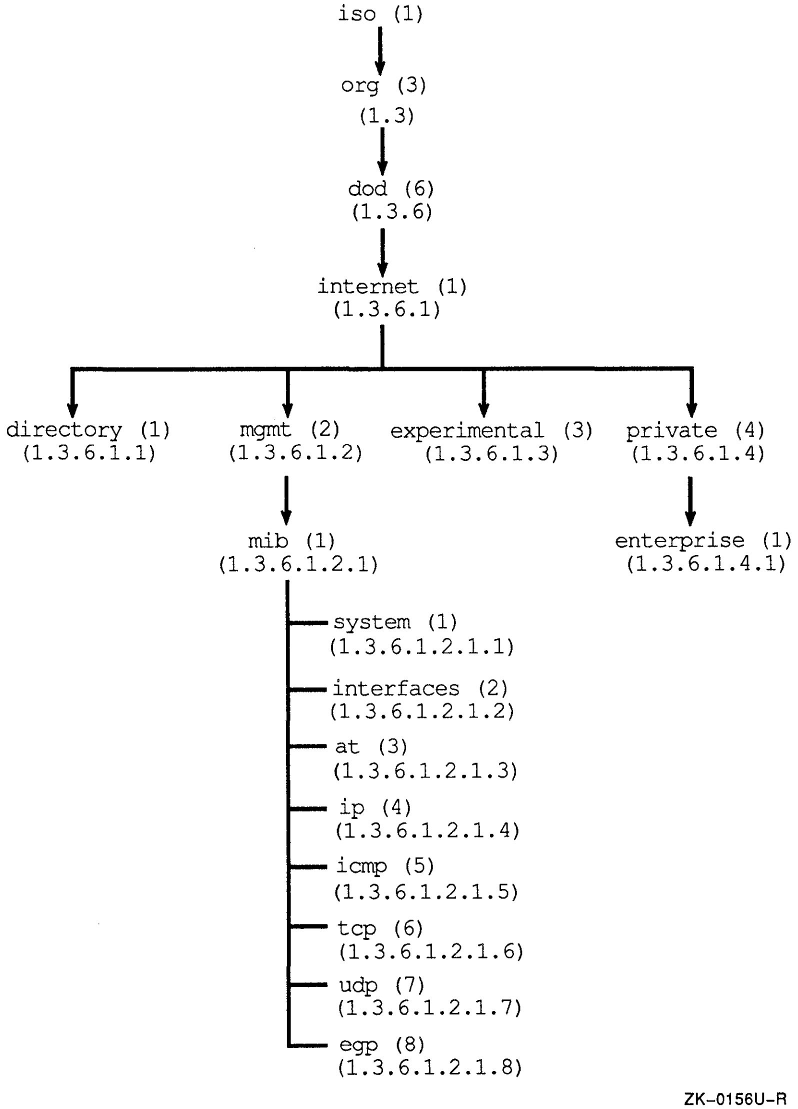
For more information see the `snmpd.conf`(5n) reference page.

3.1.4 Management Information Base

The data passed between the NMS and the Agent is obtained from the MIB. The MIB is the repository for the data collected and modified in the Agent. Elements in the MIB are interrogated by the NMS. If the NMS needs to set a variable, it makes a request to the Agent; the Agent responds by setting the requested variable.

The MIB tree is subdivided into several subtrees, each of which defines a certain class of objects. Figure 3-1 outlines the global structure of the MIB. Appendix B contains maps of the entire MIB tree.

Figure 3-1: Global Structure of the MIB



Objects in the MIB are defined with Abstract Syntax Notation One (ASN.1). Each object type has a specific name, syntax, and encoding. For detailed information on object types see RFC 1066—*Management Information Base for Network Management of TCP/IP-based internets.*

3.2 Retrieving a MIB Variable

The Agent is activated when the system boots. The Agent then reads the list of Extended Agents in the `/etc/snmpd.conf` file and activates them.

When an Extended Agent is activated, it creates a socket for communicating with the Agent. Once activated, the Extended Agent sends the Agent a registration message. The message contains information about the Extended Agent's supported MIB and the community name. The Agent uses the information in the registration message to authenticate (loosely) an SNMP request. It returns an Agent Confirmation Message to the Extended Agent that contains information about the Agent itself, and indicates whether the registration succeeded or failed. See the *Guide to Network Programming* for information on constructing Extended Agents.

When the Agent receives a request for a MIB variable from the NMS, it looks at its MIB variable tree for a process to retrieve the desired variable. If the variable is part of the Agent, the process within the Agent retrieves the variable and returns it to the NMS.

If, however, the variable is external to the Agent, the Agent constructs an Agent Request Message that it sends to the Extended Agent. When it sends the message, the Agent starts a timer.

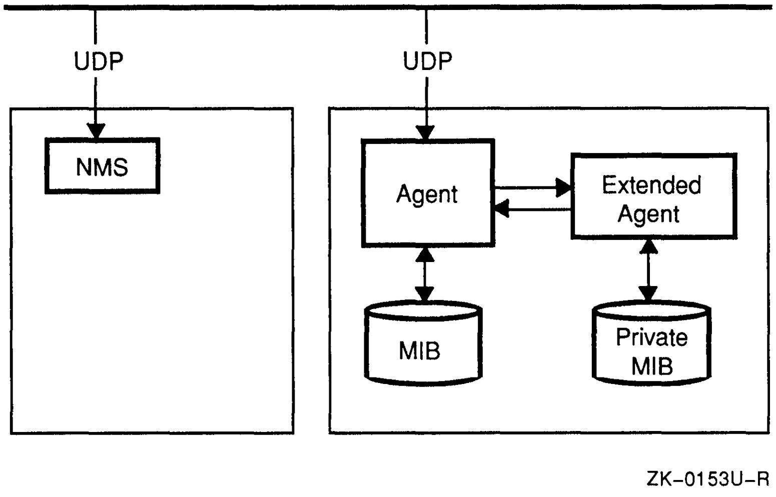
When the Extended Agent receives the Agent Request Message, it checks for the validity of the requested variable. If the check passes, the Extended Agent retrieves the variable and constructs an Agent Response Message that it sends to the Agent. When the Agent receives the Agent Response Message, it returns the requested variable to the NMS.

If the check fails, the Extended Agent returns an Agent Error Message to the Agent.

If the Agent does not receive a response from the Extended Agent within a specified period of time, the Agent returns an error to the NMS.

Figure 3-2 depicts the relationship between the NMS, Agent, Extended Agent, and MIB.

Figure 3-2: Relationship of SNMP Components



3.3 Setting Up SNMP Manually

You can set up SNMP either using the snmpsetup command or manually.

The snmpsetup command automates the task of configuring the SNMP Agent and user-defined Extended Agents on your system. For more information on setting up SNMP with snmpsetup, see the *Guide to System and Network Setup.*

You can set SNMP up on your system manually by editing the /etc/snmpd.conf file. The /etc/snmpd.conf file is a configuration file that contains information used by the snmpd daemon to define the static variables whose values are not available in the kernel.

A default /etc/snmpd.conf file is created for you by the ULTRIX installation. It contains the following entry:

```
community public 0.0.0.0 read-only
```

This entry enables any NMS to monitor your system.

You can customize this file to suit your system's needs. For information on the variables you can define, see the snmpd.conf(5n) reference page and RFC 1066—*Management Information Base for Network Management of TCP/IP-based Internets.*

Local Area Network Files A

This appendix describes the following files that are used by the networking software:

- .rhosts
- /etc/hosts
- /etc/hosts.equiv
- /etc/inetd.conf
- /etc/networks
- /etc/protocols
- /etc/rc and /etc/rc.local
- /etc/services
- /etc/svc.conf
- /usr/hosts
- /usr/spool/mqueue
- /usr/spool/rwho

A.1 The .rhosts File

The .rhosts file allows a user who has an account on your system to log in from a remote system without supplying a password. It also allows remote copies to the local host.

Each user should create a .rhosts file in his or her home directory.

The format of a .rhosts file entry is:

hostname [*username*]

The host name is the name of the remote system that the user wants to log in to without supplying a password. The user name is the user's login name. If the user does not specify a user name, the user must have the same login name on both the remote and the local system.

For example, if user ginger, whose local system is host1, wants to log in to system machine1 without supplying a password, she must do the following:

- Have an account on machine1
- Create a .rhosts file in her home directory on machine1
- Specify host1 ginger as an entry in the .rhosts file

  If user ginger has the same login name on both host1 and machine1, she can simply specify host1 in her .rhosts file entry.

Note

You can allow the superuser of a remote system to log in to your system without password protection by having a `.rhosts` file in the root (/) directory, but it is not recommended practice.

In addition to having a `.rhosts` file, the superuser needs a `secure` terminal entry in the `/etc/ttys` file for each pseudoterminal configured in the system. The secure entry looks similar to the following:

```
ttyp3      none    network         secure
```

See the `ttys`(5) reference page for more information.

A.2 The /etc/hosts File

The `/etc/hosts` file contains a list of the known hosts on the Internet network. Each entry in the `/etc/hosts` file consists of a host's Internet address, its official system name, and any aliases by which that host can be known. The following is a typical `/etc/hosts` entry:

```
137.174.7.22 host1 h1
```

To add a host system to your LAN, add an entry for that system to the `/etc/hosts` file. Likewise, to remove a host system from your LAN, delete that system's entry from the `/etc/hosts` file. See the `hosts`(5) reference page for more information.

If your LAN is managed with a naming service such as BIND/Hesiod or Yellow Pages (YP), see the *Guide to the BIND/Hesiod Service* or the *Guide to the Yellow Pages Service.*

A.3 The /etc/hosts.equiv File

The `/etc/hosts.equiv` file allows all users (except for `root`) on a remote system to log in to your system without supplying a password, if the following are true:

- The remote system has an entry in `hosts.equiv`.
- The user has an account on your system.

The `/etc/hosts.equiv` file lists the systems that are logically equivalent to the local system, and are therefore treated exactly the same as it. This means that the logically equivalent system has all of the privileges of the local system. However, the equivalency is not automatically reciprocal. If the system `host1` specifies `host2` in its `/etc/hosts.equiv` file, `host2` is logically equivalent to `host1`. But unless `host2` specifies `host1` in its `/etc/hosts.equiv` file, `host1` is not considered logically equivalent to `host2`.

For security reasons, logically equivalent systems are usually run by the same administration.

See the `hosts.equiv`(5) reference page for more information.

A.4 The /etc/inetd.conf File

The /etc/inetd.conf file contains entries for the standard services. It also includes information regarding the standard services that inetd processes by opening sockets and listening for requests. You do not need to modify the /etc/inetd.conf file unless you want to add a nonstandard service or remove a service from your LAN.

The services that you should include in /etc/inetd.conf and use on your system depend on the size of your system configuration. On most systems, however, the minimum services are login or telnet, shell, exec, and talk.

Other frequently used services, in order of their importance, include time, ftp, and tftp.

Note

> Because tftp does not require user authentication, it is not very secure. Enable it in the /etc/inetd.conf file only if necessary.

To add a service, edit the /etc/inetd.conf file and create a new entry. To comment out an existing service, edit the /etc/inetd.conf file and insert a comment character (#) at the beginning of the appropriate entry.

For the changes to take effect you must either reboot the system or use the kill command. To use the kill command (which sends the daemon a hangup signal), do the following:

1. Log in to your system as root, or become superuser.
2. Find the process identification number (PID) of the inetd daemon:

   ```
   # ps -ax | grep inetd
   ```

   The system displays a line similar to the following:

   ```
   282 ?  I     1:50 /etc/inetd
   ```

   The PID in this example is 282.
3. Kill the /etc/inetd process using the following command:

   ```
   # kill -1 282
   ```

   The kill -1 command specifies the hangup signal.

See the inetd.conf(5) reference page for more information.

A.5 The /etc/networks File

The /etc/networks file contains information about the known networks that compose the Internet. You do not need to modify the /etc/networks file unless you want to add or remove a network from your system. See the networks(5) reference page for more information.

A.6 The /etc/protocols File

The /etc/protocols file contains information about the known protocols used in the Internet. You do not need to modify the /etc/protocols file unless you want to add a nonstandard Internet protocol to your LAN. See the protocols(5) reference page for more information.

A.7 The /etc/rc and /etc/rc.local Files

As distributed, the `/etc/rc` and `/etc/rc.local` files list a number of entries that are used to set up the network when the system is brought to multiuser mode. Specifically, the `/etc/rc` and `/etc/rc.local` files do the following:

- Initialize the network device
- Initialize the loopback device
- Start network daemons

If you are using the network, the `/etc/rc` or `/etc/rc.local` file starts these daemons:

- `syslog`
- `inetd`
- `sendmail`

As it starts up each daemon, the `/etc/rc` or `/etc/rc.local` file echoes the name of the daemon to the console. Therefore, while your system is going to multiuser mode, your console lists those daemons that currently are up and running.

As distributed, the `/etc/rc.local` file does not set the date and time from the network. If you want the date and time to be set, edit the `/etc/rc.local` file and remove the comment character (#) from the beginning of the line that lists the `/etc/rdate` entry.

If you have a number of systems on your network, you can either turn on the `rwhod` daemon or specify `/etc/rwhod -b`, which tells the `rwho` daemon to ignore incoming `rwho` packets. See the `rwhod`(8c) reference page for more information about the `rwhod` daemon. In addition, if you are using a network that consists of multiple Ethernet segments, you can turn on the `routed` daemon.

The `/etc/rc.local` file contains initialization information regarding the various networks to which your system is connected. The first entry in the `/etc/rc.local` file should be for the primary network interface.

If you add a network interface, such as a DEUNA, then you need to add an `ifconfig` line to the `/etc/rc.local` file.

The `/etc/rc.local` file also needs an entry for the loopback interface, which must follow all of the network interface entries. The loopback interface allows connections between programs on your local host.

The following are example lines from `host1`'s `/etc/rc.local` file. It specifies two DEUNA interfaces (de0 and de1), a DMR-11 interface (dmc0), and a loopback interface (lo0):

```
/etc/ifconfig de0 `/bin/hostname` broadcast 128.45.255.255 \
                                                   netmask 255.255.0.0
/etc/ifconfig dmc0 host1a host2 netmask 255.255.0.0
/etc/ifconfig de1 host1b   broadcast 98.1.255.255  netmask 255.255.0.0
/etc/ifconfig lo0 localhost
```

The system `host1` now has three hardware interfaces (de0, dmc0, de1) and one software interface (lo0).

Note that the entries configure systems on two different classes of network, a Class B network (128.45.255.255) and a Class A network (98.1.255.255). The netmask

255.255.0.0 is the default for a Class B network, meaning that fields 1 and 2 express the network number.

The netmask 255.255.0.0 for the Class A network, however, indicates that field 2 is used to express the subnetwork address.

Because the first device to be configured becomes the default network device for sockets, it is important that the loopback driver be configured after the real network devices. The default network should not be the local loopback pseudodevice if your system is connected to a network.

A.8 The /etc/services File

The `/etc/services` file contains information regarding the known services available in the Internet. You do not need to modify the `/etc/services` file unless you want to add a nonstandard Internet service to your LAN. See the `services`(5) reference page for more information.

A.9 The /etc/svc.conf File

The `/etc/svc.conf` file defines the order in which to query the name services running on your system. It is a mandatory system file that is created when you install the ULTRIX software. If you want to use the BIND/Hesiod service or Yellow Pages, you must edit the `/etc/svc.conf` file with the necessary database and service order information. The following is a typical entry in the `/etc/svc.conf` file:

```
passwd=local,bind
```

This entry tells the system to search first locally for password information. If it can not find the information locally, the system then queries a BIND/Hesiod server.

Note

> It is recommended that you list `local` as the first service for all databases.

You can specify any of the following databases in the `svc.conf` file:

- `aliases`
- `auth`
- `group`
- `hosts`
- `netgroup` (served only by Yellow Pages)
- `networks`
- `passwd`
- `protocols`
- `rpc`
- `services`

See the `svc.conf`(5) reference page for more information.

A.10 The /usr/hosts Directory

The /usr/hosts directory allows a user to log in to a remote host by typing only the name of the remote host under the following conditions:

- The /usr/hosts directory is included in the search path in a user's .login or .profile file.
- The system administrator ran the /usr/hosts/MAKEHOSTS command after the network was set up and running.

  Additionally, when a new host is added to the /etc/hosts file, the BIND/Hesiod primary server's hosts database, or the YP master server's hosts database (if you are running BIND/Hesiod or YP), the system administrator must rerun the /usr/hosts/MAKEHOSTS command.

A.11 The /usr/spool/mqueue Directory

The /usr/spool/mqueue directory contains the network syslog files. Each syslog file contains the daemon error messages and information from utilities such as sendmail, the remote installation, and Ethernet remote node maintenance functions that occurred during a day's operation.

The syslog files are automatically maintained by the /usr/adm/newsyslog program, which is run once a day by /etc/cron.

See the syslog(8) reference page for more information on the syslog utility.

Normally you do not have to alter the times when /usr/adm/newsyslog is executed. However, if you want to run /usr/adm/newsyslog more often or less often, edit /etc/crontab and make the appropriate time changes.

See the cron(8) reference page for more information on the format of the /etc/crontab file.

A.12 The /usr/spool/rwho Directory

The /usr/spool/rwho directory contains a file named whod.*system-name* for each system connected to your LAN. The data for the rwho and ruptime commands is saved in this directory.

If you remove a system from your local network, you should remove that system's file from the /usr/spool/rwho directory. This clears the directory of unneeded files and helps keep the rwho and ruptime information up to date.

The Management Information Base B

This appendix contains tree diagrams of all objects in the MIB.

Some MIB objects are available only if the ULTRIX `routed` daemon is running. Those objects are marked by an **(r)** in front of the object name.

Figure B-1: Global Structure of the MIB

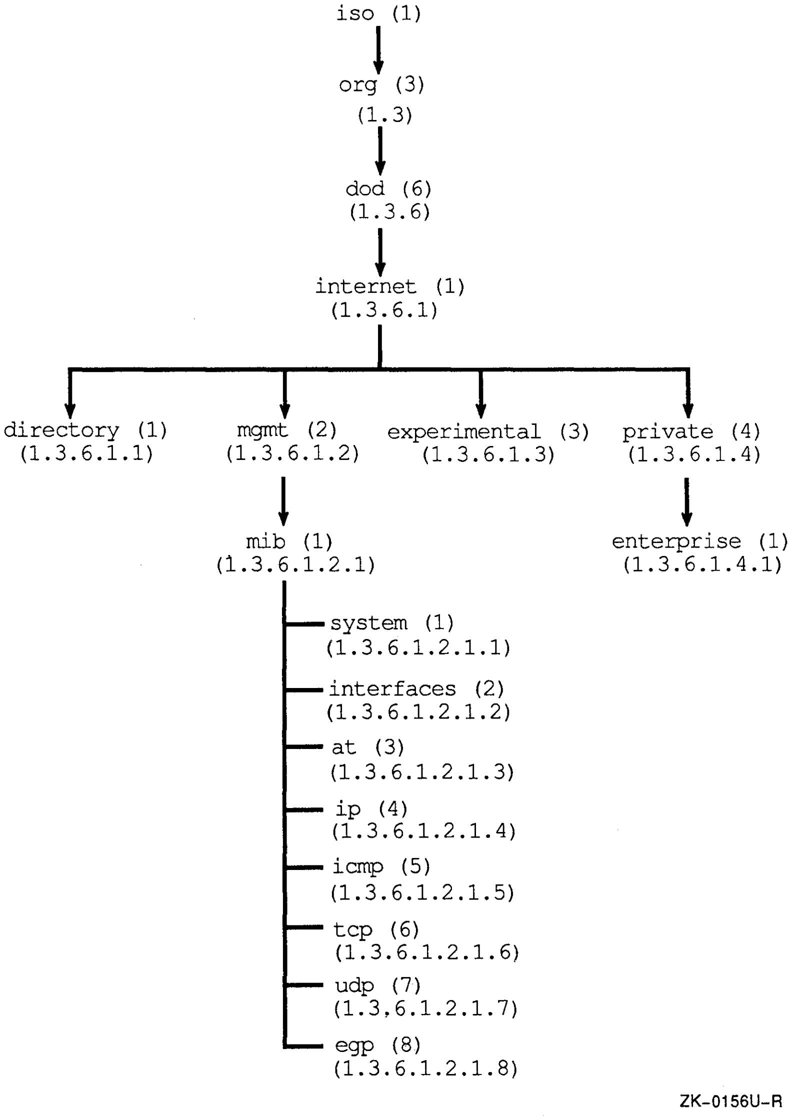


Figure B-2: Systemwide MIB Objects

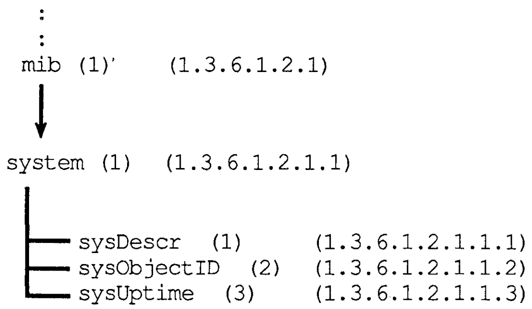

:
:
mib (1)' (1.3.6.1.2.1)
system (1) (1.3.6.1.2.1.1)
sysDescr (1) (1.3.6.1.2.1.1.1)
sysObjectID (2) (1.3.6.1.2.1.1.2)
sysUptime (3) (1.3.6.1.2.1.1.3)

Figure B-3: Interface-Managed MIB Objects

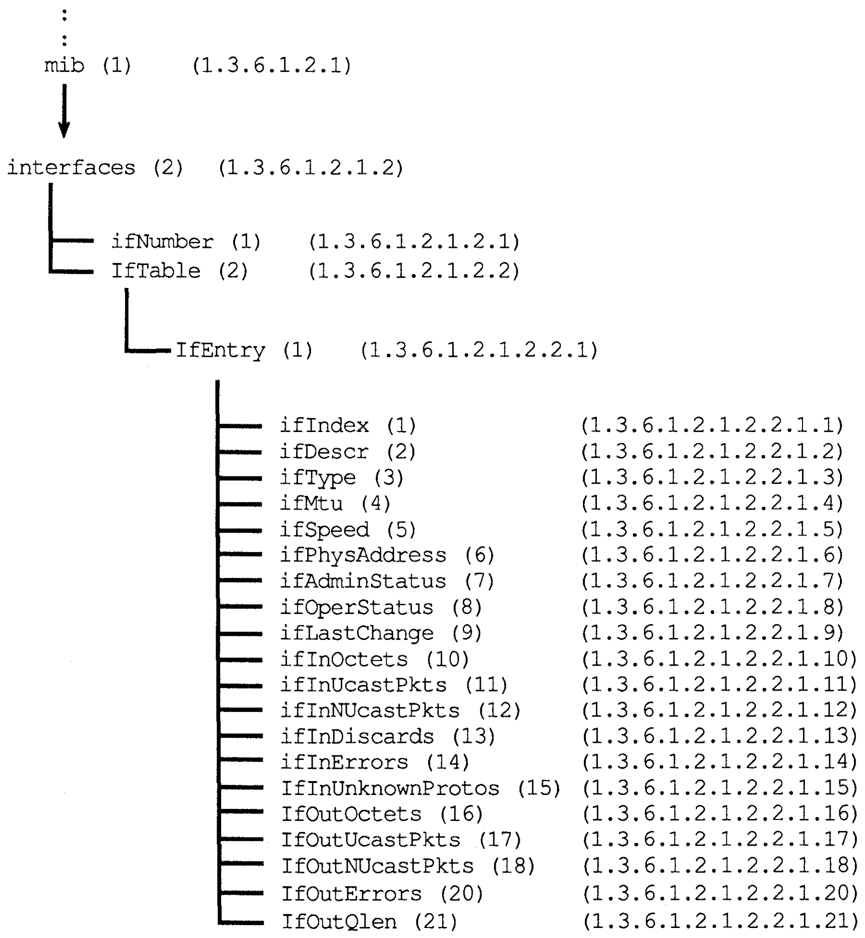

:
:
mib (1) (1.3.6.1.2.1)
interfaces (2) (1.3.6.1.2.1.2)
ifNumber (1) (1.3.6.1.2.1.2.1)
IfTable (2) (1.3.6.1.2.1.2.2)
IfEntry (1) (1.3.6.1.2.1.2.2.1)
ifIndex (1) (1.3.6.1.2.1.2.2.1.1)
ifDescr (2) (1.3.6.1.2.1.2.2.1.2)
ifType (3) (1.3.6.1.2.1.2.2.1.3)
ifMtu (4) (1.3.6.1.2.1.2.2.1.4)
ifSpeed (5) (1.3.6.1.2.1.2.2.1.5)
ifPhysAddress (6) (1.3.6.1.2.1.2.2.1.6)
ifAdminStatus (7) (1.3.6.1.2.1.2.2.1.7)
ifOperStatus (8) (1.3.6.1.2.1.2.2.1.8)
ifLastChange (9) (1.3.6.1.2.1.2.2.1.9)
ifInOctets (10) (1.3.6.1.2.1.2.2.1.10)
ifInUcastPkts (11) (1.3.6.1.2.1.2.2.1.11)
ifInNUcastPkts (12) (1.3.6.1.2.1.2.2.1.12)
ifInDiscards (13) (1.3.6.1.2.1.2.2.1.13)
ifInErrors (14) (1.3.6.1.2.1.2.2.1.14)
IfInUnknownProtos (15) (1.3.6.1.2.1.2.2.1.15)
IfOutOctets (16) (1.3.6.1.2.1.2.2.1.16)
IfOutUcastPkts (17) (1.3.6.1.2.1.2.2.1.17)
IfOutNUcastPkts (18) (1.3.6.1.2.1.2.2.1.18)
IfOutErrors (20) (1.3.6.1.2.1.2.2.1.20)
IfOutQlen (21) (1.3.6.1.2.1.2.2.1.21)

Figure B-4: Address Translation Table-Managed MIB Objects

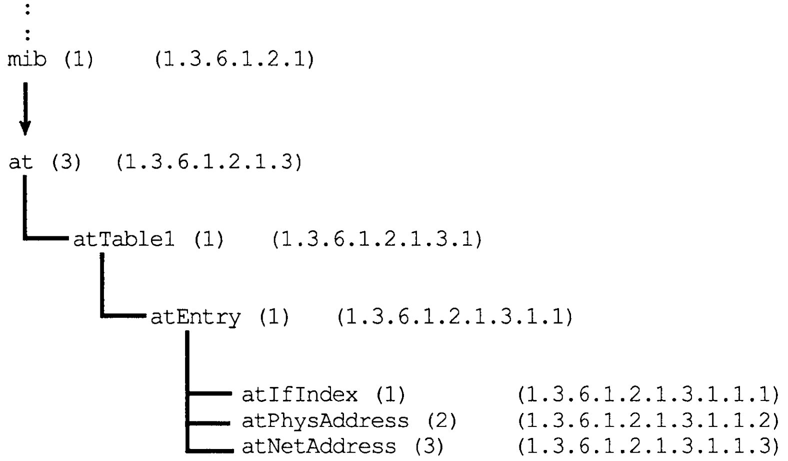

:
:
mib (1) (1.3.6.1.2.1)
at (3) (1.3.6.1.2.1.3)
atTable1 (1) (1.3.6.1.2.1.3.1)
atEntry (1) (1.3.6.1.2.1.3.1.1)
atIfIndex (1) (1.3.6.1.2.1.3.1.1.1)
atPhysAddress (2) (1.3.6.1.2.1.3.1.1.2)
atNetAddress (3) (1.3.6.1.2.1.3.1.1.3)

Figure B-5: Internet Protocol-Managed MIB Objects

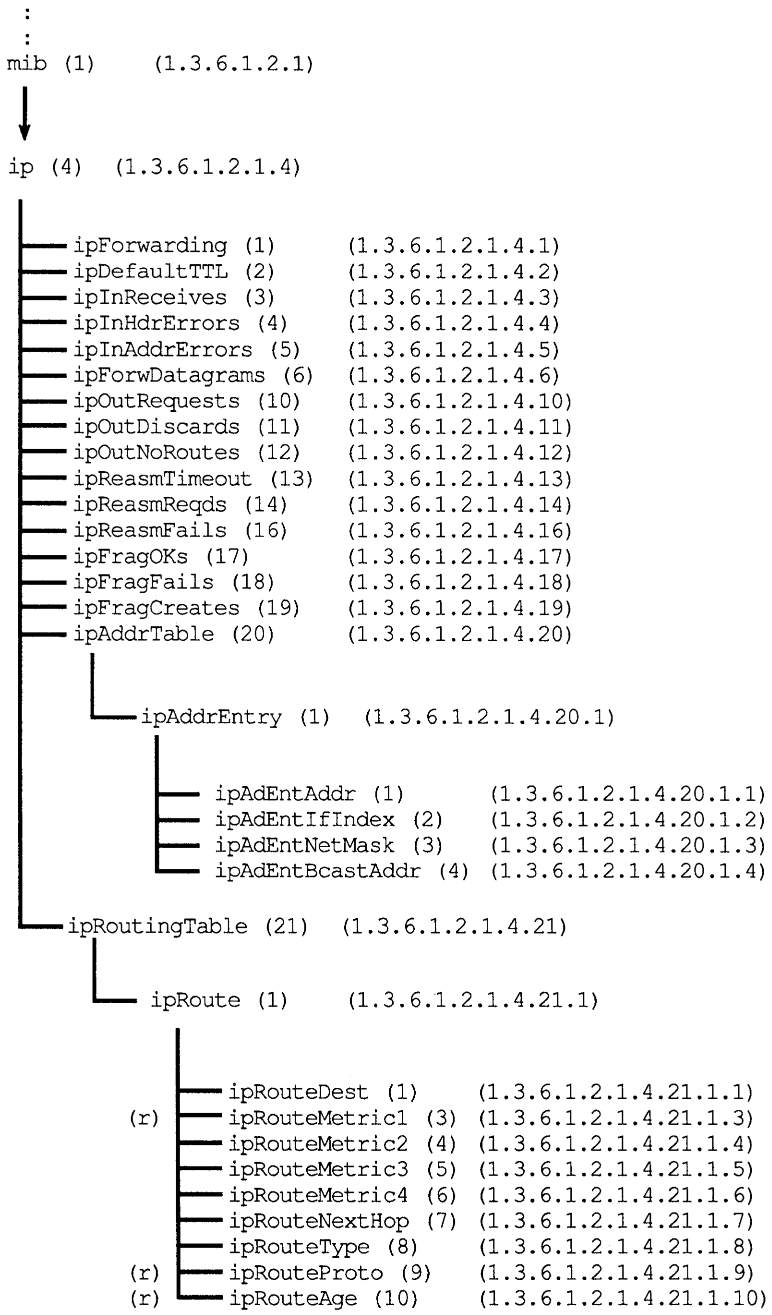
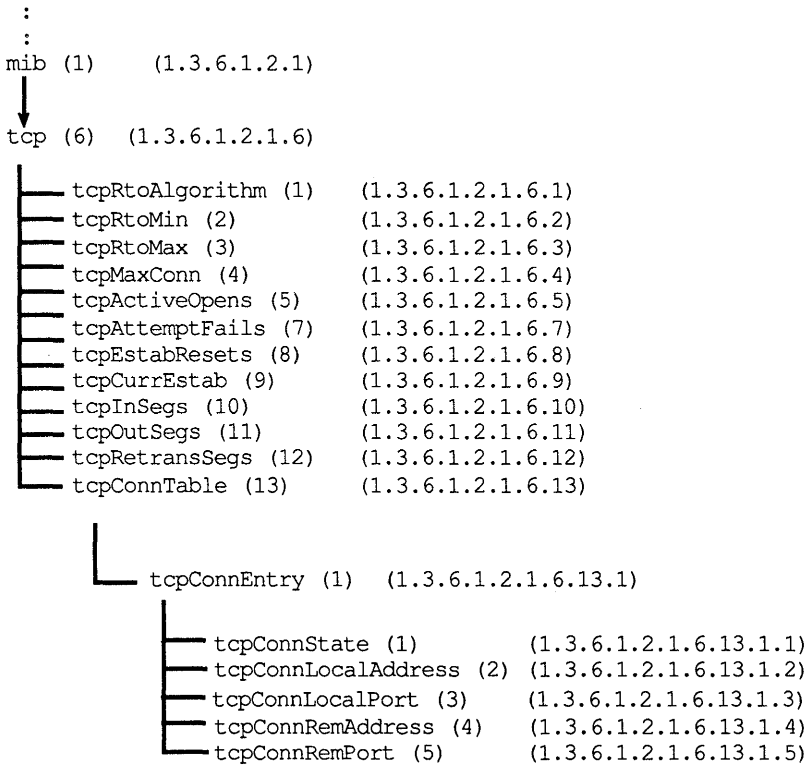

:
:
mib (1) (1.3.6.1.2.1)
ip (4) (1.3.6.1.2.1.4)
ipForwarding (1) (1.3.6.1.2.1.4.1)
ipDefaultTTL (2) (1.3.6.1.2.1.4.2)
ipInReceives (3) (1.3.6.1.2.1.4.3)
ipInHdrErrors (4) (1.3.6.1.2.1.4.4)
ipInAddrErrors (5) (1.3.6.1.2.1.4.5)
ipForwDatagrams (6) (1.3.6.1.2.1.4.6)
ipOutRequests (10) (1.3.6.1.2.1.4.10)
ipOutDiscards (11) (1.3.6.1.2.1.4.11)
ipOutNoRoutes (12) (1.3.6.1.2.1.4.12)
ipReasmTimeout (13) (1.3.6.1.2.1.4.13)
ipReasmReqds (14) (1.3.6.1.2.1.4.14)
ipReasmFails (16) (1.3.6.1.2.1.4.16)
ipFragOKs (17) (1.3.6.1.2.1.4.17)
ipFragFails (18) (1.3.6.1.2.1.4.18)
ipFragCreates (19) (1.3.6.1.2.1.4.19)
ipAddrTable (20) (1.3.6.1.2.1.4.20)
ipAddrEntry (1) (1.3.6.1.2.1.4.20.1)
ipAdEntAddr (1) (1.3.6.1.2.1.4.20.1.1)
ipAdEntIfIndex (2) (1.3.6.1.2.1.4.20.1.2)
ipAdEntNetMask (3) (1.3.6.1.2.1.4.20.1.3)
ipAdEntBcastAddr (4) (1.3.6.1.2.1.4.20.1.4)
ipRoutingTable (21) (1.3.6.1.2.1.4.21)
ipRoute (1) (1.3.6.1.2.1.4.21.1)
ipRouteDest (1) (1.3.6.1.2.1.4.21.1.1)
(r) ipRouteMetric1 (3) (1.3.6.1.2.1.4.21.1.3)
ipRouteMetric2 (4) (1.3.6.1.2.1.4.21.1.4)
ipRouteMetric3 (5) (1.3.6.1.2.1.4.21.1.5)
ipRouteMetric4 (6) (1.3.6.1.2.1.4.21.1.6)
ipRouteNextHop (7) (1.3.6.1.2.1.4.21.1.7)
ipRouteType (8) (1.3.6.1.2.1.4.21.1.8)
(r) ipRouteProto (9) (1.3.6.1.2.1.4.21.1.9)
(r) ipRouteAge (10) (1.3.6.1.2.1.4.21.1.10)

Figure B-6: Internet Control Message Protocol-Managed MIB Objects

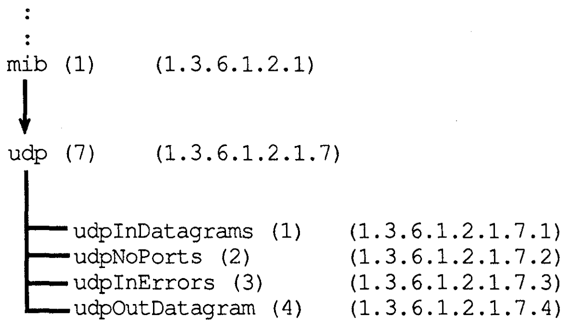
```
 :
 :
mib (1)    (1.3.6.1.2.1)
 |
 v
icmp (5)  (1.3.6.1.2.1.5)
 |
 |---icmpInMsgs (1)             (1.3.6.1.2.1.5.1)
 |---icmpInErrors (2)           (1.3.6.1.2.1.5.2)
 |---icmpInDestUnreachs (3)     (1.3.6.1.2.1.5.3)
 |---icmpInTimeExcds (4)        (1.3.6.1.2.1.5.4)
 |---icmpInParmProbs (5)        (1.3.6.1.2.1.5.5)
 |---icmpInSrcQuenchs (6)       (1.3.6.1.2.1.5.6)
 |---icmpInRedirects (7)        (1.3.6.1.2.1.5.7)
 |---icmpInEchos (8)            (1.3.6.1.2.1.5.8)
 |---icmpInEchoReps (9)         (1.3.6.1.2.1.5.9)
 |---icmpInTimestamps (10)      (1.3.6.1.2.1.5.10)
 |---icmpInTimestampReps (11)   (1.3.6.1.2.1.5.11)
 |---icmpInAddrMasks (12)       (1.3.6.1.2.1.5.12)
 |---icmpInAddrMaskReps (13)    (1.3.6.1.2.1.5.13)
 |---icmpOutMsgs (14)           (1.3.6.1.2.1.5.14)
 |---icmpOutErrors (15)         (1.3.6.1.2.1.5.15)
 |---icmpOutDestUnreachs (16)   (1.3.6.1.2.1.5.16)
 |---icmpOutTimeExcds (17)      (1.3.6.1.2.1.5.17)
 |---icmpOutParmProbs (18)      (1.3.6.1.2.1.5.18)
 |---icmpOutSrcQuenchs (19)     (1.3.6.1.2.1.5.19)
 |---icmpOutRedirects (20)      (1.3.6.1.2.1.5.20)
 |---icmpOutEchos (21)          (1.3.6.1.2.1.5.21)
 |---icmpOutEchoReps (22)       (1.3.6.1.2.1.5.22)
 |---icmpOutTimestamps (23)     (1.3.6.1.2.1.5.23)
 |---icmpOutTimeStampReps (24)  (1.3.6.1.2.1.5.24)
 |---icmpOutAddrMasks (25)      (1.3.6.1.2.1.5.25)
 |___icmpOutAddrMaskReps (26)   (1.3.6.1.2.1.5.26)
```

Figure B-7: Transmission Control Protocol-Managed MIB Objects



```
:
:
mib (1)     (1.3.6.1.2.1)
 |
tcp (6)   (1.3.6.1.2.1.6)
 |
 |-- tcpRtoAlgorithm (1)    (1.3.6.1.2.1.6.1)
 |-- tcpRtoMin (2)          (1.3.6.1.2.1.6.2)
 |-- tcpRtoMax (3)          (1.3.6.1.2.1.6.3)
 |-- tcpMaxConn (4)         (1.3.6.1.2.1.6.4)
 |-- tcpActiveOpens (5)     (1.3.6.1.2.1.6.5)
 |-- tcpAttemptFails (7)    (1.3.6.1.2.1.6.7)
 |-- tcpEstabResets (8)     (1.3.6.1.2.1.6.8)
 |-- tcpCurrEstab (9)       (1.3.6.1.2.1.6.9)
 |-- tcpInSegs (10)         (1.3.6.1.2.1.6.10)
 |-- tcpOutSegs (11)        (1.3.6.1.2.1.6.11)
 |-- tcpRetransSegs (12)    (1.3.6.1.2.1.6.12)
 |-- tcpConnTable (13)      (1.3.6.1.2.1.6.13)
       |
       |-- tcpConnEntry (1)  (1.3.6.1.2.1.6.13.1)
             |
             |-- tcpConnState (1)         (1.3.6.1.2.1.6.13.1.1)
             |-- tcpConnLocalAddress (2)  (1.3.6.1.2.1.6.13.1.2)
             |-- tcpConnLocalPort (3)     (1.3.6.1.2.1.6.13.1.3)
             |-- tcpConnRemAddress (4)    (1.3.6.1.2.1.6.13.1.4)
             |-- tcpConnRemPort (5)       (1.3.6.1.2.1.6.13.1.5)
```

Figure B-8: User Datagram Protocol-Managed MIB Objects



```
:
:
mib (1)     (1.3.6.1.2.1)
 |
udp (7)     (1.3.6.1.2.1.7)
 |
 |-- udpInDatagrams (1)   (1.3.6.1.2.1.7.1)
 |-- udpNoPorts (2)       (1.3.6.1.2.1.7.2)
 |-- udpInErrors (3)      (1.3.6.1.2.1.7.3)
 |-- udpOutDatagram (4)   (1.3.6.1.2.1.7.4)
```

Network Maintenance Commands C

The following commands are useful for monitoring and maintaining the network:

`arp` Displays and modifies the Internet-to-Ethernet address translation tables that the Address Resolution Protocol uses. See the `arp`(4p) and `arp`(8c) reference pages for more information.

`ifconfig` Assigns an address to a network interface or configures the network interface parameters. You can use `ifconfig` to redefine an interface's address, but you should not change the addresses while the system is running. See the `ifconfig`(8c) reference page for more information.

`netstat` Displays the contents of various network-related data structures. If no options are specified, `netstat` displays the state of all active sockets from those using any of the protocols listed in the `/etc/protocols` file. See the `netstat`(1) reference page for more information.

`ping` Uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. Typing **`ping`** *hostname* without any options either reports *hostname* `is alive` or `no answer from` *hostname*. See the `ping`(1) reference page for more information.

`ruptime` Reports the machine name, status, how long it has been up (days+hours:minutes), the number of current users, and the average number of jobs in the run queue in the last 1, 5, and 15 minute periods. If you designate a particular machine, the `ruptime` command prints the information for just that machine.

The following example shows how to use the `ruptime` command to inquire about `host1`, and a typical response:

```
% ruptime host1

host1   up  7+17:49, 3 users,  load  0.41,  0.25,  0.00
```

The `ruptime` command requires that `rwhod` be running on both the machine from which the command is issued and the remote machine in question.

Index

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Introduction to Networking and
Distributed System Services
AA-ME97B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What would you like to see more/less of? ______________________________

____

____

What do you like best about this manual? ______________________________

____

____

What do you like least about this manual? _____________________________

____

____

Please list errors you have found in this manual:

Page   Description

_______ ____

_______ ____

_______ ____

_______ ____

_______ ____

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? ____________

Name/Title ______________________________________ Dept. ____________

Company ______________________________________ Date ____________

Mailing Address ____

____________________ Email ____________________ Phone ____________

------ Do Not Tear – Fold Here and Tape --

digital™

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

------- Do Not Tear – Fold Here --

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Introduction to Networking and
Distributed System Services
AA-ME97B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What would you like to see more/less of? ______________________________

____

____

What do you like best about this manual? ______________________________

____

____

What do you like least about this manual? ______________________________

____

____

Please list errors you have found in this manual:

Page   Description

_______   ____

_______   ____

_______   ____

_______   ____

_______   ____

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? ____________

Name/Title ______________________________ Dept. __________

Company ______________________________ Date __________

Mailing Address ____

______________________ Email ______________ Phone __________

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line