DIGITAL TCP/IP Services for OpenVMS

Concepts and Planning

Order Number: AA-Q06TD-TE

July 1999

This manual describes concepts and planning tasks to prepare you to use the DIGITAL TCP/IP Services for OpenVMS product.

Revision Information: This is a revised manual.

Operating Systems: OpenVMS Alpha Versions 7.1, 7.2
OpenVMS VAX Versions 7.1,7.2

Software Version: DIGITAL TCP/IP Services for
OpenVMS Version 5.0A

Compaq Computer Corporation
Houston, Texas

July 1999

ZK6523

This document is available on CD-ROM.

Contents

3 BIND Service Concepts

4 Network File System Concepts

5 Planning For Your TCP/IP Environment

Tables

Preface

An open communications standard defined by the worldwide networking community, TCP/IP consists of numerous application, routing, transport, and network management protocols. These protocols enable any connected host to communicate with any other connected host, without needing to know details about the other host or the intervening network topology. Computers and networks from different manufacturers running different operating systems can interoperate seamlessly.

The DIGITAL TCP/IP Services for OpenVMS product is Compaq's implementation of the TCP/IP networking protocol suite and internet services for OpenVMS Alpha and OpenVMS VAX systems.

This manual introduces the TCP/IP Services product and provides conceptual and planning information to help you configure and manage the product.

Intended Audience

This manual is for anyone who needs an overview of the TCP/IP Services product.

See the *DIGITAL TCP/IP Services for OpenVMS User's Guide* for information on using TCP/IP Services applications and the *DIGITAL TCP/IP Services for OpenVMS Management* guide for details on configuring and managing the TCP/IP Services product.

Document Structure

This manual contains the following chapters, appendixes, and a glossary.

- Chapter 1 provides an overview of the TCP/IP Services product.
- Chapter 2 introduces TCP/IP networking concepts.
- Chapter 3 describes the TCP/IP Services implementation of the Berkeley Internet Name Domain (BIND) service.
- Chapter 4 describes the Network File System (NFS) and the differences between OpenVMS and the UNIX style file systems.
- Chapter 5 describes general planning issues to consider before configuring your system to use DIGITAL TCP/IP Services for OpenVMS.
- Appendix A describes how to register your network with a network registry and your domain and name servers with your parent-top-level domain.
- Appendix B lists the RFCs associated with the TCP/IP Services for OpenVMS implementation.
- Appendix C provides worksheets for use when configuring your system.
- The Glossary provides a glossary of terms.

Guide to Documentation

Table 1 lists the the documents available to you with this version of DIGITAL TCP/IP Services for OpenVMS.

Table 1 DIGITAL TCP/IP Services for OpenVMS Documentation

Manual	Contents
DIGITAL TCP/IP Services for OpenVMS Concepts and Planning	This manual introduces the TCP/IP Services product and provides conceptual and planning information to help you configure and manage the product. This manual also provides a glossary of terms and acronyms, lists the RFCs associated with this product, and documents how to register your network and domain and name servers.
DIGITAL TCP/IP Services for OpenVMS Release Notes	This text file describes new features and changes to the software including installation, upgrade, configuration, and compatibility information. These notes also describe new and existing software problems and restrictions, and software and documentation corrections. Print this text file at the beginning of the installation procedure and read it before you install DIGITAL TCP/IP Services for OpenVMS.
DIGITAL TCP/IP Services for OpenVMS Installation and Configuration	This manual explains how to install and configure the DIGITAL TCP/IP Services for OpenVMS layered application product.
DIGITAL TCP/IP Services for OpenVMS User's Guide	This manual describes how to use the applications available with DIGITAL TCP/IP Services for OpenVMS such as remote file operations, E-mail, TELNET, TN3270, and network printing. This manual also explains how to use these services to communicate with systems on private internets or on the worldwide Internet.
DIGITAL TCP/IP Services for OpenVMS Management	This manual describes how to configure and manage the DIGITAL TCP/IP Services for OpenVMS product. Use this manual with the *DIGITAL TCP/IP Services for OpenVMS Management Command Reference* manual.
DIGITAL TCP/IP Services for OpenVMS Management Command Reference	This manual describes the DIGITAL TCP/IP Services for OpenVMS management commands. Use this manual with the *DIGITAL TCP/IP Services for OpenVMS Management* manual.
DIGITAL TCP/IP Services for OpenVMS ONC RPC Programming	This manual presents an overview of high-level programming using open network computing remote procedure calls (ONC RPC). This manual also describes the RPC programming interface and how to use the RPCGEN protocol compiler to create applications.
DIGITAL TCP/IP Services for OpenVMS System Services and C Socket Programming	This manual describes how to use the OpenVMS system services and C Socket programming interfaces to develop network-based applications.
DIGITAL TCP/IP Services for OpenVMS eSNMP Programming and Reference	This manual describes the Extensible Simple Network Management Protocol (eSNMP), the eSNMP application programming interface (API), and how to build additional subagents to manage vendor-specific equipment.

For additional information about the DIGITAL TCP/IP Services for OpenVMS products and services, access the DIGITAL OpenVMS World Wide Web site at the following URL:

`http://www.openvms.digital.com`

if you are looking for a comprehensive overview of the TCP/IP protocol suite, you might find the following useful:

- Comer, Douglas E. *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*. Third edition. Englewood Cliffs, NJ: Prentice-Hall Inc., 1995.
- Stevens, W. Richard. *UNIX Network Programming Volume 1: Networking APIs: Sockets and XTI*. Second edition. Englewood Cliffs, NJ: Prentice-Hall Inc., 1999.
- Stevens, W. Richard. *UNIX Network Programming Volume 2: Interprocess Communications*. Second edition. Englewood Cliffs, NJ: Prentice-Hall Inc., 1999.

Terminology

DIGITAL TCP/IP Services for OpenVMS Version 5.0 completes the change initiated several releases ago when the product name changed from "ULTRIX Connection (UCX)" to "DIGITAL TCP/IP Services for OpenVMS." To complete this change, the identifier "UCX" is replaced with "TCPIP" in the following:

- Registered product facility code
- Management command prompt
- All messages, examples, and banners
- All product file names, queues, and databases
- All logical names, except those retained for compatibility
- All associated product documentation

DIGITAL TCP/IP Services for OpenVMS is used to mean both:

- DIGITAL TCP/IP Services for OpenVMS Alpha
- DIGITAL TCP/IP Services for OpenVMS VAX

The auxiliary server is the DIGITAL TCP/IP Services for OpenVMS implementation of the UNIX internet daemon (`inetd`).

NFS is the DIGITAL TCP/IP Services for OpenVMS implementation of the NFS protocols, including the NFS server, the NFS client, and PC-NFS.

TN3270 is the TELNET client software that emulates IBM 3270 model terminals.

The term UNIX refers to DIGITAL UNIX operating system. DIGITAL UNIX is fully compatible with Version 4.3 and Version 4.4 of the Berkeley Software Distribution (BSD).

Host and **node** both mean a system connected to an internet.

The term **Internet** refers to the global interconnection of networks, as defined by RFC 1208, which consists of large networks using TCP/IP to provide universal connectivity, reaching the Defense Advanced Projects Research Internet, MILNET, NSFnet, CERN, and many worldwide universities, government research labs, military installations, and business enterprises.

The term **intranet** refers to private interconnected networks that use TCP/IP to connect together and function as one virtual network.

Acronyms

For a complete list of acronyms used throughout this and other manuals in the DIGITAL TCP/IP Services for OpenVMS documentation set, see the glossary in this manual.

Reader's Comments

Compaq welcomes your comments on this manual.

Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT and send us your comments by:

Internet	**openvmsdoc@compaq.com**
Fax	603 884-0120, Attention: OSSG Documentation, ZKO3-4/U08
Mail	Compaq Computer Corporation OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following World Wide Web address for information about how to order additional documentation:

`http://www.compaq.com/openvms`

To reach the OpenVMS documentation website, click the Documentation link.

If you need help deciding which documentation best meets your needs, call 1–800–ATCOMPA.

Conventions

All IP addresses in this manual represent fictitious addresses. The following conventions apply to this manual.

Convention	Meaning
UPPERCASE TEXT	Indicates names of OpenVMS and TCP/IP Services commands, options, utilities, files, directories, hosts, and users.
`lowercase special type`	Indicates UNIX system output or user input, commands, options, files, directories, utilities, hosts, and users.
bold type	Indicates a new term.
italic type	Indicates a variable.
Return	Indicates that you press the Return or Enter key.
Ctrl/*x*	Indicates that you press the Control key while you press the key noted by *x*.
[ ]	In command format descriptions, indicates optional elements. The elements are separated by vertical bars (|). You can enter as many as you want.

Convention	Meaning
{ }	In command format descriptions, indicates you must enter at least one listed element. The elements are separated by vertical bars (|).
. . .	Horizontal ellipsis points in examples indicate additional optional arguments have been omitted.
. . .	Vertical ellipsis points indicate omission of items from a code example or display example; the items are omitted because they are not important to the topic being discussed.

1

Introduction to DIGITAL TCP/IP Services for OpenVMS

The DIGITAL TCP/IP Services for OpenVMS product is the OpenVMS implementation of the industry-standard TCP/IP suite of communications protocols. With TCP/IP, heterogeneous networks can interconnect, making it possible for users to connect to remote hosts in many ways:

- Network file access — users can access files on remote hosts.
- Electronic mail — users can exchange messages between hosts.
- Application development — application programmers can develop TCP/IP applications for communication between local and remote hosts.
- Download and file transfer — users can exchange files between hosts.
- User information — users can access information about other users logged onto the local or remote host.
- Remote management — system managers can monitor the network and applications from remote hosts.
- Remote terminal access — users can access a remote host as if their terminal is connected directly to that host.
- Remote command execution — users can issue commands to remote hosts.
- Remote printing — users can send or receive print jobs to or from remote printers.
- Remote file copy — users can copy files that reside on remote hosts.
- Remote booting — users can provide boot information for remote hosts.

Internetworking with TCP/IP hides the hardware details of each individual network and allows computers to communicate independently of their physical network connections. TCP/IP provides both a standard transport mechanism and full-duplex, reliable, stream communication services for software applications.

The DIGITAL TCP/IP Services for OpenVMS product provides interoperability and resource sharing between OpenVMS systems, UNIX systems, and other systems that support the TCP/IP protocol suite and Sun Microsystems' Network File System (NFS). TCP/IP systems and other internet hosts share data and resources by using standard TCP/IP protocols over a number of network hardware configurations: Ethernet, Fiber Distributed Data Interface (FDDI), Token Ring, and asynchronous transfer mode (ATM).

This chapter discusses the following:

- TCP/IP Defined: Requests for Comments (Section 1.1)
- TCP/IP Architecture (Section 1.2)

- Data Link Layer (Section 1.3)
- Internet Layer Protocols (Section 1.4)
- Transport Layer Protocols (Section 1.5)
- Application Layer (Section 1.6)
- Management Tools and Utilities (Section 1.7)
- Application Programming Environment (Section 1.8)
- Application Support (Section 1.9)

1.1 TCP/IP Defined: Requests for Comments

TCP/IP evolved from the U.S. Government's need to connect many different networks regardless of their hardware architecture, operating system, or subnetwork technology. The resulting **internetwork** needed to be able to route data between networks, tolerate routing errors, and easily add new subnetworks. From a simple four-host entity in 1969 to today's worldwide Internet connecting thousands of networks and millions of computers, TCP/IP has become the communications standard of the Internet.

TCP/IP is an open system interconnection. Although monitored by a number of organizations, no one entity owns TCP/IP; its specifications are publicly available and constantly growing as communications requirements evolve.

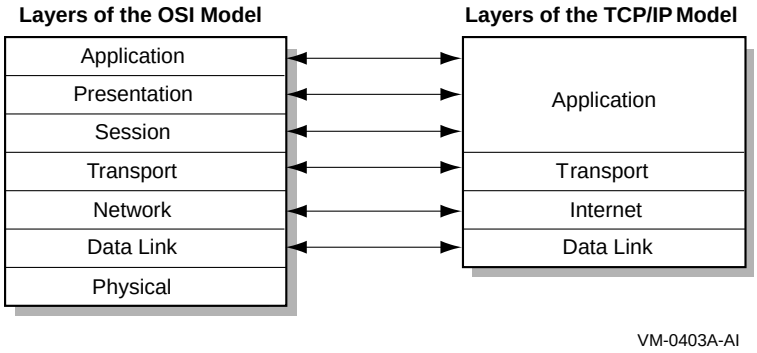
The process by which the specifications evolve is through a mechanism called **Requests for Comments** or, more commonly, **RFCs**. Basically, when someone has an idea for a new or improved capability for TCP/IP, he or she writes a proposal, posts it on the Internet as an Internet draft, and requests comments from the networking community. After a review and revision cycle, working code is developed and an RFC becomes a standard protocol.

RFCs are available on the Internet from an organization called the Internet Network Information Center, or InterNIC. Appendix B lists relative RFCs and explains how you can obtain copies of RFCs.

1.2 TCP/IP Architecture

The TCP/IP protocol suite is designed in a fashion similar to that of the OSI layered model. However, the TCP/IP protocol suite has four layers while the OSI model has seven layers. Figure 1–1 shows the relationship between the layers of the two models.

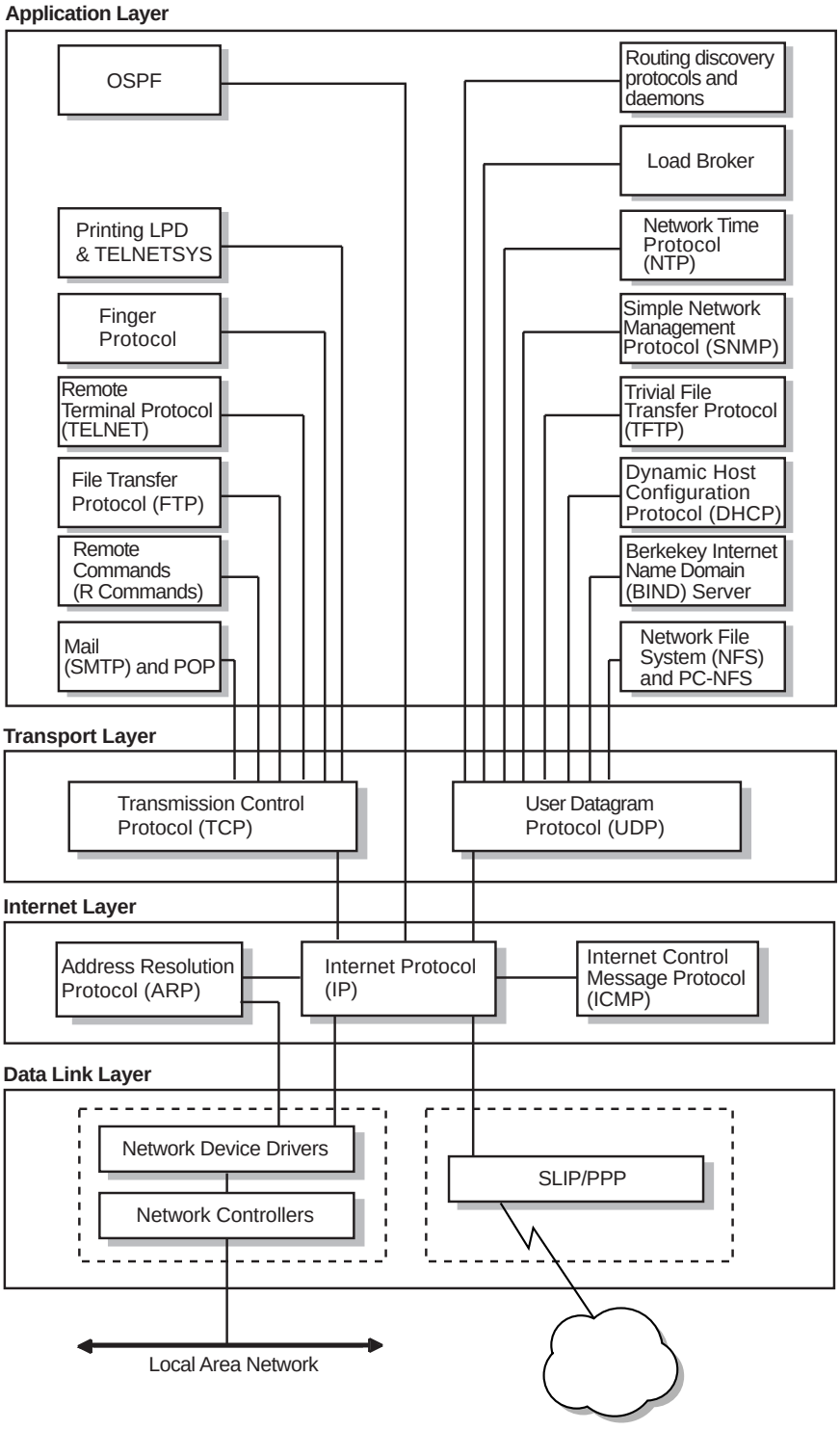
Figure 1–1 Relationship Between TCP/IP and OSI Models



As shown in the illustration, the OSI model's Session and Presentation layer functions are fulfilled by the TCP/IP Application layer protocols. Likewise, some of the functions of the OSI Physical layer are handled by the Network Interface layer and the hardware itself in the TCP/IP model.

Figure 1–2 and Table 1–1 outline the layers of the TCP/IP model. Sections 1.4 through 1.6 summarize the protocols.

Figure 1–2 DIGITAL TCP/IP Protocol Architecture



VM-0402A-AI

Table 1–1 TCP/IP Network Architecture Description

Layer	Function
Data Link	Transmits data across a single network. This layer also receives data routed from the Internet layer and transmits the data to its destination.
Internet	Moves data around the internetwork. The Internet Protocol routes packets across networks independently of the network medium. It also encapsulates datagram headers, sends ICMP error and control messages, and maps ARP address conversions.
Transport	Provides a flow of data between two hosts. The DIGITAL TCP/IP Services for OpenVMS product supports the two common transport protocols: • Transmission Control Protocol (TCP) provides a reliable data flow between two hosts. • User Datagram Protocol (UDP) provides a much simpler service to the Application layer than TCP but does not guarantee reliability.
Application	Handles the details of the particular application, protocol, or user command; not concerned with the movement of data across the network. The product supports the following TCP/IP applications, protocols, and user commands: *Remote Computing* • TELNET for remote login to other hosts in the network. • Remote commands: RLOGIN for remote login, RSH for remote shell capabilities, REXEC to execute commands to a remote host, and RMT/RCD to read magnetic tapes or CD-ROMs from remote hosts. • Finger utility to display user information. *File Transfer* • File Transfer Protocol (FTP) to transfer files between hosts. • Trivial File Transfer Protocol (TFTP) to download and transfer files. *Resource Sharing* • Line printer/line printer daemon (LPR/LPD) to provide printing services to local and remote hosts. • TELNET Print Symbiont (TELNETSYM) to provide remote printing using the TELNET protocol. • Network File System (NFS) and PC-NFS to authenticate requests and access remote files. *Electronic Mail* • Simple Mail Transfer Protocol (SMTP) for electronic mail. • Post Office Protocol (POP) for electronic mail for PC users.

Table 1–1 (Cont.) TCP/IP Network Architecture Description

Layer	Function
	Network Services • Simple Network Management Protocol (SNMP) to monitor and manage any network device running SNMP software across an internetwork. • Network Time Protocol (NTP) to synchronize time between hosts in a TCP/IP network. • Berkeley Internet Name Domain (BIND), a distributed database system, to distribute and manage host information so that hosts do not need to know the address of every other host on the internet. • The Bootstrap Protocol (BOOTP) to answer bootstrap requests from remote devices. • Dynamic Host Configuration Protocol (DHCP), a superset of BOOTP, to assign temporary or permanent IP addresses, subnet masks, and default gateways for both BOOTP and DHCP clients. Allows for the management of network connections from a single location through a graphical user interface (GUI).

1.3 Data Link Layer

The Data Link layer of the TCP/IP model (sometimes called the network interface layer) is responsible for properly sending and receiving communications signals between two communicating hosts through their network interfaces. This layer includes the device driver and the corresponding network interface.

The device driver, also called the **network interface**, is a software component that communicates with the TCP/IP software and the network interface card (the hardware connection between a computer system and a network).

Individual host computers can have multiple network interface cards per computer. Such a computer is called **multihomed**. These physical interfaces may be connected to different types of networks such as Ethernet, FDDI, Token Ring, and asynchronous transfer mode (ATM), Gigabit Ethernet, and serial communications lines. Each physical interface is associated with one device driver (network interface). A single network interface can have more than one IP address.

The TCP/IP Services product provides a pseudo-device driver that handles the communication between the network interfaces and the TCP/IP Services software. This pseudo-device driver is called the BG driver. There may be several network interfaces, depending on the number of network interface cards you have installed on your computer, but only one BG driver.

With TCP/IP — as with any layered networking protocol — each layer adds header information to the data from the layer above. Each packet contains a header from the Network Interface layer, followed by a header from the Internet layer, followed by a header from the Transport layer, followed by the application data.

At the Network Interface layer, standard encapsulation of IP packets are defined for the various hardware types. Ethernet, for example, uses the Ethernet frame standard to enclose the data being sent with header fields. Serial line connections use either Serial Line Internet Protocol (SLIP or CSLIP) or Point-to-Point Protocol (PPP).

Serial Line Internet Protocol

SLIP is a simple packet framing protocol. It defines a sequence of characters that frame IP packets on a serial line. It provides no mechanisms for addressing, packet type identification, error detection/correction, or compression. Although limited in scope, SLIP is easy to implement, but transmission speeds are relatively slow. CSLIP (Compressed SLIP) allows for faster transmission by compressing the TCP/IP headers.

For more information on SLIP and CSLIP, see the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Point-to-Point Protocol

PPP is more complex than SLIP and CSLIP, but it offers much greater functionality. As described in RFC 1331, PPP consists of three main components:

- A method for encapsulating datagrams over serial links.
- A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
- A family of Network Control Protocols (NCPs) for establishing and configuring different Network Interface layer protocols.

For more information on PPP, see the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

1.4 Internet Layer Protocols

The Internet layer (sometimes called the Network layer) provides a connectionless packet delivery service using IP (Internet Protocol). An IP datagram is a packet that has no delivery receipt and is called **connectionless** because IP does not maintain state information about successive datagrams. Each datagram is handled independently from all other datagrams.

Table 1–2 describes the Internet layer protocols available in the TCP/IP Services product.

Table 1–2 Internet Layer Protocols

Protocol	Description
Internet Protocol (IP)	IP sends or routes data across the network from its source to its destination by means of internet addressing (an IP address). The IP address identifies the connection between the network controller of a host and the network cable. IP then receives data bits from the network hardware, assembles the bits into an IP datagram, and chooses the best route to send the packet to its destination. IP also fragments and reassembles packets during the routing process.

Table 1–2 (Cont.) Internet Layer Protocols

Protocol	Description
Internet Control Message Protocol (ICMP)	ICMP provides a number of diagnostic functions and handles error and control messages. ICMP reports problems with data delivery to gateways and hosts.
Address Resolution Protocol (ARP)	ARP dynamically maps an IP address to a physical hardware address of the broadcast medium such as Ethernet, FDDI, and Token Ring. ARP is limited to a single physical network and to networks that support hardware broadcast.

The routing of packets also occurs in the Internet layer. Table 1–3 describes the routing protocols supported in the TCP/IP Services product. These protocols build and update the routing table used by the Internet layer.

Table 1–3 Internet Layer Routing Protocols

Protocol	Description
Routing Information Protocol (RIP) Versions 1 and 2	RIP is a commonly used interior protocol that selects the route with the lowest metric (hop count) as the best route.
Open Shortest Path First (OSPF) Version 2	Another interior routing protocol, OSPF is a link-state protocol (shortest path first) and is better suited than RIP for use in complex networks with many routers.
Exterior Gateway Protocol (EGP)	EGP exchanges reachability information between autonomous systems. An autonomous system is usually defined as a set of routers under a single administration, using an interior gateway protocol and common metric to route packets. Autonomous systems use exterior routing protocols to route packets to other autonomous systems.
Border Gateway Protocol (BGP)	Like EGP, BGP exchanges reachability information between autonomous systems but supports nonhierarchical topologies. BGP uses path attributes to provide more information about each route. Path attributes can include, for example, administrative preferences based on political, organizational, or security considerations.
Router Discovery	This protocol is used to inform hosts of the availability of other hosts it can send packets to, and to supplement a statically configured default router.

1.5 Transport Layer Protocols

The Transport layer protocols provide either connection-oriented or connectionless data transmission from one host to another. The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) form the bridge between the Application layer functions and Internet layer protocols such as IP.

The DIGITAL TCP/IP Services for OpenVMS product supports both TCP and UDP.

Transmission Control Protocol

TCP provides connection-oriented, reliable, sequenced data transfers between local or remote hosts. Before transmitting data, participants must establish a connection. TCP guarantees that data reaches its destination, and retransmits any data that does not get through.

User Datagram Protocol

UDP provides connectionless data transfers between local and remote hosts. It allows application programs to send and receive datagrams to applications that reside on a different host. UDP adds a checksum and addressing information, then uses IP to deliver the datagrams. UDP provides fast communications for applications that do not require delivery receipts at the Transport layer.

1.6 Application Layer

At the Application layer, data is received as commands from the user application at the other end of the connection. TCP/IP applications often communicate in client/server pairs at this layer.

Because of the large number of Application layer protocols supported by the DIGITAL TCP/IP for OpenVMS product, it is useful to group them in five general categories:

- Remote computing
- File transfer
- Resource sharing
- Electronic mail
- Network services

Sections 1.6.1 through 1.6.5 include summaries of each of the primary Application level components included in the product. See the *DIGITAL TCP/IP Services for OpenVMS User's Guide* for detailed user information about remote computing, file transfer, resource sharing, and electronic mail components. Refer to the the *DIGITAL TCP/IP Services for OpenVMS Management* guide for information about network services and tools.

1.6.1 Remote Computing

Remote computing applications enable networked users to run software on remote systems. The product includes the following remote computing application components:

- TELNET provides remote login to other hosts in the network.
- RLOGIN command provides remote login.
- RSH command provides remote shell capabilities.
- REXEC command allows users to execute commands on a remote host.
- RMT and RCD commands allow remote users to access tape or CD-ROM drives.
- Finger utility displays user information.

TELNET

TELNET is a standard protocol that provides remote terminal connection or login service. TELNET enables users at one site to interact with a remote system at another site, as if the user terminals were connected directly to the remote system.

The DIGITAL TCP/IP Services for OpenVMS product implements TELNET to provide:

- Simultaneous multiple sessions
- IBM 3270 terminal emulation (TN3270)
- Two supported interface formats:

  DCL-style
  UNIX style

Remote Commands

The remote commands, called R commands, enable users to work in accounts on remote internet hosts that also run an implementation of TCP/IP and the R commands. The DIGITAL TCP/IP Services for OpenVMS software supports the RLOGIN, RSH, REXEC, and RMT/RCD commands, as explained below. Users issue these commands at the system command line prompt ($).

Remote Command	Function
RLOGIN	Allows users of one system to log into other systems across an internet and interact as if they were directly connected. RLOGIN offers the same service as TELNET, except it also passes information about the user's environment, such as the user identification, to the remote host.
RSH	Allows users to send a command, shell script, or command procedure to a remote host for execution. RSH does not require login to the remote host to execute commands.
REXEC	Based on password information stored in the user authentication file (UAF), authenticates the users and then executes the R commands the user issues.
RMT/RCD	Allows remote users to access magnetic tape and CD-ROM drives.

For detailed information about the RLOGIN, RSH, and REXEC commands, see the *DIGITAL TCP/IP Services for OpenVMS User's Guide*. For information about RMT/RCD, see the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Finger Utility

The Finger utility is implemented by the FINGER command, which displays information about users on the local or remote host. By default, information about each user on the host is displayed.

When you specify a user name or a list of user names with the FINGER command, the Finger utility returns the following information:

- Full name
- Account name
- Program the user is running
- User's home directory

- Any plan in the file named `.PLAN` in the user's home directory
- The project on which the user is working from the file named `.PROJECT` in the user's home directory

If you do not specify a host, the information listed is about users on the local host; otherwise, the information is about users at the specified host. You can specify a user on a remote host by using the form *user@host*. If you specify *@host*, the standard format listing is displayed on the remote host. The specified host must have the Finger service enabled.

1.6.2 File Transfer

The DIGITAL TCP/IP Services for OpenVMS product includes the following file transfer components that let users transfer data files between local and remote hosts:

- FTP (File Transfer Protocol) transfers files between hosts.
- Trivial File Transfer Protocol (TFTP) downloads and transfers files.
- The remote copy (`rcp`) command copies files to or from remote hosts.

File Transfer Protocol

FTP is a TCP/IP standard, high-level protocol used to transfer files bi-directionally. FTP enables users to interactively access files, list directories on a remote host, delete and rename files on the remote host, and transfer files between hosts.

FTP also provides authentication control, which requires users or clients to correctly enter a login name and password to the server before requesting file transfers. The server can refuse access due to invalid login and password combinations.

FTP allows users who do not have a login name or a password to access certain files on a system using an anonymous login name. This functionality is called **anonymous FTP** and may include the following restrictions:

- Limited browsing through the file system. Users may have access only to the anonymous guest, or home directory and a public directory. The public directory may contain general bulletin information to which the user has read-only access.
- Accessing files from (`get`) or copying files to (`put`) the guest directory only.
- Accessing files (`get`) from the public directory only.
- Deleting privileges for files in the guest directory that are owned by the anonymous account.

Trivial File Transfer Protocol

TFTP provides a simple, unsophisticated file transfer service. It is intended for applications that do not need complex interactions between a client and server. TFTP is small and can be hardcoded in read-only memory to execute a network bootstrap program. TFTP allows the bootstrap program to use the same underlying protocols that the operating system uses once it begins execution. This makes it possible for one host to boot from a server on another physical network.

The DIGITAL TCP/IP Services for OpenVMS product supports downloading of system images and other types of information for client hosts with TFTP.

1.6.3 Resource Sharing

Resource sharing lets users access remote system resources such as disk storage space or printers as if they were directly connected to the user's local systems. With resource sharing, users can access these resources directly after the initial connection is made. This is different from file transfer programs where files must be completely transferred from the remote system before they can be used.

The resource-sharing components of DIGITAL TCP/IP Services for OpenVMS include:

- Line printer/line printer daemon (LPR/LPD), which provides print services to remote and local hosts.
- TELNET Print Symbiont (TELNETSYM), which provides for remote printing using the TELNET protocol.
- Network File System (NFS) and PC-NFS, which authenticate requests and provide access to remote files.

Line Printer Protocol and Line Printer Daemon Protocol

The DIGITAL TCP/IP Services for OpenVMS software provides network printing through LPR/LPD. LPD provides remote printing services for UNIX and OpenVMS client hosts. Each print queue is either local or remote. Local print queues handle inbound jobs. Remote print queues handle outbound jobs for remote printers.

The print setup utility (TCPIP$LPRSETUP) provides the following capabilities:

- Updates the related printcap database
- Creates and starts queues
- Allows you to add commands to the automatic startup and shutdown command procedures

To print, users at an OpenVMS client issue the DCL PRINT command.

Users working on UNIX clients typically issue the `lpr` command.

TELNET Print Symbiont

The TELNET print symbiont (TELNETSYM) provides remote printing using the TELNET protocol. With TELNETSYM, the local OpenVMS system drives a remote printer as if it were directly connected. This is achieved by attaching a printer to a remote TCP/IP terminal server.

The TELNET print symbiont has the following functions:

- Transfers record-oriented data to and from disks and printers
- Configures printers attached to terminal servers that support TELNET
- Supports outbound functions (to a remote printer) and offers preformatting to outbound print jobs

Note

TELNET does not work with terminal servers that use only the local area transport (LAT) protocol. The terminal server must support TCP/IP.

The system that originates the print jobs handles the standard print control functions, such as header page generation, pagination, queuing, and handling of multiple forms.

TELNET printing allows the standard OpenVMS output format features not available with the LPR/LPD service. The /ON=TCPIP$QUEUE=*name* qualifier to the DCL INITIALIZE/QUEUE command allows you to requeue a print job to another local OpenVMS queue after formatting, rather than using TELNET to send the print job across the network.

For detailed information on configuring and managing TELNETSYM, see the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Network File System

NFS is a protocol developed by Sun Microsystems, Inc., that allows computers to access remote files as if they were local files. With NFS, when you access files and directories from a remote system, they appear to reside on your local system regardless of operating system, hardware type, or architectural differences between the local and remote systems.

This is accomplished in a client/server environment where specific implementations of NFS exist on both the client and the server machines. In its simplest form, here is how it works: When an application program executes, it calls the operating system to open a file. If the file does not reside on the local system, the request is passed to the NFS client, which knows how to contact the NFS server on the remote machine. The remote machine then performs the requested operation and replies to the client software.

In addition to NFS server and NFS client software, the TCP/IP Services product includes a PC-NFS daemon that allows PC-NFS clients access to a TCP/IP Services NFS server. The NFS server, NFS client, and PC-NFS are summarized in Table 1–4.

Table 1–4 NFS Components

NFS Component	Function
NFS server	Enables remote users to access files that physically reside on an OpenVMS system running DIGITAL TCP/IP Services for OpenVMS software. The remote host must support an NFS client. UNIX supports only sequential, byte-stream file formats. OpenVMS supports index, relative, and sequential files that have various record formats. The DIGITAL TCP/IP Services for OpenVMS implementation of NFS allows access to the following record formats: • Fixed length • Variable length • Variable with fixed-length control (VFC) • Stream (including STREAM_LF and STREAM_CR) • Undefined

Table 1–4 (Cont.) NFS Components

NFS Component	Function
NFS client	Accepts file operation requests from the operating system and contacts the appropriate NFS server on the remote system to perform the requested operation. Supports STREAM_LF formatted files.
PC-NFS daemon	Enables access to the NFS server from a personal computer by providing authentication services to PC-NFS clients. The PC-NFS daemon software provides: • Authentication Network security in the NFS model is accomplished by identifying every user with a user identification (UID) and group identification (GID) pair. Personal computers, lacking this information, must request their UID and GID from a remote server. They obtain this information from the proxy database. With this information, PC users can then mount remote NFS file systems. • Printing From your PC, you can: – Associate a DOS or Windows printer name with an OpenVMS print queue – Print a file to the associated queue From the OpenVMS server, you can: – List queued jobs – Cancel queued jobs – Obtain a list of available print queues – Obtain the status of a particular print queue On the OpenVMS system, you can log into a privileged account or to the same account your PC uses as its proxy account.

1.6.4 Electronic Mail

Communication functions such as electronic mail are vital both within an organizational internetwork and across the worldwide Internet. The electronic mail components of DIGITAL TCP/IP Services for OpenVMS are:

- Simple Mail Transfer Protocol (SMTP) for electronic mail
- Post Office Protocol (POP) to store electronic mail for PC users

Simple Mail Transfer Protocol

SMTP is the TCP/IP standard protocol for transferring electronic mail messages from one system to another. SMTP specifies how systems interact and the format of the mail messages they exchange. The product's SMTP implementation uses the OpenVMS mail facility.

The OpenVMS mail facility automatically recognizes an SMTP host address, as shown in the following example:

```
$ MAIL
MAIL> SEND
To:    jones@widgets.com
```

Post Office Protocol (POP)

The DIGITAL TCP/IP Services for OpenVMS Post Office Protocol (POP) server and the SMTP server work together to provide a reliable mail service.

POP is a mail repository used primarily by PCs to ensure that mail is accepted even when the PC is turned off. With POP, the PC user need not be concerned with configuring the system as an SMTP server. The user logs into the client system's mail application, and the POP server forwards any new mail messages from the OpenVMS NEWMAIL folder.

The POP server is an OpenVMS implementation of the Post Office Protocol, Version 3 (RFC 1725), and is based on the Indiana University POP server (IUPOP3).

1.6.5 Network Services

The DIGITAL TCP/IP Services for OpenVMS product provides services that are used by system or network managers rather than directly by users. The following TCP/IP Services components let the system or network manager provide consistent, reliable, and efficient network services with minimal interruption:

- Simple Network Management Protocol (SNMP) that allows your host to answer requests from an SNMP client management station.
- Network Time Protocol (NTP) to synchronize time between hosts in a TCP/IP network.
- Berkeley Internet Name Domain (BIND), a name and address resolution service to distribute and manage host information so that hosts do not need to know the address of every other host on the internet.
- Portmapper service to enable client programs to determine port numbers that another service uses.
- Auxiliary server to simplify application design and manage overhead.
- Dynamic Host Configuration Protocol (DHCP) to configure BOOTP and DHCP clients from a single location through the assignment of temporary or permanent IP addresses, subnet masks, and default gateways.

Simple Network Management Protocol (SNMP)

SNMP is a vendor-independent network management standard for managing network components in a TCP/IP network.

The TCP/IP Services product implements SNMP Version 2c using an extensible SNMP (eSNMP) architecture. The product provides a master agent and two subagents to implement data items from standard Management Information Bases (MIBs): MIB-II (RFC 1213) and the Host Resources MIB (RFC 1514). Depending on how you configure your host, an SNMP management station can obtain information about your host and perform updates on your host's MIB data items. For SNMP configuration, operation and restriction information, see the *DIGITAL TCP/IP Services for OpenVMS Management* manual, *DIGITAL TCP/IP Services for OpenVMS Management Command Reference* manual and *DIGITAL TCP/IP Services for OpenVMS Release Notes*.

The TCP/IP Services product also includes an eSNMP application programming interface (API) and related tools. These features help customers create custom subagents to access their own software or hardware parameters through SNMP. For more information about the eSNMP API, see the *DIGITAL TCP/IP Services*

for OpenVMS Release Notes and the *DIGITAL TCP/IP Services for OpenVMS eSNMP Programming and Reference* manual.

Network Time Protocol

The Network Time Protocol (NTP) provides a means to synchronize time and coordinate time distribution throughout a TCP/IP network. Time synchronization is important in client/server computing. For example, systems that share common databases require coordinated transaction processing and time-stamping of instrumental data. Synchronized timekeeping means that hosts with accurate system time send accurate time quotes to each other. Hosts running NTP act as time servers, clients, or both server and client.

The TCP/IP Services product implements NTP Version 3 (xNTP) and provides several utility programs that help you manage and make changes to the NTP server. These utilities include:

- TCPIP$NTPDATE, the date and time utility that sets the local date and time by polling the specified server. Run NTPDATE manually or from the host startup script to set the clock at boot time before NTP starts. NTPDATE will not set the date if NTP is already running on the same host.
- TCPIP$NTPTRACE, the trace utility that follows the chain of NTP servers back to their master time source.
- TCPIP$NTPDC, the special query program that provides extensive state and statistics information and that can be used to set configuration options at run time. Run this program in interactive mode or with command line arguments.
- TCPIP$NTPQ, the standard query program that queries NTP servers about their current state and requests changes to that state.

See the *DIGITAL TCP/IP Services for OpenVMS Management* manual for more information about configuring and managing NTP.

Domain Name Service (DNS)

The Domain Name Service (DNS) is an Internet service that maintains and distributes information about Internet hosts. DNS consists of several databases that store host names and host IP addresses. With DNS, there is no central storage of data — no one server knows everything about all the Internet domains. In UNIX environments, DNS is implemented by the Berkeley Internet Name Domain (BIND) software. The TCP/IP Services product implements a BIND server based on the Internet Software Consortium's (ISC) BIND 8. The BIND 8 implementation provides new configuration syntax and a new format for configuring the BIND name server.

BIND is a lookup service for the Internet. BIND is divided into two components: a resolver and a name server. The resolver queries a name server and the name server responds to a resolver query:

BIND Component	Function
BIND resolver	Client software that requests host names, addresses, and other network information from BIND servers that maintain extensive information, rather than from the more limited local database.

BIND Component	Function
BIND server	Server software that translates host names into numeric Internet addresses and numeric Internet addresses into host names. BIND servers maintain databases of host names, addresses, mail records, text records, and other network objects. When client systems require this information, they query the servers with the BIND resolver.

For more information about BIND and planning your DNS environment, see Chapter 3, Chapter 5, and the *DIGITAL TCP/IP Services for OpenVMS Management* manual.

Portmapper

Internet hosts can simultaneously run multiple industry-standard and custom-developed services. With the portmapper, you do not need to preconfigure client applications with port numbers for each service. Instead, each server registers itself and the portmapper allocates the port. Each server process listens for connections on a designated port.

The portmapper maintains a database of the following:

- Registered server programs
- Unique identifiers for each server program (program number)
- Port numbers for the server program

Remote clients request port numbers to connect to particular applications.

Auxiliary Server

The TCP/IP Services product implements the UNIX internet daemon `inetd` function, through the security and event and error logging of the auxiliary server process. The auxiliary server simplifies application writing and manages overhead by reducing simultaneous server processes on the system. In addition, the auxiliary server does the following:

- Eliminates high overhead due to nonstop running of all service processes
- Uses proxy and service databases to provide system security through authentication of service requests
- Supports event and error logging

Dynamic Host Configuration Protocol

DHCP is an extension (or superset) of BOOTP that allows for the centralized management of network connections. In addition to BOOTP functionality, DHCP provides configuration services including the assignment of temporary or permanent IP addresses, subnet masks, and default gateways for both BOOTP and DHCP clients.

The TCP/IP Services implementation of DHCP allows system managers to configure a host as a DHCP server and specify server characteristics (parameters) and client information through DHCP's graphical user interface (GUI).

System managers who currently use BOOTP to manage their IP address space can easily migrate to a DHCP environment. In addition, system managers can configure a cluster failover environment to ensure that a backup system takes over as the DHCP server if the active DHCP server process should stop for any reason.

For information about configuring your DHCP environment, see Chapter 5.

1.7 Management Tools and Utilities

The Management Control Program is a comprehensive, easy-to-use network management tool that includes over 100 OpenVMS DCL-style commands. These commands allow you to locally configure, monitor, and tune DIGITAL TCP/IP Services for OpenVMS components and to write customized applications by issuing management commands at the TCPIP> prompt.

To invoke the program, enter:

```
$ TCPIP
```

You can also use UNIX management commands to manage some components of the TCP/IP Services product. Table 1–5 lists the supported UNIX commands.

Table 1–5 UNIX Management Commands

Command	Description
ifconfig	Configures or displays network interface parameters, redefines an address for a particular interface, or sets options such as an alias list, broadcast address, or access filter.
netstat	Displays network statistics of sockets, data link counters, specified protocols or aliases, network interfaces, and a host's routing table.
sysconfig	Displays and maintains the network subsystem attributes.
route	Manually manipulates the routing table. Normally a system routing table management daemon, such as GATED or ROUTED, will tend to this task.
arp	Controls and displays ARP tables for the specified host.
ping	Sends ICMP ECHO_REQUEST packets to network host.
traceroute	Prints the route that packets take to the network host.

1.8 Application Programming Environment

The DIGITAL TCP/IP Services for OpenVMS product supports the following application programming interfaces (APIs) for developing customized network applications:

- Berkeley Socket Interface
- OpenVMS QIO System Service Interface
- Sun RPC programming interface
- eSNMP programming interface

1.8.1 Berkeley Socket Interface

The Berkeley socket interface is a programming interface which provides applications with access to network communication protocols. A socket is a generalized, UNIX communication endpoint upon which the TCP/IP protocols have been implemented. Using the socket programming interface, it is easy to implement network applications. Sockets have since become a popular programming interface.

OpenVMS provides support for the socket interface through the C programming language and the DEC C Run-time Library. Benefits of using the socket interface on the OpenVMS platform include:

- Ease of porting network applications using the socket interface from other platforms to the OpenVMS platform
- Application developer familiarity with the programming environment
- Options for other types of protocols in addition to the TCP/IP protocols

Refer to the *DIGITAL TCP/IP Services for OpenVMS System Services and C Socket Programming* manual for more details.

1.8.2 OpenVMS QIO System Service Interface

The standard I/O programming interface on OpenVMS is through the QIO (Queue Input/Output) system services. QIO provides a rich set of functions for controlling devices, connections and for performing input (read) and output (write) operations.

The benefits of using the OpenVMS QIO interface include:

- Support for the QIO interface exists in the following programming languages:

  MACRO-32
  DEC C
  DEC Fortran
  VAX Ada
  VAX BASIC
  VAX BLISS-32
  VAX COBOL
  VAX Pascal
  VAX PL/1

- Ability to handle complex applications with many concurrent connections
- Potentially more efficient input/output operations
- More robust asynchronous event handling

  While sockets do offer the ability to do non-blocking I/O operations, they do not offer the ability to perform asynchronous I/O.

- Ease of portability of DECnet applications to the TCP/IP protocols

Refer to the *DIGITAL TCP/IP Services for OpenVMS System Services and C Socket Programming* manual for more details.

1.8.3 Sun RPC Programming Interface

The RPC programming interface is an industry-standard, portable API that is an efficient alternative to application development with sockets. Programmers using RPC do not need an in-depth knowledge of networking protocols.

One strong point of the RPC interface is the ability to distribute functions across the network. This is done in an architecture independent manner where floating point formats and byte address ordering would normally lead to problems when interacting between architectures.

This API includes:

– Library of RPC function calls

– Portmapper service

– External data representation (XDR) routines

Refer to the *DIGITAL TCP/IP Services for OpenVMS ONC RPC Programming* manual for more details.

1.8.4 eSNMP Programming Interface

The Extensible Simple Network Management Protocol (eSNMP) API provides routines for developing applications which remotely manage and collect data from network devices such as routers, bridges and hosts.

The network devices run software that carry out management commands that either get information from or set operating parameters for the device.

Other network applications send commands to network devices to perform configuration management, monitor network traffic or troubleshoot network problems.

The API provides routines for the following functions:

– Establish, maintain, and terminate communication with the master agent

– Manipulate, reformat, extract, and compare data

– Control information that is written to log files

The eSNMP API routines are almost identical in function and interface with the routines in the Tru64 UNIX API.

Refer to the *DIGITAL TCP/IP Services for OpenVMS eSNMP Programming and Reference* manuals for more details.

1.9 Application Support

TCP/IP Services provides support for the following:

- PATHWORKS and DECnet-Plus Internet Protocol Support
- SRI QIO Compatibility

1.9.1 PATHWORKS and DECnet-Plus Internet Protocol Support

The DIGITAL TCP/IP Services for OpenVMS software includes the the PWIP driver and the PWIPACP network ancillary control process (ACP). The PWIP driver makes possible communication between OpenVMS systems running both PATHWORKS server and TCP/IP Services software, and personal computers running PATHWORKS client software. It also enables the DECnet-over-TCP/IP feature which is included with the DECnet-Plus for OpenVMS Version 6.0 and later software. For more information, see the DECnet-Plus for OpenVMS documentation.

1.9.2 SRI QIO Compatibility

TCP/IP Services provides support for applications using the INETDRIVER QIO interface developed at Stanford Research Institute (SRI) in 1980-81. An SRI QIO emulator that translates non-TCP/IP Services QIO interfaces into TCP/IP Services QIO programming interfaces can be configured by using the TCPIP$CONFIG procedure.

2

Internetworking and TCP/IP Concepts

An internet is a set of connected networks that act as a coordinated whole, providing interconnections while allowing individual groups to select the network hardware best suited to their needs. This chapter discusses the following networking and TCP/IP concepts:

- Networks (Section 2.1)
- Internets (Section 2.2)
- Client/server model (Section 2.3)
- IP addresses (Section 2.4)
- Routing (Section 2.5)
- Fragmentation (Section 2.5.3)
- Ports (Section 2.6)
- Sockets (Section 2.7)

2.1 Networks

It is important to remember that the Internet is not a new kind of physical network. It is a method of interconnecting physical networks and includes a set of conventions for using these networks that allow the computers they reach to interact. While network hardware plays only a minor role in the overall design, understanding the internet technology requires you to distinguish between the low-level mechanisms provided by the hardware and the higher-level facilities provided by the DIGITAL TCP/IP Services for OpenVMS software.

A network consists of two or more computer systems linked by communications hardware and software. An open network is a network of open systems. An open system is a computer system with communications software that implements formal, international networking standards (for example, the TCP/IP suite of protocols).

A TCP/IP network supports remote system communication, resource sharing, and distributed processing. Network users can access resources on any system in the network and the resources of other vendors' systems.

2.1.1 Local Area Networks

A local area network (LAN) consists of two or more hosts, all connected to one broadcast medium by a high-speed communication medium over short distances. Host computers connect to the communication medium by a hardware interface that often connects to one of the following LANs: Ethernet, FDDI, or Token Ring.

2.1.2 Wide Area Networks

A wide area network (WAN) consists of two or more hosts residing in different locations. Communication transmissions are primarily carried for long distances over telephone lines or a similar medium. Each host connects to the communication medium by a hardware interface connected to the WAN. DIGITAL TCP/IP Services for OpenVMS supports the Serial Line Internet Protocol (SLIP) and the Point-to-Point Protocol (PPP) standards.

SLIP is a framing protocol that sends IP packets over dialup phone lines. SLIP treats each serial link as a separate connection.

PPP is also a framing protocol that sends IP packets over dialup phone lines. Like SLIP, PPP also treats each serial link as a separate network. However, PPP's advantage over SLIP is that it can assign a temporary network number that applies during the time the connection is established. This approach allows internet service providers to make access available to more users because these users only occupy the line, and therefore the network number, during the connection.

2.1.3 Subnets

You can divide a network into subnetworks. A **subnet** is a single network within a group of interconnected networks. Subnets are useful for organizing hosts within a network into logical groups. When you use subnet routing, multiple physical networks can share a single network address. You can use local routers and subnet addresses for each local physical network and cause the network to appear as one single network to other systems. The data from a host on another network routes through a router on to the appropriate subnet, where the destination host receives the data.

For example, your company may have only one assigned network number, even though several physical networks exist. In this scenario, you can use local routers and assign a subnet address to each physical network to make it appear to outside systems that your company has only one network.

2.2 Internets

An internet is a unified, cooperative collection of two or more networks that support a single, universal communication service. The networks are connected by a host that acts as a router. A router forwards data from one host to another host on a different network. Underlying communication mechanisms reside on each network. Between these mechanisms and application programs, low-level protocols are hidden to make the collection of networks appear to be a single large network. These interconnected systems agree to conventions, such as procedures for moving data, that enable each computer to communicate with every other computer on the Internet, whether it is locally or remotely connected.

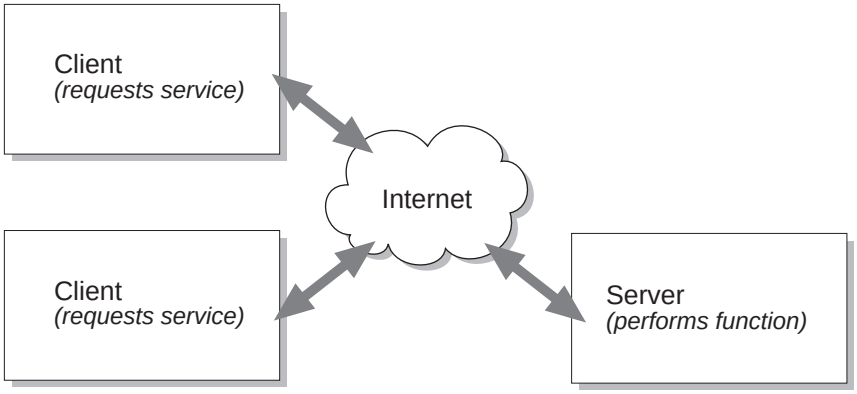
The Internet is a global internet that uses TCP/IP protocols. This entity is accessible to many universities, military installations, government research labs, private companies, and individuals.

2.3 Client/Server Model

Host-to-host communication takes place between two processes. A process is a program that executes on a host. Any process that offers a service to another process over the network is known as a server. Any process that requests a service from another process over the network is known as a client. Clients request a service from the server and wait for the result. The server performs that service as if it were local to the client. Servers are shared processes that support multiple clients.

Figure 2–1 shows a typical client/server relationship.

Figure 2–1 Client/Server Relationship



LKG-09392-94F

2.4 IP Addresses

Each host in an internet must have a unique Internet Protocol (IP) address. To communicate with a remote host, a local user must know the IP address of the remote host and both hosts must reside on the same internet.

The IP address consists of 32 bits (equivalent to 4 bytes or octets) of information. The 4 bytes are usually expressed in **dotted–decimal** notation with each byte a number between 0 and 255. For example, 98.0.2.65 is a valid IP address.

The 4-byte IP address is divided into two parts: the **network ID** and the **host ID**. Within the same network, the IP address of each host has the same network ID but a unique host ID. For example, 201.233.20.125 and 201.233.20.130 are two separate hosts on the same network (201.233.20 is the network ID and 125 and 130 the host IDs of the two hosts).

2.4.1 The Class Assignment Scheme for IP Addresses

In the past, IP addresses were organized into three classes, depending on the size of the network they represent: Class A, Class B, and Class C. Class A networks are extremely large; each Class A network can consist of more than 16 million hosts. Class B networks are smaller with a maximum of 65,534 hosts, and Class C networks must contain fewer than 254 hosts. (These numbers are theoretical. In practice, Class A and Class B networks are usually divided into subnets, which significantly reduces the number of hosts they contain.)

Given that each IP address is 4 bytes, you can tell the relative size of a network based on its IP address, as follows:

- Class A networks range from 1–126 in the first byte of the network ID; the last three bytes identify the host ID. By convention, 127 is reserved as the loopback address. The loopback address is used by the host to communicate back to itself, for testing and other special applications.
- Class B networks range from 128–191 in the first byte of the network ID and 1–254 in the second byte, leaving the last two bytes to identify the host ID.
- Class C networks range from 192–223 in the first byte of the network ID, 0–255 in the second byte, and 1–254 in the third byte, leaving only the fourth byte to identify hosts.

See Table 2–1 for examples of the network address for the three classes of networks.

Table 2–1 Network Address Ranges

Class	First Byte	Second Byte	Third Byte	Fourth Byte	Example
A	1–126	x^1	x^1	x^1	103.x.x.x^1
B	128–191	1–255	x^1	x^1	153.200.x.x^1
C	192–223	0–255	1–254	x^1	203.120.2.x^1

[1] x= Host ID and subnet address

The InterNIC is the central organization that assigned network addresses to other organizations, which in turn assigned the host IDs represented by *x* in Table 2-1. Each organization was responsible for making sure that all attached hosts were properly numbered. Currently, only Class C networks are available.

These days it is more likely that you will obtain your IP addresses from your Internet service provider (ISP) or your company's data communications department.

2.4.2 CIDR Helps Solve Problems Associated with the Class Addressing Scheme

Although the A, B, and C class addressing scheme was easy to understand and implement, it did not foster the efficient allocation of address space. Instead, this scheme resulted in a premature depletion of Class B network space and left medium-sized organizations with Class C space causing the rapid growth in the size of the global routing tables.

Classless Interdomain Routing (CIDR) was developed to keep the internet from running out of address space, replacing the old Class assignment scheme. Under CIDR, only the amount of address space that is actually needed is allocated.

Instead of limiting network identifiers to 8, 16, or 24 bits, CIDR uses prefixes from 13 to 27 bits. This way, blocks of addresses can be assigned for networks as small as 32 hosts or for very large networks with over 500,000 hosts. This addressing scheme allows for more efficient allocation of IP addresses.

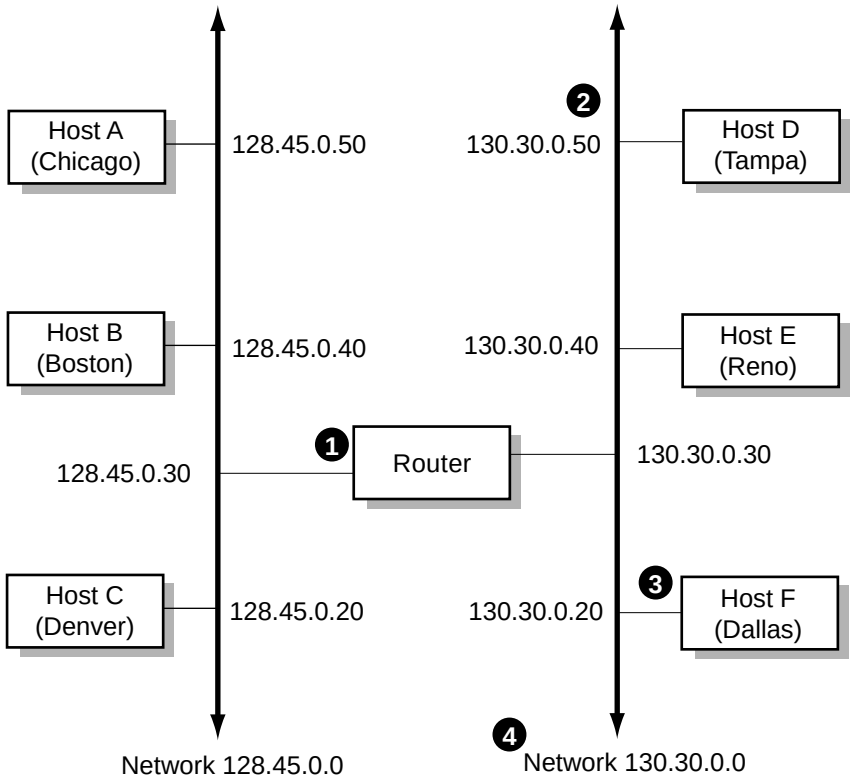
A problem related to the address space waste with the Class A, B, and C allocation scheme was the growing size of Internet global routing tables. As the number of networks on the internet increased, so did the number of routes. Global backbone Internet routers were fast approaching their limit on the number of routes they could support.

CIDR helps solve this problem by enabling **route aggregation**. With route aggregation, a single high-level route entry can represent many lower-level routes in the global routing tables thus minimizing route table entries.

2.4.3 Example of IP Addresses

Figure 2–2 shows an example of assigned IP addresses and names for an internet.

Figure 2–2 IP Addresses and Names of a Sample Internet



1 The router transfers data between hosts on different networks. Each router has an IP address for each network to which it is attached.

2 Each host in an internet has at least one unique name and IP address.

3 Hosts can have multiple names (multihomed). However, one name is usually considered the official name (Host F) and the others are aliases (Dallas).

4 IP addresses are assigned to network interfaces (for example, FDDI, token ring, Ethernet, SLIP, or PPP).

Although the name and address of a host can change, they usually remain stable for extended time periods.

2.4.4 Network Byte Order

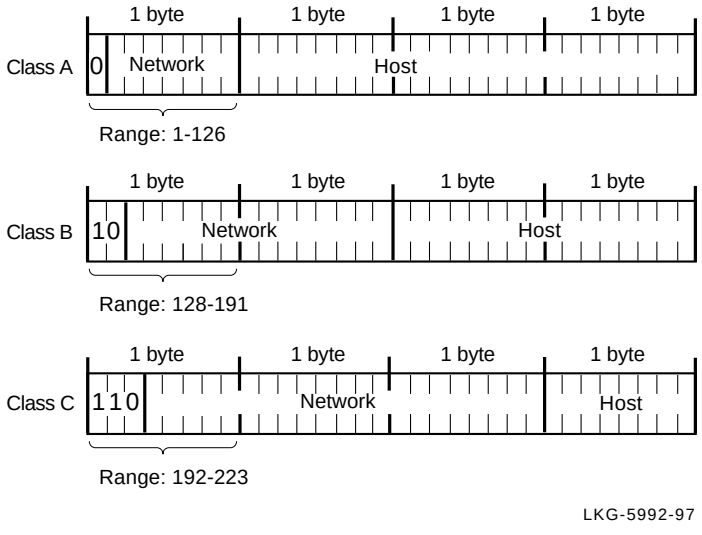
Internet packets carry binary numbers that specify information such as destination addresses and packet lengths, which must be understood by both the sending and receiving hosts. Different machines, however, store 32-bit integers in different ways. The two most common ways are called **Little Endian** and **Big Endian**. With Little Endian style, the lowest memory address contains the low-order byte of the integer whereas with Big Endian, it contains the high-order byte of the integer. Thus, direct copying of bytes from one machine to another may change the value of the number.

To solve this problem, the Internet community has defined the Big Endian style as the **network standard byte order** that all machines must use for binary fields in internet packets. Each host converts binary items from the local representation to network standard byte order before it sends a packet and converts the packet back to the local representation when a packet is received.

In the network standard byte order, the high-order bits in the network number designate the network class of the IP address. 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. For a Class C network, the first three high-order bits are 110.

Figure 2–3 shows the bit positions of the IP address for the three network classes.

Figure 2–3 IP Network Classes



2.4.5 Subnet Addressing

Subnetting hides the details of internal network organization to external routers and reduces the size of the internet's routing tables. To reach any host within a subnet, external routers only need to know the path to a single host. Subnet routing requires a different interpretation of IP addresses. A certain number of

bits are taken from the octets in the host part of the address and used to specify subnet information. This is called a **subnet mask**.

The subnet mask informs the system which bits of the IP address to interpret as the network, subnet, and host addresses. A subnet mask is a 32-bit number. There is a one-to-one correspondence between the 32 bits in the subnet mask and the 32 bits in the IP address. (A subnet mask may also be referred to as a *network mask*.)

For each bit in the subnet mask that is turned on (binary 1), the corresponding bit position in the IP address is interpreted as part of the network and subnet address.

The decimal number 255 is 11111111 in binary notation. The value 255 means that an entire 8-bit field is turned on because each bit position is a 1. Generally, the entire 8-bit field is turned either on (255) or off (0). Values other than 255 or 0 can be used. However, by using 255 or 0, you make it easier for users to differentiate between the network, host, and subnet fields.

If the subnet mask bit position is part of the host ID and is turned on, the corresponding bit in the IP address is interpreted as part of the subnet address. If the subnet mask bit position is part of the host ID and is turned off, the corresponding bit in the IP address is interpreted as part of the host ID.

All bits in the first (leftmost) byte of the subnet mask must be turned on (decimal value of 255, binary value of 11111111), because the first byte of the IP address must always be interpreted as the network ID regardless of whether there are subnets. If a bit in the first byte of the subnet mask is turned off, part of the network ID of the IP address is interpreted as part of the host ID. This may cause errors.

The second and third bytes of the new mask are usually either 255 or 0, depending on how the IP address is to be interpreted. The fourth byte is usually 0, to indicate that the fourth byte of the IP address is part of the host ID.

Figures 2–4 and 2–5 illustrate the way different subnet masks affect the subnet address. As illustrated in Figure 2–4, a Class A subnet mask can be 255.255.0.0. When the subnet mask is 255.255.0.0, the first byte is the network ID, the second byte is the subnet ID, and the third and fourth bytes are the host ID.

Figure 2–4 Class A Network Mask, Example 1

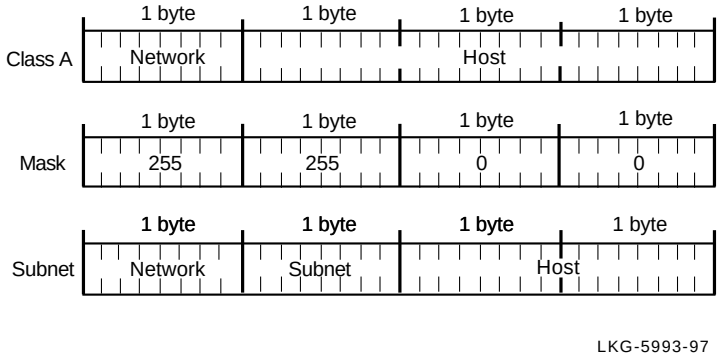
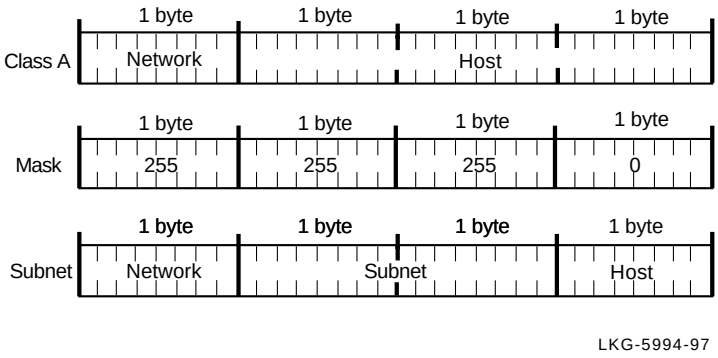


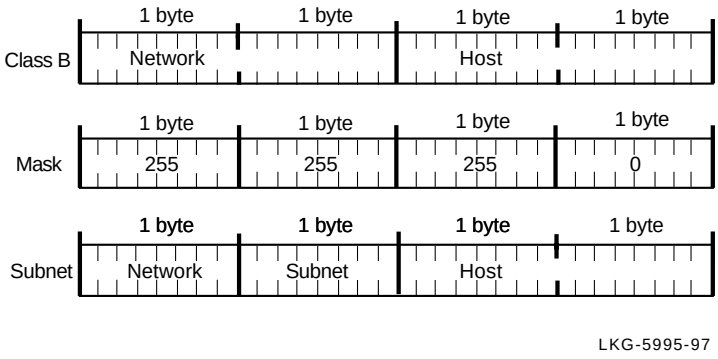
Figure 2–5 shows a Class A network with a subnet mask of 255.255.255.0. If the subnet mask is 255.255.255.0, the first byte is the network ID, the second and third bytes are the subnet ID, and the fourth byte is the host ID.

Figure 2–5 Class A Network Mask, Example 2



If a Class B network uses 255.255.255.0 (as shown in Figure 2–6) for a subnet mask, the first and second bytes are the network ID, the third byte is the subnet ID, and the fourth byte is the host ID.

Figure 2–6 Class B Network Mask



Normally, Class C networks do not have subnets, because only 8 bits are allocated for the host part of the IP address. Eight bits may not be enough to divide between a subnet address and a host address.

The default subnet masks for each class are as follows:

- Class A — 255.0.0.0
- Class B — 255.255.0.0
- Class C — 255.255.255.0

2.4.6 Broadcast Mask

The broadcast mask interprets the IP address as a broadcast address. The broadcast address allows messages to be sent to all the hosts on the network at the same time. If you use subnets, all the hosts on the same subnet must have the same IP broadcast address.

The default format of the broadcast address consists of the network ID followed by all 1s. The network ID includes the subnet, if there is one. Although the all-zeroes method of forming a broadcast address has not been used for many years, at times, you may need to specify an alternate broadcast address for testing or compatibility purposes. The default is usually adequate.

If you know the IP address and the subnet mask for a particular host, you can determine the broadcast address by using the following formula:

$$(NOT\,networkmask)\,OR\,(internetaddress)$$

For example, if a host has an IP address of 128.50.100.100 and its network mask is 255.255.0.0 (the default), its broadcast mask is 128.50.255.255. The *NOT* of its subnet mask is 0.0.255.255. You then substitute the first two fields of the IP address for the two 0s to get the broadcast address.

Table 2–2 lists examples of broadcast addresses.

Table 2–2 Broadcast Addresses

Host IP Address	Host Number	Network Class	Network Number	Network Mask	Broadcast Address
3.0.0.10	10	A	3.	255.0.0.0	3.255.255.255
11.1.0.12[1]	12	A	11.1.	255.255.0.0	11.1.255.255
129.39.0.15	15	B	129.39.	255.255.0.0	129.39.255.255
128.45.2.8[1]	2.8	B	128.45.	255.255.0.0	128.45.2.255
192.0.1.8	8	C	192.0.1.	255.255.255.0	192.0.1.255
192.0.1.223	223	C	192.0.1.	255.255.255.0	192.0.1.255

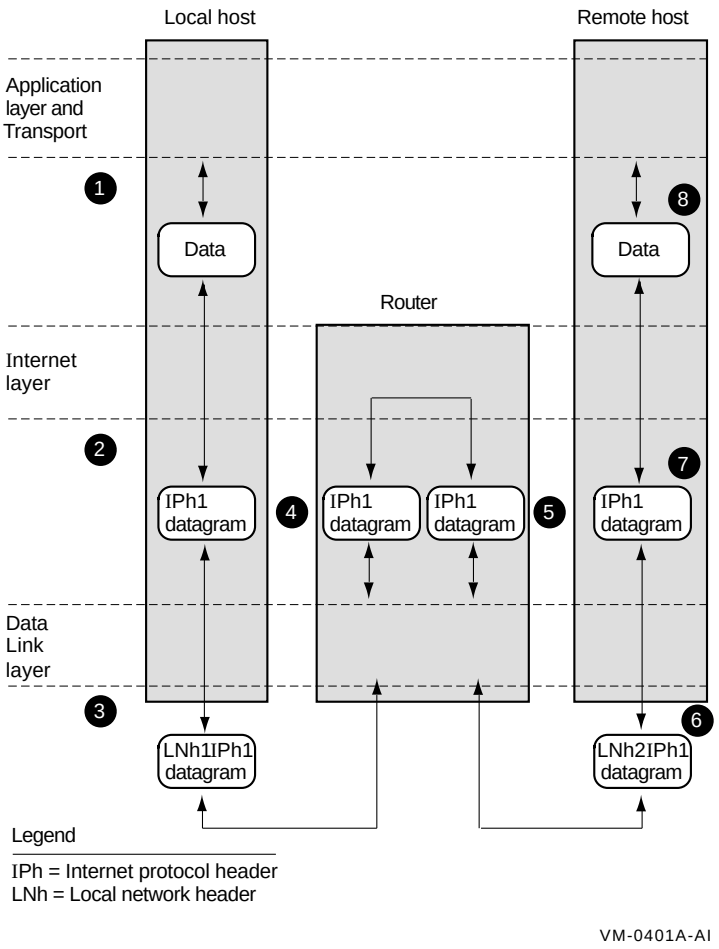
[1] Shows use of subnet address

2.5 Routing

Routing is the process of moving information, in the form of datagrams, from one host to another over the network. Under the class addressing scheme, a router receives an IP packet and extracts its destination address. The destination is classified (literally) by examining its first one to four bits. Once the address's class is determined, it is broken down into network and host bits. Routers ignore the host bits, and only needed to match the network bits to find a route to the network. Once a packet reaches the target network, its host field is examined for final delivery.

Figure 2–7 shows internet routing.

Figure 2–7 Internet Routing



Internet routing follows this progression:

1. The sending application program (Application layer and Transport layer) prepares its data and calls on its Internet layer.

   The Internet layer receives the data and the destination address as arguments of the call.

2. The Internet layer reads the destination IP address from IPh1 and consults its routing table to determine which interface to send it on.

   The Internet layer sends this datagram to the Data Link layer.

3. The Data Link layer creates a local network header (Lnh1) and attaches the datagram to it. The datagram with the attached header is sent by means of the local network (local network 1).

4. If the datagram is sent to a router, the Data Link layer of the router removes the local network header (Lnh1) and delivers the datagram to the Internet layer.

5. The Internet layer reads the destination of the IP address from the Internet header (IPh1) and consults the routing table to determine which port to send it to.

   The Internet layer sends this datagram to the Data Link layer.

6 The Data Link layer creates a local network header (Lnh2), attaches the datagram to it, and sends the results to the destination host on local network 2.

7 The destination host removes Lnh2 at the Data Link layer and passes the datagram to the Internet layer.

8 The Internet layer determines whether the datagram is for an application program in the host. If it is, the Internet layer removes the Internet header (lph2) and passes the data to the application program in response to a system call. The data, the source address, and other parameters are also passed to the application.

2.5.1 Autonomous Systems and Routing Protocols

LANs and WANs interconnected by IP routers form a group of networks called an internet. For administrative purposes, an internet is divided into **autonomous systems**. An autonomous system (AS) is simply a collection of routers and hosts.

Routers inside an autonomous system use an interior gateway protocol to communicate network topology changes to each other. Routers in separate autonomous systems use an exterior gateway protocol to communicate. The TCP/IP Services product supports two dynamic interior protocols: RIP and OSPF; and two exterior protocols: BGP and EGP.

The following sections provide more information.

2.5.1.1 Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) enables routers in the same autonomous system to exchange routing information by means of periodic RIP updates. Routers transmit their own RIP updates to neighboring networks and listen for RIP updates from the routers on those neighboring networks. Routers use the information in the RIP updates to keep their internal routing tables current. For RIP, the "best" path to a destination is the shortest path (the path with the fewest hops). RIP computes distance as a metric, usually the number of hops (or routers) from the origin network to the target network.

2.5.1.2 Open Shortest Path First (OSPF) Protocol

The Open Shortest Path First (OSPF) protocol is an IGP intended for use in large networks. Using a link state algorithm, OSPF exchanges routing information between routers in an autonomous system then routers synchronize their databases. Once the routers are synchronized and the routing tables are built, the routers will forward topology information only in response to some topological change. For OSPF, the "best" path to a destination is the path that offers the least cost metric delay. In OSPF, cost metrics are configurable, allowing you to specify preferred paths.

OSPF supports CIDR and can carry supernet advertisements within a routing domain.

2.5.1.3 Border Gateway Protocol (BGP)

The Border Gateway Protocol (BGP) is an exterior gateway protocol used to exchange network reachability information with other BGP systems. BGP routers form relationships with other BGP routers. BGP routers transmit and receive current routing information over a reliable transport layer connection. Because a reliable transport mechanism is used, periodic updates are not necessary.

BGP updates contain "path attributes" that describe the route to a set of destination networks. When multiple paths are available, BGP compares these path attributes to choose the preferred path.

2.5.1.4 Exterior Gateway Protocol (EGP)

The Exterior Gateway Protocol (EGP-2) is an exterior gateway protocol used to exchange network reachability information between routers in different autonomous systems. (A protocol such as RIP or OSPF is used within an AS to facilitate the communication of routing information within the autonomous system.) The routers that serve as the end points of a connection between two autonomous systems run an exterior gateway protocol such as EGP.

Routers establish EGP neighbor relationships in order to periodically exchange reliable network reachability information. The router uses this information to maintain a list of gateways, the networks the gateways can reach, and the corresponding distances.

2.5.2 Routing Daemons

Routing protocols distribute information that reflect dynamic network conditions and update the routing table accordingly. Dynamic routing tables use information received by means of routing protocol updates; when routes change, the routing protocol can switch to a backup route and can determine the best route to a given destination.

Routing daemons implement a routing policy, that is, the set of rules that decide which routes go into the routing table. A routing daemon writes routing messages to a routing socket causing the kernel to add a new route, delete an existing route, or modify an existing route.

The kernel also generates routing messages that can be read by any routing socket when events occur that may be of interest to the process, for example, the interface has gone down or a redirect has been received.

The TCP/IP Services product implements two routing daemons, the Routing Daemon (ROUTED) and the Gateway Routing Daemon (GATED).

2.5.2.1 The Routing Daemon (ROUTED)

ROUTED listens on a User Datagram Protocol (UDP) socket for packets with routing information. If the host is a gateway (internet router), it periodically supplies copies of its routing tables to directly connected hosts or networks.

When the dynamic routing server is started it finds all active Internet interfaces (except those marked in loopback). If multiple interfaces are present, the dynamic router assumes that the host forwards packets between networks. ROUTED then transmits a RIP request packet on each interface. If the interface supports broadcast packets, ROUTED sends a broadcast packet. Otherwise, it sends a normal packet and listens for RIP request and RIP response packets from other hosts.

When a RIP request packet is received, ROUTED formulates a reply based on the information maintained in its internal tables. The RIP response packet generated by the server contains a list of known routes, each marked with a hop count metric. The hop count is the number of hops between two hosts, based on the number of different routers needed to traverse the distance between the two hosts. A hop count of 16 or greater is considered infinite.

If one or more of the following conditions exist, the RIP response packets received by ROUTED are used to update the internet routing tables:

- No routing table entry exists for the destination network or host, and the metric indicates the destination is reachable. (That is, the hop count is not infinite.)
- The source host of the packet is the same as the gateway (router) in the existing routing table entry. That is, updated information is being received from the gateway (internet router) through which packets for the destination are being routed.
- The existing entry in the routing table has not been updated for some time (defined to be 90 seconds) and the route is at least as cost effective as the current route.
- The new route describes a shorter route to the destination than the one currently stored in the routing tables. To decide this, the hop count metric of the new route is compared to the one stored in the gateway's internal routing tables.

When an update is applied, ROUTED records the change in its internal tables and generates a RIP response packet to all hosts and networks to which it is directly connected. ROUTED waits a short period of time (no more than 30 seconds) before modifying the internet routing tables to allow possible unstable situations to be resolved.

In addition to processing incoming packets, ROUTED periodically checks the internet routing table entries. If an entry has not been updated for 3 minutes, the entry's metric is set to infinity and marked for deletion. Deletions are delayed an additional 60 seconds to ensure that the invalidation is propagated throughout the internet.

Hosts that act as routers supply their routing tables to all directly connected hosts and networks every 30 seconds. The RIP response is sent to either the broadcast destination, an address on a point-to-point link, or the gateway's address on other networks. The normal routing tables are bypassed when sending RIP response packets.

The reception of RIP response packets on a network is used to determine whether that network and interface are functioning correctly. If no RIP response packet is received on an interface, another path may be chosen to route around the interface or the route may be dropped if no alternative is available.

2.5.2.2 Gateway Routing Daemon (GATED)

In contrast to ROUTED, which handles only RIP protocols, GATED handles multiple routing protocols. GATED can be configured to handle all supported protocols or any subset. In many environments, GATED will replace ROUTED because it provides the following benefits:

- For systems using more that one routing protocol, GATED combines the routing information it learns from each and selects the "best" routes.
- External announcments can adjust dynamically when interior routes are changed.
- Routing protocols are easily configured from a single file called TCPIP$GATED.CONF.

2.5.3 Fragmentation and Path MTU

Most types of networks have an upper limit on the number of bytes of data that can be transmitted at one time. This limit is called the **MTU**, or maximum transmission unit. If a datagram is larger that the MTU, IP performs **fragmentation**, breaking a large datagram into fragments so that each fragment is smaller than the MTU.

Fragmentation is used, for example, when the datagram originates in a network that allows large packets, but to reach its destination, the packet must cross a network that supports a limited packet size. Fragmentation is also used when there is no router, but applications send messages that are longer than the Network Interface layer supports. For example, the NFS server transfers information in 8000-byte packets, but Ethernet only supports 1518-byte packets. Therefore, IP fragments the 8000-byte datagram into 6 segments of no more than 1518 bytes each.

A router can break up an internet datagram into smaller internet datagram fragments. The fragments can be further broken into smaller fragments at subsequent routers.

The fragment format is designed so that the destination IP layer can reassemble fragments into their original form before delivering the complete datagram to a user.

When two hosts are communicating across multiple networks, it is important to know the smallest MTU of any data link between the two hosts. This is called **path MTU**. Because the path MTU between hosts can change at any time, a **path MTU discovery** mechanism is employed to determine the path MTU at any given time.

Knowing the current path MTU value allows a host to limit the size of packets it transmits, thereby avoiding fragmentation by intervening routers. The TCP/IP Services product performs path MTU discovery and limits the size of its TCP packets.

2.6 Ports

As explained in Section 2.4, each host on an internet is identified by a unique IP address. However, because numerous processes run concurrently, a client process on a local host needs more than just the remote host's IP address to connect to a server process on the remote host.

In addition to the IP address, the client process must specify the **port number** of the server process to which it wants to connect. The combination of an IP address and a port number identifies the unique connection point of the requested process. Port numbers range from 1 to 65535.

Every client and server process has an associated port number. The UDP header and TCP header each contain the source port number and the destination port number. Because the IP header specifies the protocol, TCP ports are independent of the UDP ports. TCP port 1035, for example, is different from UDP port 1035.

There are numerous advantages to using ports instead of attempting to send messages directly to receiving processes with process names or identification numbers:

- Heterogeneous operating systems define processes differently. To use process names requires that the internet architecture include a definition of each process or process name.

- Not all senders have enough information to identify a particular process on another host.
- Process IDs change.

2.6.1 Well-Known Ports

By convention, port numbers 1 to 1023 are called **well-known ports** and are assigned to specific applications on all servers running TCP/IP. For example, some of the most common well-known port numbers are:

Port Number	Application
21	FTP (File Transfer Protocol)
23	TELNET
25	SMTP (Simple Mail Transfer Protocol)
69	TFTP (Trivial File Transfer Protocol)
80	HTTP (Hypertext Transfer Protocol)
110	POP (Post Office Protocol)

2.6.2 Privileged Ports

In addition to well-known ports, port numbers 256 to 1023 are called **privileged ports**. This meaning of privilege depends on the operating system. In general, when a host receives a message from a privileged port on a remote sender, the local host assumes that the remote host has checked the security or authenticated the application using the port. The remote host is responsible for ensuring that only privileged applications or users can access privileged ports.

Note

Under OpenVMS, a process needs one of the following privileges to bind to the local privileged ports (1 to 1023):

- SYSPRV
- OPER
- BYPASS

2.6.3 Ephemeral Ports

Before requesting a process from a server, a client process is assigned an unused port number (usually ports 1024 to 5000) from its local host. This temporary port number is contained in the header information along with the IP address.

After the server completes the request, it can reply to the client using the port and IP address information contained in the requesting header.

Because the port number assigned to the client process is temporary, it is called an **ephemeral port number**. When the first client process is finished, the port number is free to be assigned to another process.

2.6.4 Port Binding

To communicate through either TCP or UDP, a process must be **bound** to a port. This means that the sending and receiving processes establish a connection and exchange command requests. A port that is bound to a process is known as an **active port**. A process can bind to any number of ports.

2.6.5 Port Assignment

Ports can be permanently associated with specific servers provided by specific image files. An association between a port and an image file is called a **port assignment**. To create a port assignment, the system manager or a process assigns the port to a server. If the service is to be started by the Auxiliary server, the service must have an entry in the service database (TCPIP$SERVICE.DAT).

The server's entry in the DIGITAL TCP/IP Services for OpenVMS service database contains the following information:

- Server file name
- Protocol used by the server (TCP or UDP)
- Port number
- Startup information
- Logging information
- Access information

For more information, see the SET SERVICE command in the *DIGITAL TCP/IP Services for OpenVMS Management Command Reference* manual.

2.7 Sockets

As explained in Section 2.6, the unique connection point of a process is identified by the IP address and port number. This point is called a **socket**. Because network communication consists of two connection points (source and destination), a **socket pair** fully describes the connection:

```
{source_IP_address, source_port, destination_IP_address, destination_port}
```

For example, the following is a valid socket pair:

```
{192.43.235.2, 1500, 192.43.235.6, 21}
```

where 192.43.235.2 is the source IP address and 1500 is the source port number and 192.43.235.6 is the destination IP address and 21 is the destination port number. (It's also interesting to note that 1500 is an ephemeral port number and 21 is the well known port number for FTP.)

2.7.1 Socket APIs

The concept of a socket is important to creating networked applications. The BSD **socket API** consists of calls that programmers use to write application programs that transfer data between two hosts. Each application generally needs to contain both client and server functionality.

The following is a typical sequence of socket calls a *client* process could use to access a server:

- Create a socket with a call to `socket( )`.
- Connect the socket to the server with a call to `connect( )`.

- Send and receive data with calls to `send( )` and `recv( )`.
- Shut down the connection with a call to `shutdown( )` and `close( )`.

A *server* process needs to prepare itself before it can accept clients. To do this, it might use a sequence of calls such as the following:

- Create a socket with a call to `socket( )`.
- Register the socket with a well-known port address with a call to `bind( )`.
- Create a queue where clients can place connection requests with a call to `listen( )`.
- Accept client request with a call to `accept( )`.
- Receive and send data with calls to `recv( )` and `send( )`.

For more detailed information about socket calls, see the *DIGITAL TCP/IP Services for OpenVMS System Services and C Socket Programming* manual.

The DIGITAL TCP/IP Services for OpenVMS product supports two socket communication APIs:

- OpenVMS QIO system services
- BSD socket interface

In both of these APIs, three characteristics may be specified to create a socket:

- Address family
- Protocol type
- Protocol

2.7.1.1 Address Family

An address family is the set of collective common properties of processes that communicate through sockets. The DIGITAL TCP/IP Services for OpenVMS product supports the Internet (AF_INET) address family.

2.7.1.2 Socket Type

Socket types are the communication properties visible to the user. Normally, processes communicate only between sockets of the same type. The available socket types are: stream, datagram, and raw.

- A stream socket (TCP) supports a bidirectional, reliable, and sequenced flow of data without record boundaries. The data is guaranteed to arrive at the receiving processes in sequence, without duplication.
- A datagram socket (UDP) provides a bidirectional flow of data that does not guarantee the receiving process gets the messages in sequence, without duplication, or at all. The record boundaries of the data are preserved.
- A raw socket (RAW IP) provides access to the underlying communication protocols that support sockets. Users developing new communication protocols, for example, can take advantage of raw sockets to provide direct access to the IP protocol.

2.7.1.3 Protocol

In most cases, the socket protocol properties are either TCP or UDP. TCP is used for connection-oriented sockets and is more complex than the lower-overhead connectionless sockets of UDP. This parameter is usually omitted.

See the *DIGITAL TCP/IP Services for OpenVMS System Services and C Socket Programming* manual for detailed information.

2.8 NTP Concepts

Time synchronization is important in client/server computing. For example, systems that share common databases require coordinated transaction processing and timestamping of instrumental data. Network Time Protocol (NTP) provides synchronization traceable to clocks of high absolute accuracy and avoids synchronization to clocks with incorrect time.

2.8.1 Synchronized Time Keeping

Synchronized time keeping means that hosts with accurate system timestamps send time quotes to each other. Hosts running NTP may be either time servers or clients although they are often both servers and clients.

NTP does not attempt to synchronize clocks to each other. Rather, each server attempts to synchronize to **Universal Coordinated Time** (UTC) using the best available source and best available transmission paths to that source. NTP expects that the time being distributed from the synchronization root is derived from an external source of UTC, such as a radio clock.

If your network is isolated and you cannot access other NTP servers on the internet, you can designate one of your hosts as the reference clock to which all other hosts will synchronize.

2.8.2 How Hosts Negotiate Synchronization

Each host has its identifying **stratum** number encoded within UDP datagrams. Peers communicate by exchanging these timestamped UDP datagrams. NTP uses these exchanges to construct a list of possible synchronization sources then sorts them based on the results of complex filtering and selection algorithms. Peers are accepted or rejected leaving only the most accurate and precise sources.

NTP evaluates any new peer to determine if it qualifies as a new (more suitable) synchronization source.

NTP rejects the peer under the following conditions:

- The peer is not synchronized.
- The stratum is higher than the current source's stratum.
- The peer is synchronized to the local node.

NTP accepts the peer under the following conditions:

- There is no current time source.
- The current source is unreachable.
- The current source is not synchronized
- The new source's stratum is lower than the current source.
- The new source's stratum is the same as the current source, but its distance is closer to the synchronization source by more than 50%.

3

BIND Service Concepts

The DIGITAL TCP/IP Services for OpenVMS software supports the Berkeley Internet Name Domain (BIND) service, which is a popular implementation of the Domain Name Service (DNS). BIND has been ported to many platforms including UNIX, Windows NT and OpenVMS. DIGITAL TCP/IP Services for OpenVMS release version 5.0 implements the Internet Software Consortium's (ISC) BIND 8.1.2.

Before you add BIND servers to your network, it is useful to understand basic BIND service concepts as they apply to the TCP/IP Services for OpenVMS product. This chapter describes:

- Overview of the BIND Service (Section 3.1)
- BIND Service Components (Section 3.2)
- Domains (Section 3.3)
- Domain Names (Section 3.4)
- Zones (Section 3.5)
- Reverse Domain (Section 3.6)
- BIND Server Functions (Section 3.7)
- BIND Server Configuration File (Section 3.8)
- BIND Server Database Files (Section 3.9)
- BIND Resolver (Section 3.10)

3.1 Overview of the BIND Service

DNS has a hierarchical, distributed namespace which makes it easy for humans to remember and locate the many hosts located throughout the internet. Since computers remember and locate the same hosts through a numerical address, computers need a method for converting the host name to a numerical address.

BIND is a lookup service that maps host names to IP addresses and IP addresses to hostnames in response to queries from other BIND servers and clients in the network. BIND can also provide information on available mail servers and well-known services for a domain.

Based on a client/server model, BIND servers maintain databases of host names, IP addresses, mail records, text records, and other network objects. When client systems require this information, they query the servers.

The Internet Network Information Center (InterNIC) provides the Internet community with services for domain registration, directories and databases, and other information. See Appendix A for information about network and domain registration.

3.2 BIND Service Components

The BIND service contains two parts: the BIND resolver and the BIND server.

- **BIND resolver** — the client software interface that:
  - Formulates queries
  - Sends queries to BIND servers for answers
  - Interprets the server's answer and
  - Returns the information to the requesting network application.
- **BIND server** — the server software that responds to client queries by providing:
  - Authoritative or non-authoritative answers to queries about host names/IP addresses for which the server has an answer.
  - Information on other authoritative servers that can answer queries about host names/IP addresses for which the server does not have an answer.
  - Information on how to get closer to the answer if the server does not have an answer or information on other authoritative servers.
  - Information about mail servers and other network application servers (for example, `ftp`, `telnet`).

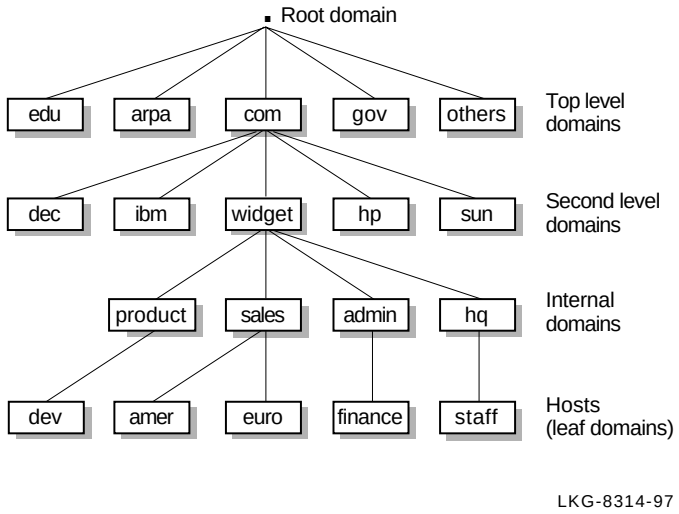
3.3 Domains

The Internet name space is based on a hierarchical tree structure. Each node on the tree is referred to as a domain or a subdomain. A domain is an administrative entity that allows for decentralized management of host names, addresses, and user information. Domains can refer to an administrative point on the name space tree or a specific host. A domain is identified by a domain name and includes the name space at or below the domain name.

A subdomain is a domain that is part of a larger domain. You can consider every domain in the name space below the root domain to be a subdomain. You can also refer to any subdomain as a domain.

Figure 3–1 illustrates a typical Internet domain hierarchy.

Figure 3–1 Internet Domain Hierarchy



3.3.1 Top-Level Domains

Table 3–1 lists some of the commonly used top-level domains.

Table 3–1 Top-Level Domains

Domain	Description
arpa	The Arpanet (gradually being phased out)
ca	Canada
com	Commercial institutions
edu	Educational institutions
gov	Government departments or agencies
mil	Military organizations
net	Network-type organizations, such as network service centers, consortia, and information centers
org	Miscellaneous organizations, such as professional societies and similar nonprofit organizations
us	United States

Countries can register with the InterNIC as top-level domains provided they name themselves after a two-letter country code listed in the international standard ISO-3166. If a country code is identical to a state code that the U.S. Postal Service uses, the country can request a three-letter code.

3.3.2 Domain Administrator Role

Typically, each domain has a domain administrator responsible for coordinating and managing the domain. The domain administrator registers a second-level or lower domain by interacting with the domain administrator in the next higher level domain.

The domain administrator's duties include:

- Understanding the concepts and procedures of the BIND service
- Ensuring reliable service
- Ensuring that the BIND data is current
- Taking prompt action when necessary, for example, if protocols are violated or other serious misbehavior occurs
- Controlling the assignments of the host and domain names

The domain administrator furnishes users with access to names and name-related information both inside and outside the local domain.

3.4 Domain Names

The InterNIC assigns names for all top-level domains as well as domains directly below the top-level domains. Individuals are responsible for assigning lower-level domains and host names.

Each domain (or subdomain) has a label. For example, the label for the top-level domain for commercial organizations is `com`. A label is unique within its parent domain.

The concatenation of all the domain labels from the top-level domain to the lowest-level domains listed from right to left and separated by dots is called a fully qualified domain name. For example, the domain name for a subdomain within the `com` domain, would be `abc.com`; `abc` is the label for the ABC company's subdomain, and `com` is the label for the commercial domain. This structure allows administration and data maintenance to be delegated down the hierarchical tree.

Notes

The term **domain name** is sometimes used when referring to a specific domain label.

The name of the root domain of the name space is a dot (.) .

3.4.1 Types of Domain Names

There are two types of domain names: the fully qualified name and the relative name.

- The fully qualified name represents the complete domain name. This is also known as the absolute or canonical name. For example:

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

  A domain name that is fully qualified is absolute. You should not append further BIND extensions to the name.

- The relative name represents the starting name (label) of an absolute domain name. Relative names are incomplete, but are completed by the BIND service, using knowledge of the local domain. Relative host names such as `chicago.cities` are automatically expanded to the fully qualified domain name when given in a typical command.

3.4.2 Canonical Names and Aliases

Hosts and resources often have more than one name that identifies them. The BIND service supports the use of canonical names and aliases. A canonical name is a host's or resource's official name, while other names that identify the same host or resource are considered aliases or nicknames. Nicknames are useful if a host changes any part of its canonical name (for example, host name or domain). People who continue to use the nickname can still reach the right host or resource.

3.4.3 Domain Name Format

Domain and host labels have the following format:

- Contains characters, digits or a hyphen
- Must begin with a character or digit
- Must not end with a hyphen
- A maximum of 63 characters for each label
- A maximum of 255 characters in a fully qualified domain name

Note

Characters in the range of 128 through 255 are subject to having their high bit cleared because some software does not preserve the high bit.

Although label names can contain up to 63 characters, it is best to choose names that are 12 characters or less because the canonical (fully qualified) domain names are easier to keep track of if they are short. The sum of all the label characters and label lengths is limited to 255.

Note

Domain names are not case sensitive. However, the case of entered names is preserved whenever possible.

Read from right to left for the following fully qualified domain name: `euro.sales.widget.com`.

- The `com` label refers to the commercial top-level domain.
- The `widget` label refers to the widget domain, a subdomain of the commercial domain.
- The `sales` label refers to the sales domain, a subdomain of the widget domain.
- The `euro` label refers to the host called `euro`, a subdomain of the sales domain.

3.5 Zones

For management reasons, a domain can be divided into zones which are discrete, non-overlapping subsets of the domain. A zone usually represents an administrative or geographic boundary and authority for the zone may or may not be delegated to another responsible group or person. Each zone starts at a designated level in the domain name tree and extends down to the leaf domains (individual host names), or to that point in the tree where authority has been delegated to another domain.

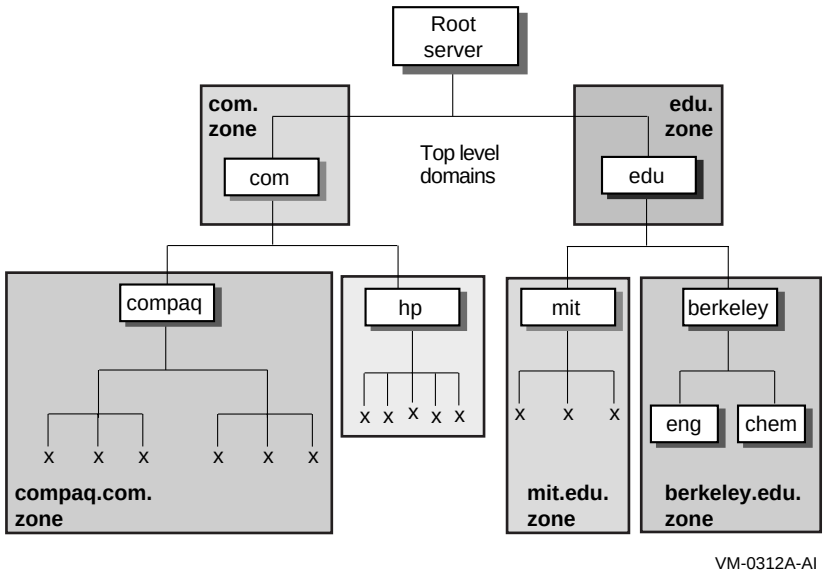
A common zone is a second-level domain `abc.com`, for example. Many second-level domains divide their zones into smaller zones. For example, a university might divide their domain name space into zones based on departments. A company might divide their domain name space into zones based on branch offices or internal divisions. Authority for the zone is generally delegated to the department or branch office. The department or branch office then has the responsibility for maintaining the zone data.

All the data for the zone is stored on the master server in zone files.

3.5.1 Zone Hierarchy Example

Figure 3–2 shows the hierarchy of the internet, two top-level domains, and some of the major zones. For example, in Figure 3–2, everything below `com` is in the `com` top-level domain; the zones are within the shaded boxes. The host names are depicted by an `x`.

Figure 3–2 Hierarchy of BIND Zones and Domains on the Internet



3.5.2 Delegation

When a zone is very large and difficult to manage, authority for a portion of the zone can be delegated to another server and the responsibility for maintaining the zone information is also delegated.

For example, in Figure 3–2 the edu zone contains many educational organizations. Each organization is delegated the authority for managing their portion of the edu zone, thereby creating a subzone. In the example, mit.edu and berkeley.edu are subzones of the edu zone and each organization has the responsibility for maintaining the zone information and the master and slave servers for their respective zones.

3.6 Reverse Domain

The internet has a special domain used for locating gateways and supporting internet address-to-host name lookups. The mapping of internet addresses to domain names is called reverse translation. The special domain for reverse translation is the IN-ADDR.ARPA domain.

3.7 BIND Server Functions

If a network consists of relatively few hosts, host name/IP address translations can be accomplished by using a centralized hosts database file.

As soon as a network connects to another network or the number of hosts grows large, there needs to be a more robust method for performing host name/IP address translation. And in particular, when a network is part of the worldwide Internet, no single database can keep track of all addressing information. A considerable number of hosts and network domains are added, changed, and deleted every day.

BIND uses several different types of name servers to ensure that all queries are resolved quickly and efficiently:

- Root servers
- Master name servers
- Slave name servers
- Forwarder servers
- Caching-only servers

When a client makes a query, a name server can be in one of three possible states:

- It knows the IP address authoritatively, based on addresses residing in its data files.
- It knows the IP address but not authoritatively from data cached in its memory from a previous query.
- It does not know the address and must refer the query to another server.

The following sections discuss the different types of name servers and their primary responsibilities in the distributed environment of BIND and DNS.

3.7.1 Root Name Servers

Root name servers are the master name servers for the top-level domains of the Internet root zone. If they are not the authority for a zone, they know who to contact to find out who is the authority.

If a non-root server receives a request for a name not within its zone, the server starts name resolution at the root zone and accesses the root servers to get the needed information.

The InterNIC determines root servers for the top-level domain. The following root servers are valid:

Table 3–2 Internet Root Servers

Current Server Name	Former Server
A.ROOT_SERVERS.NET	ns.internic.net
B.ROOT_SERVERS.NET	ns1.isi.edu
C.ROOT_SERVERS.NET	c.psi.net
D.ROOT_SERVERS.NET	terp.umd.edu
E.ROOT_SERVERS.NET	ns.nasa.gov
F.ROOT_SERVERS.NET	ns.isc.org
G.ROOT_SERVERS.NET	ns.nic.ddn.mil
H.ROOT_SERVERS.NET	aos.arl.army.mil
I.ROOT_SERVERS.NET	nic.nordu.net
J.ROOT_SERVERS.NET	
K.ROOT_SERVERS.NET	
L.ROOT_SERVERS.NET	
M.ROOT_SERVERS.NET	

These servers change from time to time, so the servers listed in Table 3–2 may not be the current list. You can obtain the up-to-date list by:

- Copying the `named.root` file maintained at the InterNIC by using FTP anonymous login to `ftp.rs.internic.net` (198.41.0.6). The file is in the `domain` subdirectory.
- Using the DIG utility
- Using the on-line registration process at the InterNIC web site.

These servers know about all the top-level DNS domains on the Internet. You must know about these servers when making queries about hosts outside of your local domain. The host names and internet addresses of these machines change periodically. Therefore, check with the InterNIC periodically to obtain changes and store them in the hints file of the BIND name servers (usually called `TCPIP$ROOT.HINT` on a TCP/IP Services system).

3.7.2 Master Name Server

There are two types of master servers: a primary master name server and a slave name server (also called a secondary master name server).

The primary master server is the primary authority for the zone. The primary master server has complete information about the zone which is stored in it's database files. If network information changes, those changes are captured in the master server's database files.

A server can be a master server for more than one zone, acting as the primary master name server for some zones and a slave name server for others.

It is possible to have more than one master server; however, maintaining two sets of database files requires making the same changes to both sets of files. A more efficient solution is to have one master server and one or more slave servers who obtain their zone information from the master server.

3.7.3 Slave Name Server

A slave name server is an administrative convenience providing redundancy of information and sharing the load of the primary name server. A slave name server receives its authority and zone data from a primary master name server. Once running, a slave name server periodically checks with the primary master name server for zone changes. If the slave's serial number is less than the master's serial number, the slave requests a zone transfer.

The slave name servers poll the master server at predetermined intervals specified in the zone database files. A time lapse between changing the master server's databases and the slave name servers requesting the update may exist.

3.7.4 Forwarder Servers

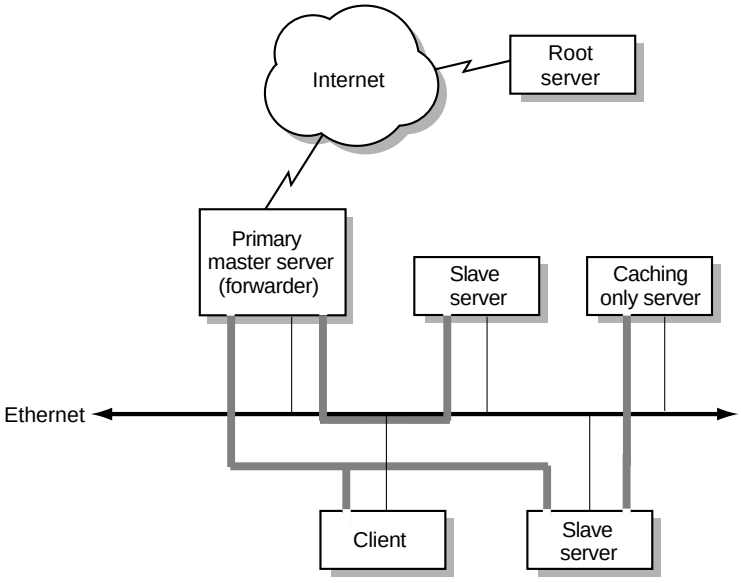
Often it is beneficial to limit the traffic to the Internet. The reason may be a slow internet connection or you are being charged by the number of packets.

Funneling DNS internet queries through one name server can reduce the number of queries going out to the internet. A name server that performs this function is a forwarder. The forwarder handles all off-site queries and in doing so builds up a cache of information which reduces the number of queries that the forwarder needs to make to satisfy a query.

Forwarder servers have access to the Internet and are able to obtain information regarding other servers not currently found in local caches. Because a forwarder server can receive requests from several slave servers, it can acquire a larger local cache than a slave server. All hosts in the domain have more information locally available because the forwarder servers have a large cache. This means that the server sends fewer queries from that site to root servers on networks outside the internet.

Figure 3–3 shows the relationship among root, primary master, slave, forwarder servers, and clients.

Figure 3–3 Relationship of Master/Forwarder Server and Slave Servers



3.7.5 Caching-Only Servers

All servers cache the information they receive for use until the data expires. The length of time a server caches the information is based on a time-to-live (TTL) field attached to the data the server receives.

Caching-only servers have no authority for any zone, and thus do not have complete information for any zone. Their database contains information acquired in the process of finding answers to clients' queries.

3.7.6 Configurations Without Internet Access

You can run the BIND service on a local network that does not have Internet access. In this configuration, the servers resolve local queries only. Any request that depends on Internet access goes unresolved.

3.7.7 Zone Transfers

Zone transfers are the process by which slave servers obtain their zone data. When a slave server starts up and periodically thereafter, the server checks to see if its data is up-to-date. It does this by polling a master server to see if the master server's zone database serial number is greater than the slave's. If so, the slave performs a zone transfer over the network.

An essential point in this polling environment is that whenever a change is made to a master server's zone database file, the zone's serial number must be incremented for the change to propagate to other servers. If the serial number does not change, slave servers do not know to perform a zone transfer.

3.7.7.1 Zone Change Notification

In addition to slave servers polling to determine the necessity for a zone transfer, BIND 8 provides a mechanism for a primary master server to notify slaves of changes to a zone's database.

When a master server determines a change has been made to a database, it will send a NOTIFY message to all the slave servers for the zone. The slave servers respond with a NOTIFY response to stop any further NOTIFY messages from the master and then queries the master server for the SOA record of the zone. When the query is answered, the slave checks the serial number in the SOA record and if indeed the serial number has changed, the slave transfers the zone.

This interrupt feature combined with polling provides a good balance between slow propagation of data due to a long refresh times and periods of inconsistent data between authority servers when zone data is updated.

3.7.7.2 Dynamic Update

DNS Dynamic Update, a BIND version 8 feature, provides for zone changes in real time; that is dynamically, without having to change a database file and then signaling the master server to re-load the zone data. Most often these changes come from other network applications like DHCP servers that automatically assign an IP address to a host and then want to register the host name and IP address with BIND.

Dynamic Update provides for:

- Adding and deleting individual resource records
- Deleting a set of resource records with the same name, class and type
- Deleting all records associated with a given name
- Specifying prerequisite records exist before adding an address record

Dynamic updates are remembered over system reboots or restart of the BIND server. Whenever the BIND server starts up, it looks for and reads the file where it logged updates (typically *domain*.db_log)) and merges the updates into its cache of zone data. If you define the logical TCPIP$BIND_SERVER_MERGE_DYNAMIC_UPDATES, the dynamic updates are also automatically written to the master zone database file that gets reloaded upon each startup of the BIND server.

3.8 BIND Server Configuration File

BIND reades information from an ASCII file called TCPIP$BIND.CONF. On UNIX systems, the filename is named.boot. This configuration file consists of statements that specify:

- The location of each BIND database file
- Global configuration options
- Logging options
- Zone definitions
- Information used for authentication.

Example 3–1 shows an example of a BIND 8 configuration file.

Example 3–1 BIND 8 Configuration File

```
//------------------------------------------------------------------
//
//           Copyright (c) Digital Equipment Corporation, 1998
//
//          TCPIP$BIND.CONF - BIND server configuration file
//
//                            IMPORTANT
//
//   This file has been generated by the TCP/IP Services for OpenVMS
//              TCPIP CONVERT /CONFIGURATION BIND command.
//
// File:          SYS$SPECIFIC:[TCPIP$BIND]TCPIP$BIND.CONF
// Date:          27-Jan-1999 14:40:02
//
//   See the DIGITAL TCP/IP Services for OpenVMS Management guide for
//   instructions on editing and using this file.
//
//_________________________________________________________________

options {
        directory "SYS$SPECIFIC:[TCPIP$BIND]";
};

zone "FRED.PARROT.BIRD.COM" in {
        type master;
        file "FRED_PARROT_BIRD_COM.DB";
};

zone "0.0.127.IN-ADDR.ARPA" in {
        type master;
        file "127_0_0.DB";
};

zone "LOCALHOST" in {
        type master;
        file "LOCALHOST.DB";
};

zone "4.33.198.IN-ADDR.ARPA" in {
        type master;
        file "4_33_198_IN-ADDR_ARPA.DB";
};

zone "." in {
        type hint;
        file "ROOT.HINT";
};
```

3.9 BIND Server Database Files

Files residing on BIND server systems contain the database of information needed to resolve BIND queries. The following sections describe the four database files used by the server:

- Master Zone File (Section 3.9.1)
- Reverse Domain File (Section 3.9.2)
- Loopback Interface Files (Section 3.9.3)
- Hints File (Section 3.9.4)

Detailed information on how to create and name these files is discussed in the *DIGITAL TCP/IP Services for OpenVMS Management* manual.

3.9.1 Master Zone File

A primary master server maintains the master zone file. This file contains:

- Start of Authority records (SOA) which specify the domain name for the zone, a serial number, refresh time, retry and other administrative information
- NS records which specify all the servers for the zone
- Address resource records (A) for each host in the zone
- MX records for mail servers
- CNAME records for specifying alias names for hosts

There is one master zone file for each zone for which the server has authority.

Example 3–2 shows a typical master zone file.

Example 3–2 Master Zone File

```
$ORIGIN ucx.ern.sea.com.
@               IN      SOA     owl.ucx.ern.sea.com. pmaster.owl.ern.sea.com.
(
                                23      ; Serial
                                600     ; Refresh
                                300     ; Retry
                                172800  ; Expire
                                43200 ) ; Minimum
;
                IN      NS      owl.ucx.ern.sea.com.
                IN      NS      condor.ucx.ern.sea.com.
;
thrush          IN      A       9.20.208.53
condor          IN      A       9.20.208.10
birdy           IN      A       9.20.208.47
                IN      MX      10 birdy.ucx.ern.sea.com.
                IN      MX      100 inet-gw-1.pa.emu.com.
                IN      MX      100 mts-gw.pa.emu.com.
                IN      MX      200 crl.emu.com.
                IN      MX      300 nester.emu.com.
seagull         IN      A       9.20.208.30
                IN      MX      10 seagull.ucx.ern.sea.com.
                IN      MX      100 inet-gw-1.pa.emu.com.
                IN      MX      100 mts-gw.pa.emu.com.
                IN      MX      200 crl.emu.com.
                IN      MX      300 nester.emu.com.
owl             IN      A       9.20.208.72
                IN      MX      10 owl.ucx.ern.sea.com.
                IN      MX      100 inet-gw-1.pa.emu.com.
                IN      MX      100 mts-gw.pa.emu.com.
                IN      MX      200 crl.emu.com.
                IN      MX      300 nester.emu.com.
peacock         IN      A       9.20.208.73
                IN      MX      10 pultdown.ucx.ern.sea.com.
                IN      MX      100 inet-gw-1.pa.emu.com.
                IN      MX      100 mts-gw.pa.emu.com.
                IN      MX      200 crl.emu.com.
                IN      MX      300 nester.emu.com.
redwing         IN      A       9.20.208.79
                IN      MX      10 redwing.ucx.ern.sea.com.
                IN      MX      100 inet-gw-1.pa.emu.com.
                IN      MX      100 mts-gw.pa.emu.com.
                IN      MX      200 crl.emu.com.
                IN      MX      300 nester.emu.com.
robin           IN      A       9.20.208.47
                IN      A       9.20.208.30
                IN      A       9.20.208.72
```

3.9.2 Reverse Domain File

For every host with an A record in the master zone file, there needs to be a way to map an IP address back to a host name. This is accomplished by using a zone file for a special domain called the IN-ADDR.ARPA domain.

The zone file for this domain contains PTR records which specify the reverse translations (address-to-host name) required for the zone. There is a IN-ADDR.ARPA zone file for each network represented in the master zone file including the loopback interface.

Example 3–3 shows the contents of a typical reverse domain file.

Example 3–3 Reverse Domain File

```
$ORIGIN 208.20.9.in-addr.arpa.
@     IN   SOA   owl.ucx.ern.sea.com. pmaster.owl.ucx.ern.sea.com.
(
                          1       ; Serial
                          600     ; Refresh
                          300     ; Retry
                          172800  ; Expire
                          43200 ) ; Minimum
;
      IN      NS      owl.ucx.ern.sea.com.
      IN      NS      condor.ucx.ern.sea.com.
;
53              IN      PTR     thrush.ucx.ern.sea.com.
10              IN      PTR     condor.ucx.ern.sea.com.
47              IN      PTR     birdy.ucx.ern.sea.com.
30              IN      PTR     seagull.ucx.ern.sea.com.
72              IN      PTR     owl.ucx.ern.sea.com.
73              IN      PTR     peacock.ucx.ern.sea.com.
79              IN      PTR     redwing.ucx.ern.sea.com.
```

3.9.3 Loopback Interface Files

The loopback interface files define the zone of the local loopback interface, known as LOCALHOST. There is a master zone file and a reverse domain file for the LOCALHOST. The resource record for this file defines LOCALHOST with a network address of 127.0.0.1. The DIGITAL TCP/IP Services for OpenVMS configuration procedure creates these two files, and calls them LOCALHOST.DB and 127_0_0.DB.

Example 3–4 shows the contents of the master zone file for the loopback interface.

Example 3–4 Loopback Interface Zone File

```
;
; BIND data file for local loopback interface (forward translation).
;
; Provided for Digital TCP/IP Services for OpenVMS.
;
$ORIGIN localhost.
@                       1D IN SOA       @ root (
                                        42              ; Serial
                                        3H              ; Refresh
                                        15M             ; Retry
                                        1W              ; Expire
                                        1D )            ; Minimum
;
                        1D IN NS        @
                        1D IN A         127.0.0.1
```

Example 3–5 shows the contents of the reverse domain file for the loopback interface.

Example 3–5 Loopback Reverse Domain File

```
;
; BIND data file for local loopback interface (reverse translation).
;
; Provided for Digital TCP/IP Services for OpenVMS.
;
$ORIGIN 0.0.127.in-addr.arpa.
@                       1D IN SOA       localhost. root.localhost. (
                                        42              ; Serial
                                        3H              ; Refresh
                                        15M             ; Retry
                                        1W              ; Expire
                                        1D )            ; Minimum
;
                        1D IN NS        localhost.
1                       1D IN PTR       localhost.
```

3.9.4 Hints File

The hints file contains information about the authoritative name servers for top-level domains. You can obtain this information from the InterNIC. However, the TCP/IP Services TCPIP$CONFIG procedure creates this file during the configuration procedure.

Example 3–6 Hints File

```
; Data file for initial cache data for root domain servers.
;
; Provided for DIGITAL TCP/IP Services for OpenVMS.
;
; <<>> DiG 8.1 <<>> @192.5.5.241
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN
;
;; ANSWER SECTION:
.             6D IN NS H.ROOT-SERVERS.NET.
.             6D IN NS B.ROOT-SERVERS.NET.
.             6D IN NS C.ROOT-SERVERS.NET.
.             6D IN NS D.ROOT-SERVERS.NET.
.             6D IN NS E.ROOT-SERVERS.NET.
.             6D IN NS I.ROOT-SERVERS.NET.
.             6D IN NS F.ROOT-SERVERS.NET.
.             6D IN NS G.ROOT-SERVERS.NET.
.             6D IN NS J.ROOT-SERVERS.NET.
.             6D IN NS K.ROOT-SERVERS.NET.
.             6D IN NS L.ROOT-SERVERS.NET.
.             6D IN NS M.ROOT-SERVERS.NET.
.             6D IN NS A.ROOT-SERVERS.NET.
;
;; ADDITIONAL SECTION:
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
;
;; Total query time: 608 msec
;; FROM: ucx.ern.sea.com to SERVER: 192.5.5.241
;; WHEN: Mon May 18 15:26:19 1998
;; MSG SIZE  sent: 17  rcvd: 436
```

3.10 BIND Resolver

The BIND resolver is a set of routines that is linked into each network application needing DNS name resolution services. The resolver formulates one or more queries based on the resolver's configuration and information supplied by network applications and sends the queries to a server to obtain an answer.

You can configure the following resolver features:

- Enable or disable the use of a hosts database file for name resolution in addition to using a name server
- Define the default domain

- Specify a domain search list
- Specify the name servers to query
- Specify a transport (either UDP or TCP)
- Specify a timeout interval for requests.

The *DIGITAL TCP/IP Services for OpenVMS Management* manual contains information on how to configure the resolver.

3.10.1 Default Domain

The default domain is the domain in which the client host resides. When resolving a query when just the hostname is supplied, the resolver appends the default domain to the host name and then processes the query. This is a convenience for the user. It saves typing a fully qualified domain name.

3.10.2 Search List

The search list is also another convenience for the user. The default search list is derived from the default domain and is applied if the user enters a non-fully qualified domain name.

3.10.3 Name Servers

You can configure the resolver to query any name server including the local host and you can specify a maximum of three name servers. The resolver queries each name server in the order listed until it receives an answer or times out.

4

Network File System Concepts

The Network File System (NFS) is a facility for sharing files in a heterogeneous environment of hardware platforms, operating systems, and networks. NFS allows users to access files distributed across a network in such a way that remote files appear as if they reside on the local host. NFS has become a standard for the exchange of data between machines running different operating systems.

Another way the NFS protocol achieves portability between different machines, operating systems, network architectures, and transport protocols is through the use of Remote Procedure Calls (RPCs) and the External Data Representation (XDR), two network programming constructs that handle reliability issues. For more information about RPCs and XDR, see the *DIGITAL TCP/IP Services for OpenVMS ONC RPC Programming* manual.

Using NFS is simple. Configuring and implementing NFS, however, are more complex. A summary of NFS concepts and considerations is included in this chapter, but you should refer to the *DIGITAL TCP/IP Services for OpenVMS Management* guide for detailed configuration and implementation information.

Specific topics covered in this chapter include:

- Overview of NFS (Section 4.1)
- NFS protocol (Section 4.2)
- Client and server software (Section 4.3)
- Related databases (Section 4.4)
- PC-NFS daemon (Section 4.5)
- UNIX-OpenVMS differences accommodated by NFS, including:
  - Directory hierarchies (Section 4.6.1)
  - File specifications (Section 4.6.2)
  - Linking files (Section 4.6.3)
  - File structures (Section 4.6.4)
  - File ownership (Section 4.6.5)
  - File protections (Section 4.6.6)
  - UNIX style file system on hosts (Section 4.6.7)

4.1 Overview

NFS was originally designed for UNIX systems, so it follows UNIX conventions for files, file types, file names, file ownership, user information, and so forth. NFS in an OpenVMS environment must accommodate the differences between UNIX and OpenVMS in such a way that when an OpenVMS user accesses a file from a UNIX system, the file looks like an OpenVMS file. Conversely, when a UNIX user accesses a file from an OpenVMS system, it looks like a UNIX file.

In a local environment, file systems reside on physical disks directly connected to the system. NFS provides a distributed environment where the users on one system can access files that physically reside on disks attached to another networked system. These files are called **remote file systems**.

Remote files are made accessible to local users through the process called **mounting**. After a file system or the entire disk is mounted, users access files through the operating system's services. A mount operation makes a remote file system, or a subtree within it, part of the local file system.

Some general characteristics of NFS include the following:

- **Client/Server Environment** — NFS software consists of client and server software.
  - The NFS client requests resources provided by NFS servers. Local users on client systems access files that reside on systems running NFS server software.
  - The NFS server makes particular file systems available to users on NFS client systemss. The file systems are called **exported file systems**.

  When applications request file operations, such as open, store, or retrieve, the operating system passes the request to either the local file system or the NFS client. When it receives a request, the NFS client contacts the appropriate NFS server on the remote host to perform the requested operation.
- **Transparent File Access** — After a mount is complete, users can access the files they want without knowing the name or network address of the host where the files reside. To the user, there seems to be no difference between reading or writing a file on a local disk and reading or writing a file on a disk on a remote host.
- **Easily Extensible** — As a distributed service, NFS is designed to allow integration of new software without disturbing the existing software environment. To accomplish this, NFS provides a network service rather than a network operating system. NFS does not extend the underlying operating system, but instead offers a set of procedures for data exchange.
- **High Performance** — The flexibility of NFS permits configuration for a variety of cost and performance tradeoffs. For example, you might configure servers with high-performance disks and clients with no disks. This environment may yield better performance at lower cost than many systems with small inexpensive disks.

  Also, it is possible to distribute a file system across many servers and get the benefit of multiprocessing along with transparency.

  For read-only files, copies can be kept on several servers to avoid bottlenecks.

- **Idempotent Operations** — The NFS server is **stateless**, which means it can function correctly without needing to maintain protocol state information about any of its clients. The NFS client, on the other hand, is **stateful**; it needs to be able to detect a server failure and rebuild the server's state when it comes back up or cause client operations to fail. Working with stateless NFS servers, an NFS client just retries a request until the server responds; it does not need to know that the server has crashed or that the network temporarily went down.

  The basic way to simplify recovery is to make operations as **idempotent** as possible. This means that operations can be requested more than once. Completing the same operation more than once with the same arguments produces the same outcome. This simplifies recovery after client, server, or network failures.

Table 4–1 defines basic NFS terms.

Table 4–1 Basic NFS Definitions

Term	Definition
File system	Top-level directory, its lower-level directories, and all the files in those directories. The top-level is called the master file directory (MFD) in OpenVMS file systems and the root in UNIX style file systems.
Container file system	File system on an OpenVMS host that has a UNIX style directory structure and UNIX style file attributes. Created by the DIGITAL TCP/IP Services for OpenVMS software.
UNIX style file system	Same as container file system.
OpenVMS file system	File system with an OpenVMS directory structure and OpenVMS file attributes.
UNIX file system	File system on a UNIX host with a UNIX directory structure and UNIX file attributes.
Proxy	Record in the proxy database that gives a remote user access to local file systems or a local user access to remote file systems.
Disk	Physical device, or volume, on which a file system resides.
Mapping	Process that makes local OpenVMS disks and container file systems accessible to an NFS client users, thus identifying them as "NFS file systems."
Exporting	Adding a file system name to the export database. This allows an NFS client to mount a mapped local file system.
Mounting	Process that makes physically remote file systems, directories, or individual files available to NFS clients.
Mount point	Directory location of the mounted file system.

4.2 The NFS Protocol

The NFS protocol specification is defined in several RFCs:

- RFC 1014 and RFC 1057 define the format for NFS messages exchanged between client and server.
- RFC 1094 defines the procedures to request file operations and return results.

The protocol provides for stateless operations where:

- NFS servers retain no client state information.
- Each NFS request contains all the information required for completion.
- The server does not incur performance overhead associated with maintaining state.

Stateless servers provide robustness when there are client, server, or network failures. If a client fails, an NFS server need not take action to continue normal operation. If a server or the network fails, NFS clients continue to attempt completing NFS calls until the server or network is fixed. This robustness can be important in a complex network of heterogeneous systems that is not under the control of a single network manager.

4.3 NFS Client and Server Software

NFS consists of the NFS server and the NFS client software. The NFS server is implemented as a daemon, waiting for requests from clients. The NFS server does not retain the state of the NFS client.

The NFS server daemon is multithreaded; it processes multiple NFS client calls, in parallel. The NFS server daemon is also an event-driven, asynchronous process. Each NFS client call contains all the information necessary to complete the request.

The NFS client software implements a state mechanism that maintains all information required for processing client requests. Each client operation can be requested more than once and contains all the information necessary to complete the request. This model presumes no file open and close requests because these require saving the state of the object and that the server write data to the disk before returning the reply message to the user.

When mounting a remote file system, an NFS client sends a message that makes the remote file system part of the local file directory. The remote host redirects operations that access files on the remote file system to the NFS client software. The NFS client and NFS server then exchange messages.

4.4 Related Databases

NFS servers and NFS clients use the proxy database to provide users access to remote file systems. The NFS server also uses the export database. The export database includes file system names.

All OpenVMS nodes running the NFS server software can share the proxy and export databases. Table 4–2 shows the databases used by the NFS server and NFS client.

Table 4–2 Databases Used by NFS Server and Client

Entity	Database	File Name	Logical Name
Server	Export database	TCPIP$EXPORT.DAT	TCPIP$EXPORT
Server	Proxy database	TCPIP$PROXY.DAT	TCPIP$PROXY
Client	Proxy database	TCPIP$PROXY.DAT	TCPIP$PROXY

4.5 The PC-NFS Daemon

The PC-NFS daemon (PC-NFSd) provides authentication and printing services for PCs.

The PC-NFS daemon provides the following functions required for printing:

- Associates a DOS printer device name with an OpenVMS print queue on the server host where the PC-NFS daemon is running.
- Prints a file to the associated queue.

4.6 UNIX and OpenVMS Differences Accommodated by NFS

NFS accommodates numerous key differences between UNIX and OpenVMS to make user interaction between the two operating systems appear transparent. These differences are discussed in the remainder of this chapter and include:

- Directory hierarchies
- File specifications, including absolute and relative file specifications, file names, case sensitivity, file types, and version numbers
- Linking files
- File structures
- File ownership
- File protection
- UNIX style file system on DIGITAL TCP/IP Services for OpenVMS hosts

4.6.1 Directory Hierarchies

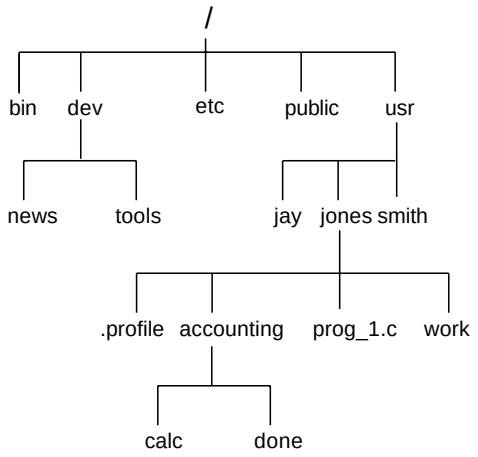
Table 4–3 lists differences between the OpenVMS and UNIX directory hierarchies.

Table 4–3 Directory Hierarchy Differences

UNIX	OpenVMS
May reside on multiple volumes.	Resides on one volume having one root above all directories on the volume.
Devices not included in file specification.	Devices included in file specifications.

Figure 4–1 shows a UNIX directory. The UNIX hierarchy appears as one tree that can be located on more than one device.

Figure 4–1 UNIX Directory Hierarchy



4.6.2 File Specifications

An OpenVMS file specification is limited to eight directory levels and has the following format:

device:[*directory.subdirectory*] *filename.type*;*version*

The following delimiters separate the file specification components:

- The colon (:) separates the device from the directory.
- Brackets ([ ]) or angle brackets (<>) enclose the directory and any subdirectories.
- A period (.) separates directories from subdirectories and separates the file name from the file type.
- A semicolon (;) or period (.) separates the file type from the version number.

A UNIX file specification, called a path name, has the following format:

/directory/directory/filename

The slash (/) is the only delimiter that the UNIX file specification format uses. The first slash in a UNIX file specification represents the root directory. Subsequent slashes separate each component in the file specification (the directories from the other directories and the file name). In theory, there is no limit to the number of directory levels in a UNIX file specification, whereas an OpenVMS file specification is limited to eight directory levels.

Absolute and Relative File Specifications

OpenVMS and UNIX both have two types of file specifications or path names: absolute path name and relative path name.

On UNIX systems, absolute path names use the entire directory path that leads to the file, beginning with the root, which is represented by an initial slash. The `root` directory is the first directory in the file system. All other files and directories trace their ancestry back to the root. Relative path names begin the directory path with the current working directory and exclude the current working directory name in the path name. There is no initial slash in a relative name.

For example, using Figure 4–1, a UNIX absolute path name would be `/usr/jones/accounting/calc` whereas the relative path name for the file `calc` in the current directory `/usr/jones` is `accounting/calc`.

A UNIX path name can have a maximum of 1024 characters; an OpenVMS file specification can have a maximum of 255 characters.

File Names

A complete OpenVMS file name specification includes the file name, the file type, and an optional version number, from left to right in that order. The file name and file type can each have up to 39 characters and are separated with a period (for example: FILE_NAME.TXT;1). The valid characters in a file name or type include: A–Z, 0–9, underscore (_), hyphen (–), and dollar sign ($). Version numbers (following a semicolon) are decimal numbers from 1 to 32767; they differentiate versions of the same file.

A UNIX file specification generally contains up to 1024 characters, with each element of the path name containing up to 255 characters. Some older versions of the UNIX operating system limit the size of one element to 14 characters or have other limits that you can change if you recompile the kernel.

In theory, you can use any ASCII character in a UNIX path name except for the slash (/) and null characters. For example, a file name of `report.from.january_24` is valid. However, you should avoid using some characters (such as the pipe (|) character) because these characters can have special meaning to the UNIX shell.

Case Sensitivity

The OpenVMS file system is not case sensitive. However, the UNIX operating system treats upper and lowercase characters as different characters.

For example, on a UNIX system the following filenames represent three different files; however, on an OpenVMS system they represent one file.

- CHAPTER_ONE
- Chapter_One
- chapter_one

File Types

File types are important in OpenVMS file name identification. The file type usually describes the kind of data in the file. For example, a text file typically has a file type of .TXT. Directories all have file types of .DIR;1.

Although UNIX systems do not use file types, UNIX does use certain naming conventions that resemble OpenVMS file types. For example, file names ending in *.txt* are text files. UNIX directories do not have special file types.

Version Numbers

Every OpenVMS file has a version number. When a file is created, the system assigns it a version number of 1. Subsequently, when a file is edited or additional versions of that file are created, the version number automatically increases by 1. Therefore, many versions of a file with the same file name can exist in the same directory.

The UNIX file system does not support automatic creation of multiple versions. In most cases, if you edit a UNIX file, the system saves only the most recently edited copy.

4.6.3 Linking Files

A link is a directory entry that refers to a file or a directory. On UNIX systems, files cannot exist without links and a file can have multiple links. On an OpenVMS system, files can exist without any links.

There are two kinds of links: hard links and symbolic links.

Hard Links

A hard link to a file is indistinguishable from the original link established when the file was created. These additional links allow users to share the same file under different path names. A hard link cannot span file systems.

On UNIX systems, any changes to the file are independent of the link used to refer to the file. The UNIX system maintains a count of the number of links to each file. If removing a link results in the link count becoming zero, the file is deleted. A file cannot be deleted except by removing all of its links.

The OpenVMS system also allows you to perform a similar function with the SET FILE/ENTER and SET FILE/REMOVE commands. The OpenVMS operating system does not maintain a count of links to a file. As a result, you can delete a file, and not delete its links.

Note

The UNIX operating system cannot distinguish the order in which links are created. Therefore, all links to a file are of equal value.

Symbolic Links

A symbolic link is a type of file that contains the name of the file to which it is connected. Symbolic links provide a path to the original file.

A UNIX symbolic link can span file systems. Unlike the hard link, the symbolic link does not maintain a link count. Symbolic links can exist after the file has been deleted. However, an error is returned if the symbolic link file is accessed after the file it names is deleted.

Note

OpenVMS file systems do not support symbolic links.

4.6.4 File Structures

The OpenVMS file system supports three file organizations: indexed, relative, and sequential. OpenVMS also supports the following record formats and record attributes:

- Fixed length
- Variable length
- Variable with fixed-length control (VFC)
- Stream (including STREAM_LF and STREAM_CR)
- Undefined
- Carriage return carriage control
- Fortran carriage control

- VFC carriage control

The UNIX file system supports only byte streams. The records in UNIX text files have the same format as the OpenVMS Record Management Services (RMS) STREAM_LF record format.

A DIGITAL TCP/IP Services for OpenVMS NFS server dynamically converts sequential files that are not streams (STREAM_LF or STREAM_CR) when read by NFS clients.

Note

NFS clients have read-only access to non-stream files.

File Size Discrepancies

Data conversion to STREAM_LF format may change the size of a file because of differences in the record formatting overhead. Therefore, the product's NFS server does not know the correct size of a non-stream file until it has dynamically converted the file at least once.

For NFS client requests that require the file size to be returned, but do not require reading the file (for example, ls -l), the product's NFS server returns an estimate of the size based on the unconverted size. The first time the server needs to read the file, it computes the correct converted file size and caches the information for future use.

4.6.5 File Ownership

The OpenVMS and UNIX operating systems use different mechanisms for file ownership.

OpenVMS File Ownership

The OpenVMS operating system controls file ownership and protection through a user identification code (UIC).

A UIC is a 32-bit value that consists of a 14-bit group number, a 16-bit member number, and 2 reserved bits. Each user of the system has a UIC defined in the SYSUAF file. Access to objects depends on the relationship between the UIC of the accessing process and the UIC of the object (the file or directory).

OpenVMS controls file access through an access control list (ACL). You can deny or grant read, write, execute, delete, and control access to a user or group of users who have the identifier specified by the ACL. For additional ACL information, refer to the OpenVMS documentation set.

Because the NFS protocol does not provide ACL support, the NFS client is not aware of an ACL applied to the file by the NFS server. Therefore, the NFS client cannot use an ACL to control file access. Only the NFS server can use an ACL when accessing files. Other access control is determined through standard file protections. Attempts to implement access control through NFS software cause ambiguity. Therefore, an ACL is only used to deny access to OpenVMS files.

UNIX File Ownership

The UNIX operating system controls access to files with user identification (UID) and group identification (GID). Some versions of UNIX use a 16-bit UID and GID; others may use different values. For example, DIGITAL UNIX and NFS use 32-bit UIDs and GIDs.

4.6.6 File Protections

The OpenVMS and UNIX operating systems use similar file protection schemes as shown in Table 4–4.

Table 4–4 File Protection Comparison

Mechanism	OpenVMS File System	UNIX File System
User classifications	SYSTEM (S) OWNER (O) GROUP (G) WORLD (W) Classification depends on the relationship between the UID of the accessing process and the object.	user (u) — The user has a matching UID group (g) — The group has a matching GID other (o) — Any other user System category is not used; system administrators always have access to UNIX files.
Protection levels	READ (R) WRITE (W) EXECUTE (E) (controls file execution and directory search access) DELETE (D)	read (r) write (w) — controls delete access; if a file has write protection enabled, you can delete it execute (x) — controls file execution and directory search access
Syntax	`s:rwed,o:rwed,g:rwed,w:rwed`	`rwxrwxrwx` The protection levels are divided into groups of three characters, using the following format: • First three characters: protection levels for the owner • Second three characters: protection levels for the group • Last three characters: protection levels for all other users

4.6.7 UNIX Style File System on OpenVMS Hosts

The DIGITAL TCP/IP Services for OpenVMS software allows you to create a logical UNIX style file system on an OpenVMS host. Remote UNIX hosts that have NFS software can then access this file system. When a remote UNIX system accesses files, these files conform to the UNIX file system rules, not OpenVMS rules. This ensures that existing UNIX applications will work without change. For information about creating a UNIX file system on an OpenVMS host, refer to the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

An NFS server on a DIGITAL TCP/IP Services for OpenVMS host can support multiple logical UNIX style file systems. A logical UNIX style file system is organized as a tree with a single root node, non-leaf nodes being directory files, and leaf nodes being either directory or regular data files. The logical UNIX style file system resides on a Files-11 formatted disk and is represented as a set of Files-11 files called a container file system (CFS).

The UNIX style file names and attributes are catalogued in the container file, one of the files in the CFS. The container file also has a representation of the UNIX style directory hierarchy and a pointer to the data file for each file name. In addition to its UNIX style name, each file in the CFS has a valid Files-11 file name assigned by the system.

An OpenVMS directory exists for each UNIX directory stored in the container file. All files cataloged in a UNIX directory are also cataloged in the corresponding OpenVMS directory. However, the UNIX directory hierarchy is not duplicated in the OpenVMS directory hierarchy.

Each UNIX style file is represented as an OpenVMS data file. Therefore, OpenVMS utilities, such as BACKUP, can use standard methods to access these files.

5

Planning For Your TCP/IP Environment

This chapter describes how to plan for the installation of your TCP/IP network. It includes information about planning for:

- A Network Interface (Section 5.1)
- Routing (Section 5.2)
- BIND (Section 5.3)
- DHCP or BOOTP (Section 5.4)
- Serial Lines (Section 5.5)
- NTP (Section 5.6)
- SNMP (reference>(plan_snmp)
- User Accounts and Proxy Identities (Section 5.8)

Refer to the *DIGITAL TCP/IP Services for OpenVMS Management* guide for details about managing other services.

5.1 Network Interface

The SYS$MANAGER:TCPIP$CONFIG procedure automatically detects the installed network interfaces on the installation system. After displaying a list of the installed network interfaces, TCPIP$CONFIG queries for more information on each interface.

Use the worksheet in Figure 5–1 to record network interface parameter information for your system.

Figure 5–1 Network Interface Configuration Worksheet

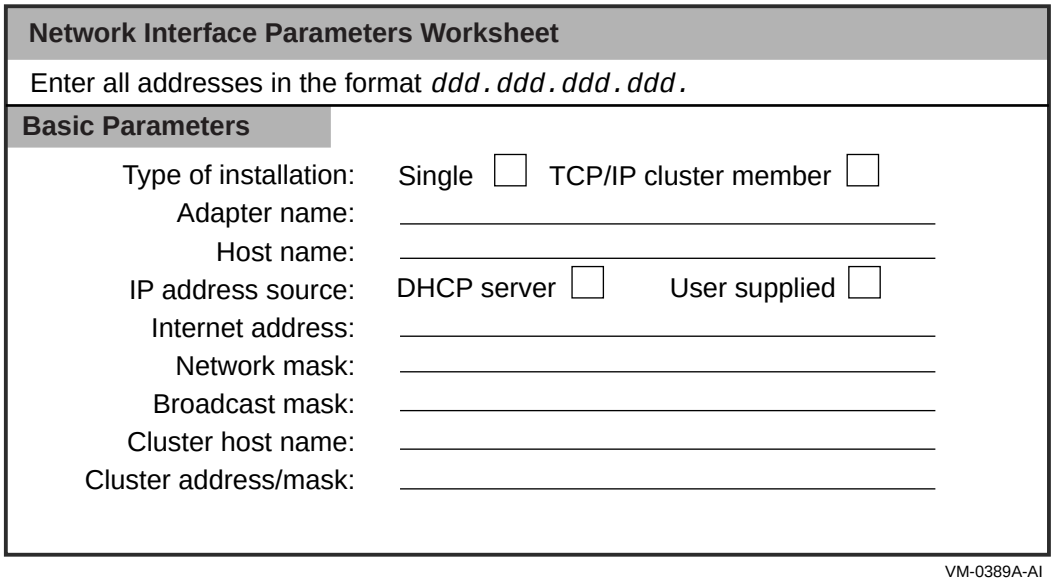


Table 5–1 describes the information necessary for configuring a network interface.

Table 5–1 Network Interface Parameters

Parameter	Enter on Worksheet...
Installation type	Check single if the installation system is not participating in a cluster. Check cluster member, if the installation system is to be configured as a TCP/IP cluster.
Adapter name	Enter the interface name for the communication controller. For example, DE0, EZ0, SL0, PP0
Host name	Enter the fully qualified host name assigned to your system. You may have to ask your network manager for a unique host name.
IP address source	Check whether DHCP or the user assigns the IP address.
Internet address	Enter the Internet Protocol (IP) address, if user supplied.
Network mask	If your network has subnets, enter the network mask for the local network. The network mask is the same for all systems on the local network.
Broadcast mask	Enter YES if you want the installation system to receive all broadcast messages.
Cluster host name	Enter the host name to associate with each interface in the cluster.
Cluster address/mask	Enter the IP address and the network mask associated with the cluster.

5.2 Routing

If the hosts on your network need to communicate with computers on other networks, a route through a gateway must be defined. All hosts and gateways on a network store information about routes in routing tables. Routing tables are maintained in both dynamic and permanent memory.

You can define routes manually (static routing) or you can enable routing protocols that exchange information and build routing tables based on the information exchanged (dynamic routing).

5.2.1 Static Routing

Because static routing requires manual configuration, it is most useful when the number of gateways is limited and where routes do not change frequently. For information on manually configuring routing, see the *DIGITAL TCP/IP Services for OpenVMS Management* manual.

5.2.2 Dynamic Routing

Complex environments require a more flexible approach to routing than a static routing table provides. Routing protocols distribute information that reflect changing network conditions and update the routing table accordingly. Routing protocols can switch to a backup route when a primary route becomes unavailable and can determine the best route to a given destination.

Dynamic routing tables use information received by means of routing protocol updates; when routes change, the routing protocol provides information on the changes.

Routing daemons implement a routing policy, that is, the set of rules that decide which routes go into the routing table. A routing daemon writes routing messages to a routing socket causing the kernel to add a new route, delete an existing route, or modify an existing route.

The TCP/IP Services for OpenVMS product implements two routing daemons: the Routing Daemon (ROUTED) and the Gateway Routing Daemon (GATED). The following sections provide more information.

5.2.2.1 Routing Daemon (ROUTED)

This daemon (pronounced route-dee) supports only the Routing Information Protocol (RIP). When ROUTED starts, it issues routing update requests then listens for responses. A system configured to supply RIP information responds to the request with an update packet. The update packet contains destination addresses and routing metrics associated with each destination. After receiving a RIP update, ROUTED uses the information to update its routing table.

To configure dynamic routing with ROUTED, see Section 5.2.2.2.

5.2.2.2 Gateway Routing Daemon (GATED)

This daemon (pronounced gate-dee) supports interior and exterior gateway protocols. It obtains information from several routing protocols and selects the best routes based on that information. You can configure GATED to use one or more of the following protocols:

- Routing Information Protocol (RIP) Versions 1 and 2

  RIP is a commonly used interior protocol that selects the route with the lowest metric (hop count) as the best route.

- Open Shortest Path First (OSPF)

  Another interior routing protocol, OSPF is a link-state protocol (shortest path first). Use OSPF in complex networks with many routers.

- Exterior Gateway Protocol (EGP)

  EGP exchanges reachability information between autonomous systems. An autonomous system is usually defined as a set of routers under a single administration, using an interior gateway protocol and common metric to route packets. Autonomous systems use exterior routing protocols to route packets to other autonomous systems.

- Border Gateway Protocol (BGP)

  Like EGP, BGP exchanges reachability information between autonomous systems but supports nonhierarchical topologies. Use BGP to specify path attributes that provide more information about each route. Path attributes can include, for example, administrative preferences based on political, organizational, or security considerations.

- Router Discovery

  Use this protocol to supplement a statically configured default router. Router discovery informs hosts of the availability of local routers.

Use the worksheet in Figure 5–2 to record routing parameter information for your system.

Figure 5–2 Routing Configuration Worksheet

Routing Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Type of routing: Static ☐ Dynamic ☐
Static default route: _____
Type of dynamic routing: ROUTED ☐ GATED ☐

ROUTED

Supply dynamic routing information Yes ☐ No ☐

GATED

Configuration file: _____
Routing protocols: RIP ☐ OSPF ☐ EGP ☐ BGP ☐ Hello ☐
Router discovery: Yes ☐ No ☐

VM-0390A-AI

Table 5–2 describes the routing parameters.

Table 5–2 Routing Parameters

Parameter	Enter on Worksheet...
Routing type	Check the type of routing: STATIC or DYNAMIC
Default route	If using STATIC routing, enter a default route (mandatory). You will need to add this by using the TCPIP SET ROUTE command.
Router	If using dynamic routing, check either ROUTED or GATED.
ROUTED:	
Supply dynamic routing	If using ROUTED, check YES if you want the installation system to send dynamic routing information to peers.
GATED:	

Table 5–2 (Cont.) Routing Parameters

Parameter	Enter on Worksheet...
Interior/Exterior Protocol	Check one or more routing protocols to be configured on the installation system.
Configuration file name	If using GATED, enter the name, location, and date of the current GATED configuration file.

5.3 BIND

This section describes planning steps for implementing a BIND server on a DIGITAL TCP/IP Services for OpenVMS host:

Consider configuring your network to use the BIND service if you have:

- A small configuration with incoming connections from the Internet.
- A large configuration connected to the Internet.
- Any configuration that requires frequent and highly reliable access to remote hosts.

If you have a small local network that requires infrequent access to remote hosts or is not connected to the Internet, consider using a local host database instead of a BIND server.

5.3.1 Planning a Domain Hierarchy Strategy

The effectiveness of your BIND service depends on careful planning of your domain hierarchy. As you plan the domain hierarchy, you need to do the following:

1. Understand the existing domain hierarchy in your organization or company. Contact your system administrator to find the person responsible for the existing domains. If you want your domain on the public network, you need to get the correct domain registration from the InterNIC Registration Services (see the InterNIC web site. `http://internic.net/`).
2. If you plan to join an existing domain, negotiate with the administrator of the upper-level domain to ensure that you create an acceptable hierarchy.
3. If you plan to administer zones for your hosts and servers, identify the owners of the parent zone to which you will be a subzone.

When designing your domain hierarchy, select the scheme that suits the needs and preferences of your organization. A common design strategy is to base the domain hierarchy on functional areas of an organization or on geographic areas of a network. For example, you could divide your domain hierarchy into geographic zones used mainly by a group of users concentrated in geographic areas. Similarly, you could create a functional zone with names used mainly by people in a particular branch of an organization.

Table 5–3 describes some of the benefits of using each hierarchy scheme.

Table 5–3 Functional and Geographic Hierarchies

Consider This Hierarchy:	If You Have:	To Implement:
Functional	An organization consisting of: • A well-established and stable functional structure. • Several independently functioning units with their own resources.	Use your company's organizational chart as a template for designing this hierarchy.
Geographic	Hosts that are: • Organized and managed by site or geographic area. • Grouped in a specific geographic area together with other hosts that usually communicate with each other.	Use a map as an aid in naming servers. Place slave servers in the same geographic area. Off-site slave servers have availability benefits.

5.3.1.1 Finding Existing BIND Service Information

When planning your domain hierarchy, it may be useful to look at information for existing domains and name servers. The TCP/IP Services for OpenVMS product supports the TCPIP$NSLOOKUP utility, which allows you to retrieve the following information:

- Host names and addresses on the local domain
- Host names and addresses on remote domains
- Host names that have mail exchanger (MX) records
- Name servers for a specific zone

5.3.1.2 Domain Hierarchy Guidelines

Consider the following domain hierarchy guidelines:

- Limit creation of domains to three or four levels.

  In very large networks, people at individual sites create and own lower-level domains used mainly by a specific group of local users. In such cases, users create as many levels of domains as they deem necessary and convenient to manage.

- You can divide your domain hierarchy into several zones. Compaq recommends that you divide your domain hierarchy into multiple zones if your network is large and growing. Multiple zones offer the following management advantages over a single zone structure:

  – Creates a greater opportunity to distribute information in the network, decreasing the processing load on any one system.

  – Decreases the probability of duplicate names (for example, `aero.sales.compaq.com` and `aero.dev.compaq.com` are unique).

  – Eases the delegation of administrative responsibility for domains, subdomains, and zones.

- Limit domains and zones to a manageable size.

  Zones require additional network resources to administer. Also, long domain names are difficult to remember.

  Domains and zones have a practical size limit based on manageability. Factors to consider include the following:

  – Identify how widely zone information must be copied.

    A zone containing many domain names may need several slave servers in different places to locate the names close to users. Propagating updates to slave servers adds overhead, especially across WAN links. For example, if you have three sites and 100,000 domain names, and all of the sites need those domain names, keep them in one zone. Conversely, if specific domain names are mostly used at one site, save overhead by locating those domain names in a zone for that site.

  – Anticipate growth.

    Design a domain that allows horizontal expansion (more subdomains directly under the upper-level domain) and avoids vertical expansion (more levels of subdomains). This design helps limit the number of subdomain levels and makes the domain easier to manage, even as it grows.

5.3.1.3 Deciding to Create Zones

Table 5–4 lists some criteria to use when deciding if you want to create new zones.

Table 5–4 Joining or Creating a Zone

Consider:	If:
Joining an existing zone	A suitable zone already exists in the domain space.
	You plan to manage a small number of hosts.
	You expect little change or growth in your domain space.
	You expect a low demand for host and address lookups.
	The current domain administrator is willing to accept the extra work.
Creating new zones	You are creating the initial domain hierarchy in your organization.
	You want control over your domain space.
	You plan to manage a large number of hosts.
	You expect a lot of change or growth in your domain space.
	You expect a high demand for host and address lookups.

Whether you decide to join an existing zone or create a new one, identify the proper parent zone and get the owner's agreement to your approach.

If you decide to create zones, you need to:

- Decide which hosts you want in a particular zone.

- Select the servers for the zone.

5.3.2 Developing Domain Naming Conventions

After you decide how to structure your domain hierarchy, establish domain naming conventions. Table 5–5 lists domain naming conventions and their supporting reasons.

Table 5–5 Domain Naming Conventions

Convention:	Supporting Reasons:	Example:
Use domain names that match the BIND hierarchy.	Your naming policy and the BIND hierarchy are likely to evolve at the same time.	If you have existing host names (such as: host1, warrick, and marcom), you may want to give them geographic domain names (such as: albany, warrick, hartford) or functional department names (such as: eng, prodmgt, marcom).
Choose domain names that are not likely to change.	Creates a more stable naming hierarchy because it is difficult to change existing labels, especially at higher domain levels. Also, a change in a domain name effects all applications that use them and users who memorized the name of a resource or created an abbreviation.	If you use a functional model, derive domain names from business functions, not current titles. For example, consider using `sales.compaq.com`, `admin.compaq.com`, and `eng.compaq.com` to store names used by the sales, administrative, and engineering branches of the ABC organization.
Use a multi-level domain naming strategy to manage large domains.	Creates a complex, but more manageable domain than a large, single-level domain.	If you have a large domain, you could name upper-level domains after cities such as, `nyc.compaq.com`, `paris.compaq.com`, and `geneva.compaq.com`, and lower-level domains based on site codes or some other more specific geographic name.
Select domain names that are short and describe the resource represented.	These names are easier to remember.	For example, you could use `ftp.aero.dev.compaq.com` as the domain name of the FTP access point used by the ABC's aerospace development division.

5.3.2.1 Case Sensitivity

The BIND service preserves the case of names as they are entered. Lookups, however, are case insensitive, so it is not possible to create two names that differ only in their case. For example, requests to look up `mynode.lkg.compaq.com`, `MYNODE.lkg.compaq.com`, and `MyNode.lkg.compaq.com` would all produce the same result. If someone attempted to create entries in the zone database files for all three domain names, you would have multiple records for the same name.

5.3.2.2 Planning Domain Names for Reverse Lookups

The IN-ADDR.ARPA domain names include up to four domain labels in addition to the IN-ADDR.ARPA suffix. Each label represents one octet of an Internet address, in reverse order. For example, if your host has Internet address 37.20.16.08, its domain name for reverse translation would be: 08.16.20.37.IN-ADDR.ARPA.

Use the following guidelines:

- Make sure that each address has a record in the zone file that provides the canonical name of the host (this is the PTR record). Some applications do not function properly unless they can find a host name.
- When you receive a block of addresses to use, ask the organization that supplied them to you about any required actions you need to take regarding reverse translation domain servers.
- If you use a small number of addresses and part of a set of addresses that are owned by a service provider, the service provider typically ensures that reverse translations will be handled properly.
- If you receive a block of addresses from the InterNIC, you are responsible for ensuring reverse translations servers exist.

5.3.3 Defining Zone Contents and Administration

Deciding which domain names and hosts belong in a zone is a simple task if you planned your domain hierarchy carefully. Your zone will contain domain names, hosts, and servers. The master zone file, maintained on master servers, contains all the information for the zone.

If you decide to create zones, consider an overall administration scheme. Your plan for who will use and manage names can have a strong influence on zone structure. For example, the upper-level domains should be stable, widely known, and limited in content. Only a few trusted people should be able to create or modify their contents.

Typically, an individual acts as the technical/zone contact for zones. This person is concerned with the technical aspects of maintaining the BIND server and resolver software and the data files. The technical/zone contact keeps the BIND server running and interacts with technical people in other domains and zones to solve problems effecting the local domain.

5.3.4 Selecting Servers

Your main goals in choosing hosts for servers should be to achieve availability of data, reliability, and optimum performance. If you create your own zones, you must configure at least one master and one slave server for each zone.

Consider configuring servers even if you are not creating your own zones. If you configure a slave server for the zones (forward and reverse) where your hosts are members and point your hosts to that slave server, the BIND service will continue to work for local names even if you lose your link to the outside world.

5.3.4.1 Server Selection Guidelines

Study your network and keep in mind the following guidelines to help you achieve your goals:

- Plan to configure at least three servers for each zone.

  Although each zone requires only one master server, use at least three servers to increase reliability and performance. Use a master and slave server on the same network, and another slave server outside of the network. Ensure that this other slave server is on a different LAN or in another building. This provides an alternate source of information if a server is temporarily unavailable. This scheme also provides a backup of BIND service data if the master server becomes permanently unavailable.

- Choose stable hosts.

  BIND servers should reside on stable systems that experience minimal downtime and restart quickly. It is important that a BIND server be one of the first systems available in the network, because client applications on the network are limited to the information in caches until a BIND server is available.

- Estimate disk space, CPU, and memory requirements.

  In small zones, the impact of the BIND service on timesharing systems should be minimal. The impact depends partly on the number of entries in the database files, the frequency of use of those names, and the load placed on the system by other applications.

5.3.4.2 Selecting Master Servers

The master server is the authority, or best source of information, for one or more zones. You need one master server for each zone in your domain hierarchy.

Every time a host changes addresses, you need to update the zone file and reverse domain zone files on the master server. If your zone has many hosts, consider dividing the zone into separate subzones, to balance the administrative work.

A master server can also be a master or a slave server for other zones that exist in a contiguous or non-contiguous part of the name space.

5.3.4.3 Selecting Slave Servers

Your strategy for configuring slave servers is especially important in the initial stages of BIND operation. Once a server establishes a cache of frequently used names, the server will rarely need to locate a copy of the information it needs. However, well-planned copying of zone files can make the server's process of learning about names in a large network easier and more efficient.

When selecting slave servers for large networks, consider the following guidelines to enhance BIND service performance:

- Ensure availability.

  Every zone should have at least two slave servers. Slave servers ensure an alternate source of information if a master server becomes temporarily unavailable. Very small zones with one BIND server can use traditional operating system backups to preserve BIND service information.

- Distribute the lookup load.

Identify master and slave servers that receive the most lookups and updates. Use the TCPIP SHOW NAME_SERVICE/STATISTICS command to determine the load on each server. Then configure additional servers within and outside of the local domain to act as slave servers.

- Place a slave server where users are located.

  If a zone's users are widely dispersed geographically, create subzones and configure slave servers to avoid long-distance copying of zone files. These subzones populate the new zone file with domain names used mainly by a group of users concentrated in one geographic area.

5.3.4.4 Selecting Caching-Only Servers

The value of implementing a caching-only server over a slave server comes from not having to perform zone transfers. Over time, a caching-only server builds up a cache of information most requested by the resolvers querying the caching-only server.

5.3.4.5 Selecting Forwarder and Forwarding-Slave Servers

You can configure any server to act as a forwarder server. A BIND server can use a forwarder server to resolve queries. Because a forwarder server accepts queries from many other servers, it can develop an extensive cache compared to caches on other servers. Having a forwarder server in your zone can reduce the total number of queries from the zone to the rest of the Internet.

Configure forwarding-slave servers if you do not want specific hosts to have access to the Internet or you want to restrict the server to using only specified forwarder servers. If you have a forwarding-slave server in your zone, you must have a forwarder server as well.

5.3.4.6 Determining Server Placement for LANs and Extended LANs

You might be able to use just one server on a LAN. Factors influencing your decision can include the expected lookup load and how you want to distribute it, and the capacity of the systems that you plan to use as BIND servers.

5.3.4.7 Determining Server Placement for Sites Connected by a WAN

When planning the placement of BIND servers in a wide area network (WAN) environment, avoid connections through WAN links. Place BIND servers so that most systems can access at least one server even if a WAN connection is unavailable.

At small sites connected to the rest of the network through a WAN, a BIND server is not necessary if the small site only occasionally uses resources on the other side of the WAN link. For example, if users at a small site sometimes contact nodes at the company's headquarters, it is probably sufficient to store the node names at headquarters, and it is not necessary to configure a BIND server at the small site.

Conversely, if a small site has many domains, configure a server there. Also, if you expect users to make frequent name changes, create a zone and store the information at the site's server. This further reduces WAN traffic and improves performance.

5.3.5 Planning SOA Values

The SOA resource record for the master zone and reverse domain files contain parameters that effect the operation of the slave servers. These include:

- refresh time interval

  This value specifies how often a slave checks that its data is up-to-date.

- retry time interval

  This is how often the slave will retry reaching a master name server after an attempt has failed.

- expire

  This value specifies how long after the slave fails to reach a master name server that the slave will expire its data.

- TTL

  This value specifies how long a server maintains cached information.

The values you choose for these parameters effect the load on the servers and the propagation time of changes. Longer times lessen the load but increase the time for changes to take effect. Shorter times lessen the time for changes to take effect but increase the load on the servers and the network.

RFC 1537 recommends the following values for top-level domain servers:

```
  86400 ;  Refresh    24 hours
   7200 ;  Retry       2 hours
2592000 ;  Expire     30 days
 345600 ;  TTL         4 days
```

5.3.6 Capacity Planning

When planning for the number and types of BIND servers, keep the following system points in mind.

- Memory utilization can be high causing the system to spend an excessive amount of time swapping pages in and out of memory
  - BIND creates new processes to handle zone transfers
  - Need enough memory to run all processes
- CPU utilization should be low:
  - 5% CPU utilization is acceptable
  - 10% CPU utilization is high
  - High CPU utilization is a symptom of a BIND configuration error
- DNS traffic on LANs should be a small proportion of the network bandwidth; slow leased lines and dial-up connections could be cause for concern
  - Use the following formula to estimate bandwidth used by DNS traffic:

$$\frac{(queries\ received\ per\ hour + answers\ sent\ per\ hour) * 800\ bits}{3600\ \ seconds\ /\ hour}$$

5.3.7 Planning Domain Registration

After you plan your domains and zones, your next steps are to create the necessary server files and to register zone and domain information with the upper-level domain administrator and zone technical contact. See Appendix A for information about domain registration.

5.3.8 Planning for Configuring BIND

Before you begin the installation and configuration process, you need to make some decisions about your BIND configuration. You can configure BIND as one of the following:

- Resolver only (Section 5.3.8.1)
- Name server only (Section 5.3.8.2)
- Resolver and name server (Section 5.3.8.1 and Section 5.3.8.2)

5.3.8.1 BIND Resolver

Before using TCPIP$CONFIG to configure the system as a BIND resolver, use the worksheet in Figure 5–3 to record BIND information about your system.

Figure 5–3 BIND Configuration Worksheet

BIND Configuration Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Default domain [1]: _____

BIND server for the domain

You can configure a maximum of three servers [1,2,3]

Server1 Host name: _____
Server1 IP address: _____
Server2 Host name: _____
Server2 IP address: _____
Server3 Host name: _____
Server3 IP address: _____

Domain Search List [2]: _____

[1] Use TCPIP$CONFIG procedure to set.
[2] Use management command to set.
[3] If configuring both client and server on the same system, use LOCALHOST IP address.

VM-0317A-AI

Table 5–6 describes BIND configuration parameters.

Table 5–6 BIND Parameters

Parameter	Enter on Worksheet...
Default domain	Enter the domain name for the system.
BIND server for the domain	Enter the name and IP address for one to three bind servers.
Server host name	Enter the host name of the BIND server.
Server IP address	Enter the IP address of the BIND server.
Domain search list	Enter a list of subdomains to be searched.

5.3.8.2 BIND Server

If you are configuring a system as a BIND server, you need to:

- Determine the BIND database files you need for the type of server
- Create and/or edit the database files
- Perform the configuration steps in Table 5–9

Table 5–7 shows the BIND database files that you need for each type of BIND server.

Table 5–7 Required BIND Server Files

File	Master	Slave	Caching-only
Forward translation file: *domain_name*.DB	Yes	No	No
Reverse translation file: *address*_IN-ADDR_ARPA.DB	Yes	No	No
Hints file: ROOT.HINT	Yes	Yes	Yes
Loopback files: 127_0_0.DB, LOCALHOST.DB	Yes	Yes	Yes
Configuration file: TCPIP$BIND.CONF	Yes	Yes	Yes

Table 5–8 shows the BIND database files that you may need to create or edit depending upon your current configuration and the type of server you are configuring.

Table 5–8 BIND Server Files to Create or Edit

File	Master	Slave	Caching-only
Forward translation file: *domain_name*.DB	Create	No	No
Reverse translation file: *address*_IN-ADDR_ARPA.DB	Create	No	No
Hints file: ROOT.HINT	No	No	No
Loopback files: 127_0_0.DB, LOCALHOST.DB	No	No	No
Configuration file: TCPIP$BIND.CONF	Edit	Edit	Edit

For instructions on how to populate the forward translation, reverse translation and the TCPIP$BIND.CONF files, refer to the *"Configuring and Managing BIND"* chapter in the *DIGITAL TCP/IP Services for OpenVMS Management* manual. The remaining files (ROOT.HINT, 127_0_0.DB and LOCALHOST.DB) are created for you when you execute TCPIP$CONFIG.

After collecting the configuration information and preparing your database and configuration files, following the steps in Table 5–9 to configure BIND on each server.

Table 5–9 BIND Configuration Steps

Is this your situation?	If so, after you install DIGITAL TCP/IP Services for OpenVMS version 5.0, you need to:
First time installation of DIGITAL TCP/IP Services for OpenVMS	1. Execute TCPIP$CONFIG to enable BIND. TCPIP$CONFIG creates the following files for you: • TCPIP$BIND_CONF.TEMPLATE • 127_0_0.DB, LOCALHOST.DB • ROOT.HINT 2. Make a copy of the TCPIP$BIND_CONF.TEMPLATE giving it a name of TCPIP$BIND.CONF. Edit this file with system specific information. 3. Manually create the forward and reverse translation database files.
A previous UCX BIND configuration exists on the system	1. Execute TCPIP$CONFIG to configure BIND. TCPIP$CONFIG will rename the old database files to the new names and create a BIND 8.1.2 configuration file.
You want to use existing UNIX BIND database and configuration files	1. Copy the files to SYS$SPECIFIC:[TCPIP$BIND] using the TCP/IP Services file naming conventions. (Refer to the *DIGITAL TCP/IP Services for OpenVMS Management* manual for file naming conventions.) 2. If your existing configuration file is in pre-BIND 8.1.2 format, you need to create a BIND 8.1.2 formatted configuration file. 3. If you have a BIND 8.1.2 formatted configuration file, you need to change any UNIX file specifications to OpenVMS file specifications and add any additional configuration features you may want. 4. Execute TCPIP$CONFIG to enable BIND.

5.4 DHCP or BOOTP

The DHCP server and BOOTP server components both allow you to set up your system to pass configuration information to diskless hosts on the TCP/IP network.

With TCP/IP Services, you can configure your system to use the newer, more robust DHCP server functionality or the older BOOTP server functionality. You must choose one or the other; the TCPIP$CONFIG procedure prevents you from enabling both server components on your system.

The server capabilities are summarized in Table 5–10.

Table 5–10 BOOTP and DHCP Capabilities

Component	Description
BOOTP server	Answers network bootstrap requests from diskless workstations and other network devices such as routers, terminal servers, and network switching equipment. Allows the diskless client machine to discover its own IP address, the address of a server host, and the name of a file to be loaded into memory and executed. The BOOTP server uses BOOTP to communicate the configuration information Then, if the client must retrieve the load file, it uses TFTP to transfer the file.
DHCP server	Provides BOOTP functionality as well as the ability to assign temporary or permanent IP addresses, subnet masks, and default gateways and other configuration parameters for both BOOTP and DHCP clients. An extension (or superset) of BOOTP, DHCP lets you centralize and automate IP address administration. You can use the DHCP graphical user interface (GUI) to configure various clients on the network from a single location to ensure that the configurations are consistent and accurate.

5.4.1 Configuring the BOOTP Server

If you choose to configure your system as a BOOTP server, TCPIP$CONFIG creates the BOOTP database. This database contains entries for each client that requests service from the BOOTP server.

Use the worksheet in Figure 5–4 to record your client entries for the BOOTP database.

For more information about configuring the BOOTP server on your system, see the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Figure 5–4 BOOTP Configuration Worksheet

BOOTP Configuration Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

BOOTP Clients That Will Access the BOOTP Server

Client Name: _____ _____ _____
System image: _____ _____ _____
System image location: _____ _____ _____
Hardware address: _____ _____ _____
IP address: _____ _____ _____
Additional information: _____
_____

Hosts That Will Access the BOOTP Server

Host Name: _____ _____ _____
System image: _____ _____ _____
System image location: _____ _____ _____
Hardware address: _____ _____ _____
IP address: _____ _____ _____
Additional information: _____
_____

Gateways To Be Used For Downloading

Name: _____ _____
Address: _____ _____
Name: _____ _____
Address: _____ _____

VM-0314A-AI

5.4.2 Configuring the DHCP Server

If you currently use BOOTP to manage your IP address space, the TCPIP$CONFIG procedure can migrate your environment to DHCP. Before you invoke TCPIP$CONFIG, answer these questions:

- Do you want to change some or all of your clients to DHCP or keep them as BOOTP-only clients?
- Do you want DHCP clients to always use the same IP address (static mapping) or do you want them to receive an arbitrary address from the IP address pool (dynamic mapping)?

  For example, you might want to use static IP addressing because you want to maintain an existing static assignment policy currently used in your network or you have devices or hosts that require permanent IP addresses.

The TCPIP$CONFIG procedures does the following:

- Converts the existing BOOTP database, if found, to the appropriate DHCP format so you can begin to serve your BOOTP clients without making any further changes

- Sets up the DHCP server to support the BOOTP clients

If you choose not to migrate your BOOTP clients to DHCP, the newly created DHCP configuration file remains empty and your BOOTP clients will not be served until you configure them with DHCP configuration.

Once the DHCP server is up and running on your system, you can use the DHCP graphical user interface (GUI) to do the following:

- Define IP addresses for any DHCP/BOOTP clients you add to the DHCP database.
- Modify certain characteristics of the DHCP server. For example, you can define the IP address ranges that are available for assignment to clients and set security parameters.
- Define various DHCP parameters to be offered to clients such as default gateways and DNS domain names.

The basic DHCP GUI server parameters are listed in Table 5–11 and explained in more detail in the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Use the worksheet in Figure 5–5 to record the server parameter information for your system.

Figure 5–5 DHCP Server Parameters

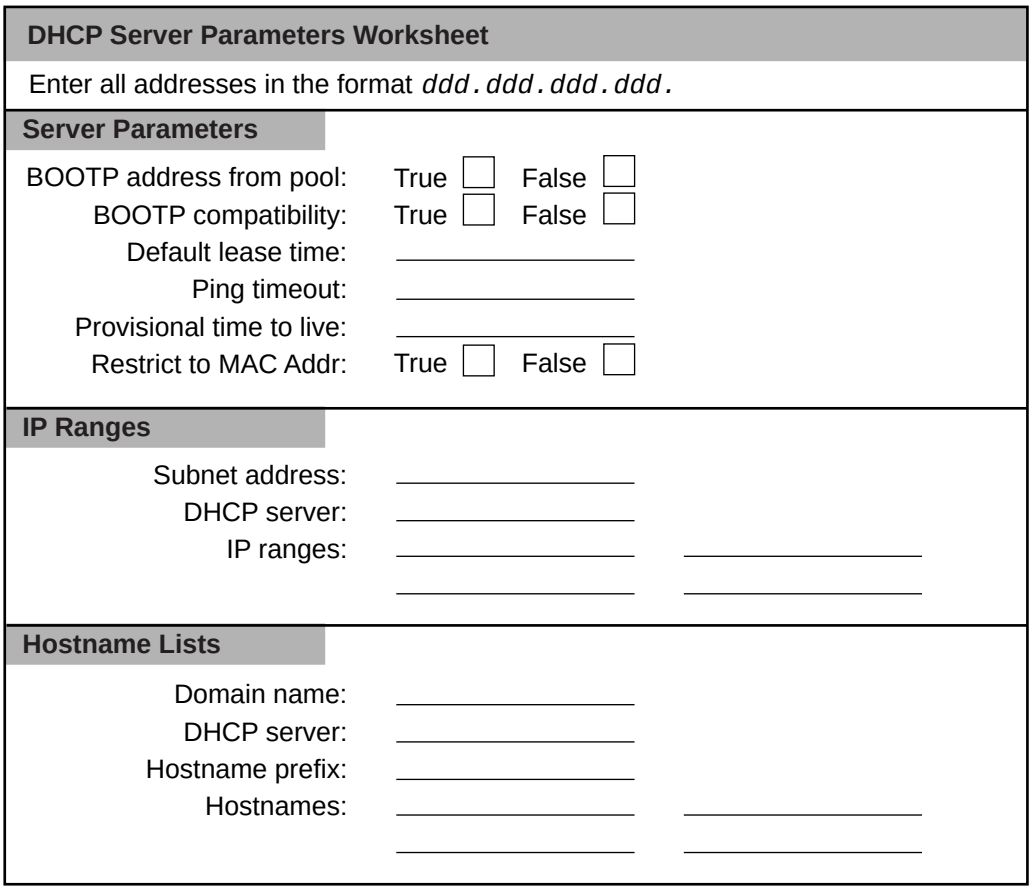


Table 5–11 describes the Basic DHCP Server Parameters.

Table 5–11 Basic DHCP Server Parameters

Parameter	Enter on Worksheet...
BOOTP address from pool	If you want the DHCP server to support only BOOTP clients whose addresses are configured in the BOOTP database, check False (default). If you want the DHCP server to allocate an address from the pool to BOOTP clients, check True. The address allocation is permanent.
BOOTP compatibility	If you want the DHCP server to also function as a BOOTP server when a client requests a BOOTP address, check True (default). Otherwise, check False.
Default lease time	The value (in days, hours, minutes, and seconds) that lease times extend for all clients that have no other value configured. The default lease time is one day.

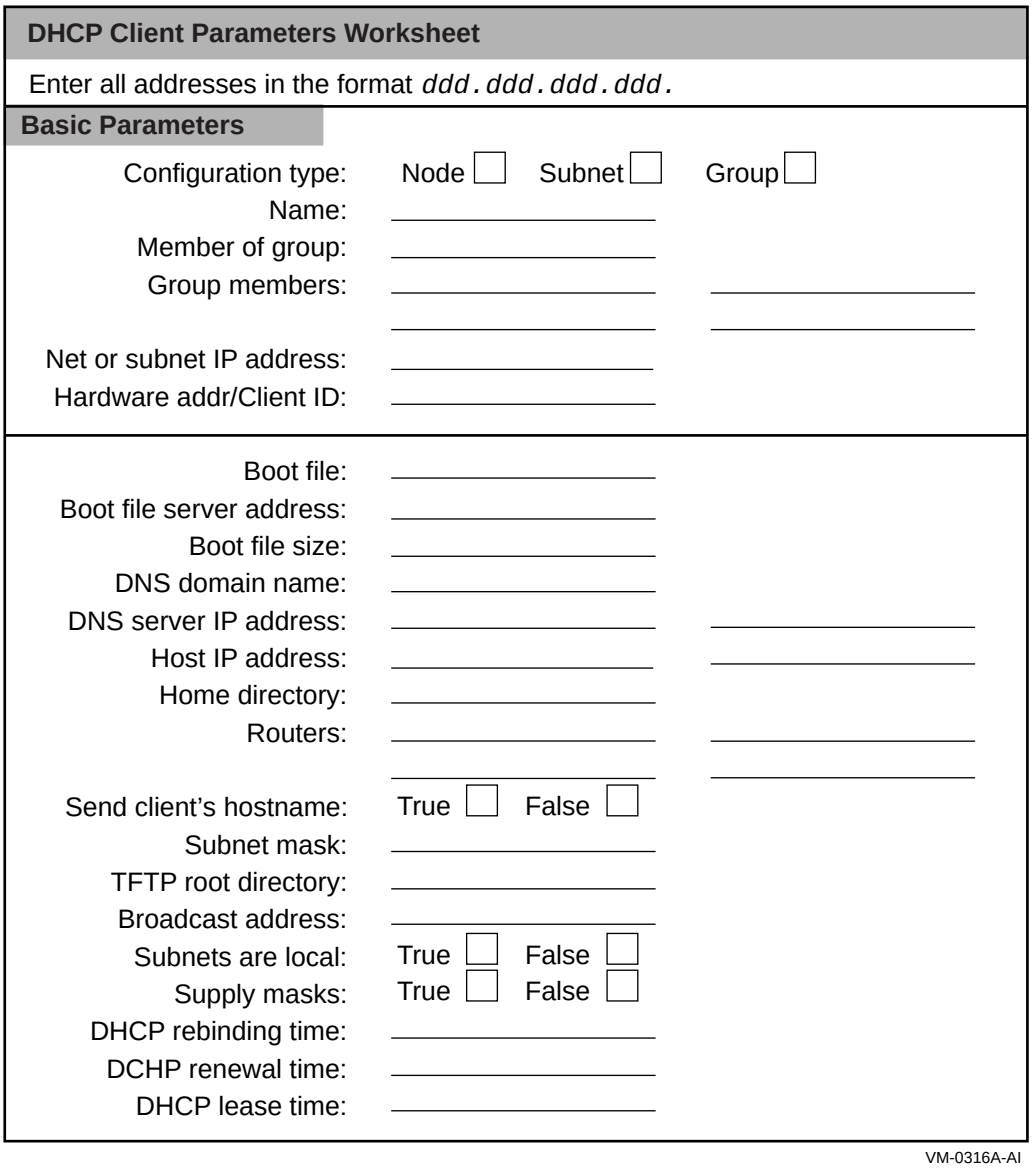
Table 5–11 (Cont.) Basic DHCP Server Parameters

Parameter	Enter on Worksheet...
Ping timeout	The time (in milliseconds) after which the ping request times out. The default is 500 milliseconds. If you do not want the DHCP server to ping before giving out an IP address, specify 0.
Provisional time to live	The maximum time (in hours, minutes, and seconds) that an IP address remains on the provisionally allocated list before it can be allocated to another client. This functionality prevents an IP address from being reused too quickly after a lease has expired.
Restrict to known MAC addresses	If you want to restrict service to only a client with a recognized, preconfigured MAC address that has been registered as known with the DHCP server, check True (default). To register a client for this purpose, use the DHCP GUI or DHCPDBREG utility. Otherwise, check False.

The basic DHCP GUI client parameters are listed in Table 5–12 and explained in more detail in the *DIGITAL TCP/IP Services for OpenVMS Management* guide.

Use the worksheet in Figure 5–6 to record the client parameter information for your system.

Figure 5–6 DHCP Client Parameters Worksheet



DHCP Client Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Configuration type: Node ☐ Subnet ☐ Group ☐
Name: _____
Member of group: _____
Group members: _____ _____
_____ _____
Net or subnet IP address: _____
Hardware addr/Client ID: _____

Boot file: _____
Boot file server address: _____
Boot file size: _____
DNS domain name: _____
DNS server IP address: _____ _____
Host IP address: _____ _____
Home directory: _____
Routers: _____ _____
_____ _____
Send client's hostname: True ☐ False ☐
Subnet mask: _____
TFTP root directory: _____
Broadcast address: _____
Subnets are local: True ☐ False ☐
Supply masks: True ☐ False ☐
DHCP rebinding time: _____
DCHP renewal time: _____
DHCP lease time: _____

VM-0316A-AI

Table 5–12 provides information of the basic DHCP parameters for responding to client.s

Table 5–12 Basic DHCP Parameters for Responding to Clients

Parameter	Enter on Worksheet...
Type of configuration	Check the configuration: node, subnet, or group.
Name of configuration	The name of the node, group, or subnet.
Member of group	The name of the configuration that provides the DHCP parameter values. Applicable for node, subnet, and group configurations.

Table 5–12 (Cont.) Basic DHCP Parameters for Responding to Clients

Parameter	Enter on Worksheet...
Group members	The nodes, subnets, and groups that compose this group. For group configuration only.
Net or subnet IP address	The IP address of the subnet. For example, if your subnet is 16.128, enter 16.128.0.0. You must include the trailing zeros.
Hardware address/Client ID	For node type configurations only. Enter the Ethernet address of the client node.
Boot file	The fully qualified directory or path name of the client's default boot image.
Boot file server address	The IP address of the server that stores the boot file.
Bootfile size	The length, in 512-octet blocks, of the default boot image for the client. Specify as a decimal number.
DNS domain name	The domain name the client should use when resolving hostnames using the Domain Name System (DNS).
DNS servers	A list of IP addresses for DNS name servers available to the client, in order of preference.
Home directory	The pathname for the boot file, if it is not specified in the boot file name.
Host IP address (BOOTP only)	The host IP address for BOOTP clients.
Routers	A list of IP addresses for routers.
Send client's hostname	If you want to send the client's host name, check True. Otherwise, check False (default).
Subnet mask	The client's subnet mask as defined in RFC 950. The subnet mask allows the addition of subnetwork numbers to an address and provides for more complex address assignments. If both the subnet mask and the router option are specified in a DHCP reply, the subnet mask option must be specified first.
TFTP root directory	The root directory for Trivial File Transfer Protocol (TFTP).
Broadcast address	The broadcast address in use on the client's subnet.
Subnets are local	If all subnets of the IP network to which the client is connected use the same MTU as the subnet of the network to which the client is directly connected, check True. Otherwise, check False. The client should assume that some subnets of the directly connected network might have smaller MTUs.
Supply masks	If the client needs to respond to subnet mask requests using ICMP, check True. Otherwise, check False.
DHCP rebinding time	The time interval (in seconds) from IP address assignment until the client requests a new IP address lease from any server on the network.
DHCP renewal time	The time interval (in seconds) from IP address assignment until the client attempts to extend the duration of its lease with the original server.
Lease time	The amount of time (in months, days, hours, minutes, and seconds) the DHCP server allows a DHCP client to use an IP address. For example, 2 months 5 days 45 minutes. The actual lease time is negotiated between the client and server.

5.5 Serial Lines

A serial connection is made between two systems using modems and telephone lines or other serial lines. DIGITAL TCP/IP Services for OpenVMS supports serial connections using the PPP (Point-to-Point Protocol) and SLIP (Serial Line

IP) (SLIP) (including CSLIP) protocols[1]. You can use any standard OpenVMS terminal device as a PPP or SLIP line. If the remote system is configured as a gateway to a network, local users can also reach other systems on that network through the serial connection.

If your OpenVMS system is part of a large network, you will probably use both PPP and SLIP for serial connections. An Internet standard, PPP is often preferred because it ensures interoperability between systems from a wide variety of vendors. PPP provides a way for your OpenVMS Alpha system to establish a dynamic IP network connection over a serial line without the use of an additional router or server hardware.

However, SLIP has been in use for a longer period of time and thus is available for more kinds of hardware. SLIP is available for most terminal servers and in most PC implementations of TCP/IP. Because SLIP and PPP do not communicate with each other, hosts wanting to communicate must use the same protocol. For example, if your terminal server supports only SLIP, remote hosts that connect through this server must also use SLIP.

5.5.1 Uses for PPP and SLIP

One of the largest applications for IP over serial lines is dialup access. With this type of configuration, your OpenVMS host answers calls and establishes a connection initiated by a user on a client host. The client host may be another OpenVMS system, a UNIX system, or a PC. Or, users on your host can originate the dialup connection to a remote host or terminal server running the same protocol.

Dedicated serial lines running PPP or SLIP can also be used to connect separate LANs into a single WAN. In such a configuration, the host at each end of the serial connection is always the same; no other hosts are allowed to connect to either serial device.

5.5.2 SLIP

Use the worksheet in Figure 5–7 to record SLIP information for your system.

[1] PPP is available for OpenVMS Alpha systems only.

Figure 5–7 SLIP Configuration Worksheet

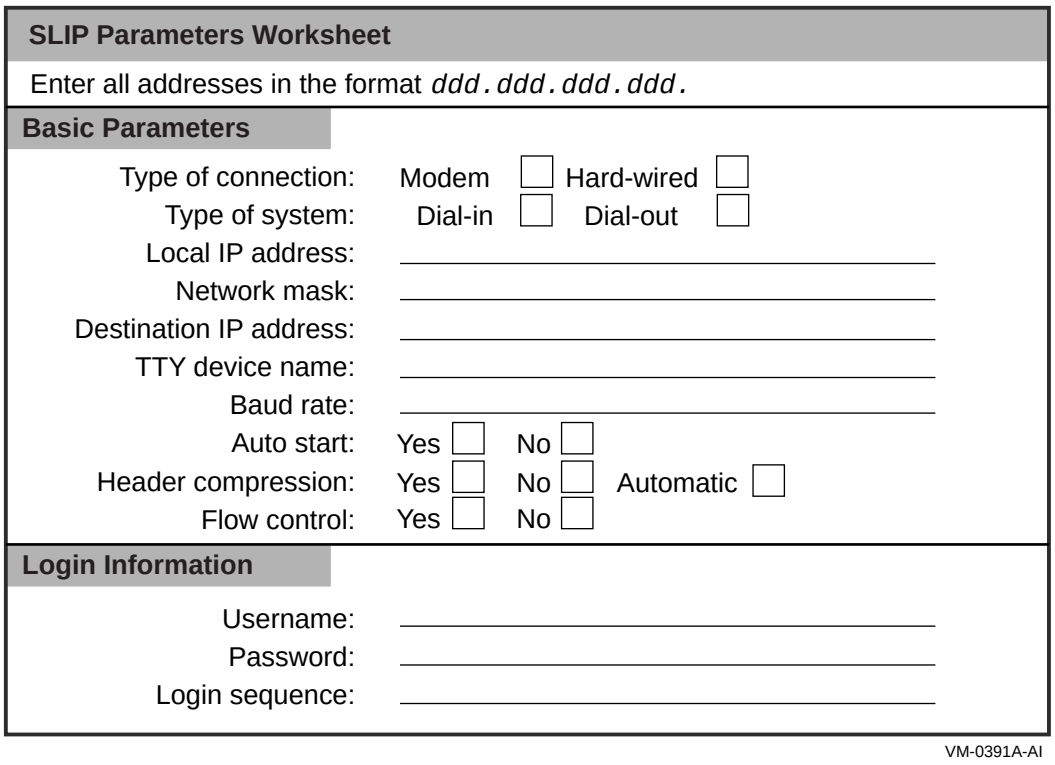


Table 5–13 describes the SLIP parameters.

Table 5–13 SLIP Parameters

Parameter	Enter on Worksheet...
Type of connection	Check whether the connection is hardwired or you are using a modem. If the systems are to be hardwired, make sure they are connected with an appropriate null modem cable. If they are to be connected by modem and telephone network, make sure modems and cables are configured correctly.
Type of system	Check dial-in if the system is to answer calls from remote systems (dialup provider). Check dial-out if the system is to place calls to a remote system (client).
Local IP address	Obtain the IP address for the SLIP interface. Each SLIP interface must have an IP address.
Network mask	Obtain the subnet mask. This must be the same for both systems.
Destination IP address	Obtain the IP address of the remote system's SLIP interface
TTY device name	Obtain the name of a valid terminal device that has a cable connection.
Baud rate	Determine the serial port speed used to connect the systems to each other or a system and the modem.

Table 5–13 (Cont.) SLIP Parameters

Parameter	Enter on Worksheet...
Auto start	Check YES if you want TCP/IP to automatically create the interface when TCP/IP starts.
Header compression	Check ON, if SLIP is to use header compression. Check AUTOMATIC, if SLIP is to turn on header compress if the remote site begins using header compression. Check NO, if SLIP is not to use header compression.
Flow control	Check YES, if you want SLIP to handle XON and XOFF characters and the remote host is running DIGITAL TCP/IP Services for OpenVMS.
Login information	Enter the user name, password and login sequence (For example, the login prompt to used on the dial-out connection) to use on this interface.

5.5.3 PPP

Use the worksheet in Figure 5–8 to record PPP parameter information for your system.

Figure 5–8 PPP Configuration Worksheet

PPP Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Type of connection: Modem ☐ Hard-wired ☐
Type of system: Dial-in ☐ Dial-out ☐
Local IP address: _____
Remote IP address: _____
TTY device name: _____
Baud rate: _____
PPPD otions: _____

VM-0392A-AI

Table 5–14 describes the PPP parameters.

Table 5–14 PPP Parameters

Parameter	Enter on Worksheet...
Type of connection	Check whether the connection is hardwired or you are using a modem. If the systems are to be hardwired, make sure they are connected with an appropriate null modem cable. If they are to be connected by modem and telephone network, make sure modems and cables are configured correctly.

Table 5–14 (Cont.) PPP Parameters

Parameter	Enter on Worksheet...
Type of system	Check dial-in if the system is to answer calls from remote systems (dialup provider). Check dial-out if the system is to place calls to a remote system (client).
Local IP address	Obtain the IP address for the PPP interface. If you have a standalone system, you must assign it an IP address. If you are using PPP to link it to another host that is connected to the Internet, assign the local system an address that is on the same subnet as the remote host. If the other host is not connected to the Internet, assign the local system any IP address.
Remote IP address	Obtain the remote system's IP address if you are dialing out as a client. If the local host is acting as a dialup provider, obtain the IP addresses to be assigned to the remote clients.
TTY device name	Obtain the name of any valid terminal device, for example, TTA0.
Baud rate	Determine the baud rate of the modem (or hard-wired line) used to connect the systems and the terminal line specification. If your modem automatically senses the line speed or if you are using a null modem cable between hosts, you can specify any rate up to the maximum supported by the hosts.
PPPD options	Determine the required PPPD options.

5.6 NTP

The Network Time Protocol provides a means to synchronize time and coordinate time distribution throughout a TCP/IP network. In TCP/IP Services Version 5.0, NTP is an implementation of the NTP Version 3 specification and maintains compatibility with NTP versions 1 and 2.

5.6.1 Selecting a Time Source

In the NTP environment, time is distributed through a hierarchy of NTP time servers. Each server adopts a stratum that indicates how far away it is operating from an external source of UTC.[2] Stratum 1 servers have access to an external time source, usually a radio clock. A stratum 2 server is one that is currently obtaining time from a stratum 1 server; a stratum 3 server gets its time from a stratum 2 server, and so on. To avoid long-lived synchronization loops, the number of strata is limited to 15.

Stratum 2 (and higher) hosts might be company or campus servers that obtain time from some number of primary servers and provide time to many local clients. In general:

- Lower strata hosts act as time servers.
- Higher strata hosts are clients who adjust their time clocks according to the servers.

Internet time servers are stratum 1 servers. Other hosts connected to an internet time server have stratum numbers of 2 or higher and may act as time servers for other hosts on the network. Clients choose one of the available servers with which to synchronize. Usually this is one from among the lowest stratum servers to which they have access.

[2] NTP times are an offset of UTC, formerly Greenwich Mean Time (GMT).

5.6.2 Determine the Operating Mode

The system manager of the local host determines which network hosts to use for synchronization and populates an NTP configuration file with a list of the participating hosts.

NTP hosts may be configured in one or more of the following modes:

- Client/server mode

  This mode indicates that the local host wants to obtain time from the remote server *and is willing* to supply time to the remote server if need be. This mode is appropriate in configurations involving a number of redundant time servers interconnected through diverse network paths. Internet time servers generally use this mode.

  Indicate this mode with a `peer` declaration in the configuration file. For example:

  ```
  peer 18.72.0.3
  ```

- Client mode

  This mode indicates that the local host wants to obtain time from the remote server *but it is not willing* to provide time to the remote server. Client mode is appropriate for file server and workstation clients that do not provide synchronization to other local clients. A host with higher stratum generally uses this mode.

  Indicate client mode with the `server` declaration in the configuration file. For example:

  ```
  server 18.72.0.3
  ```

- Broadcast mode

  This mode indicates that the local server will send periodic broadcast messages to a client population at the broadcast/multicast address specified. This specification normally applies to the local server operating as a sender.

5.6.3 Using NTP with Another Time Service

A local host may run more than one time service. For example, a host may have both NTP and DTSS (Digital Time Synchronization Service) installed. However, only one of these time services is allowed to set the system clock.

If you are running a time service in addition to NTP, you must stop NTP from setting the system clock by adding the following statements in the configuration file:

```
server 127.127.0 prefer
fudge 127.127.1.0 stratum 0
```

These statements make NTP use its own system clock as a reference clock. The host continues to respond to NTP time queries but won't make any adjustments to the system clock.

Use the worksheet in Figure 5–9 to record NTP parameter information for your system.

Figure 5–9 NTP Configuration Worksheet (TBS)

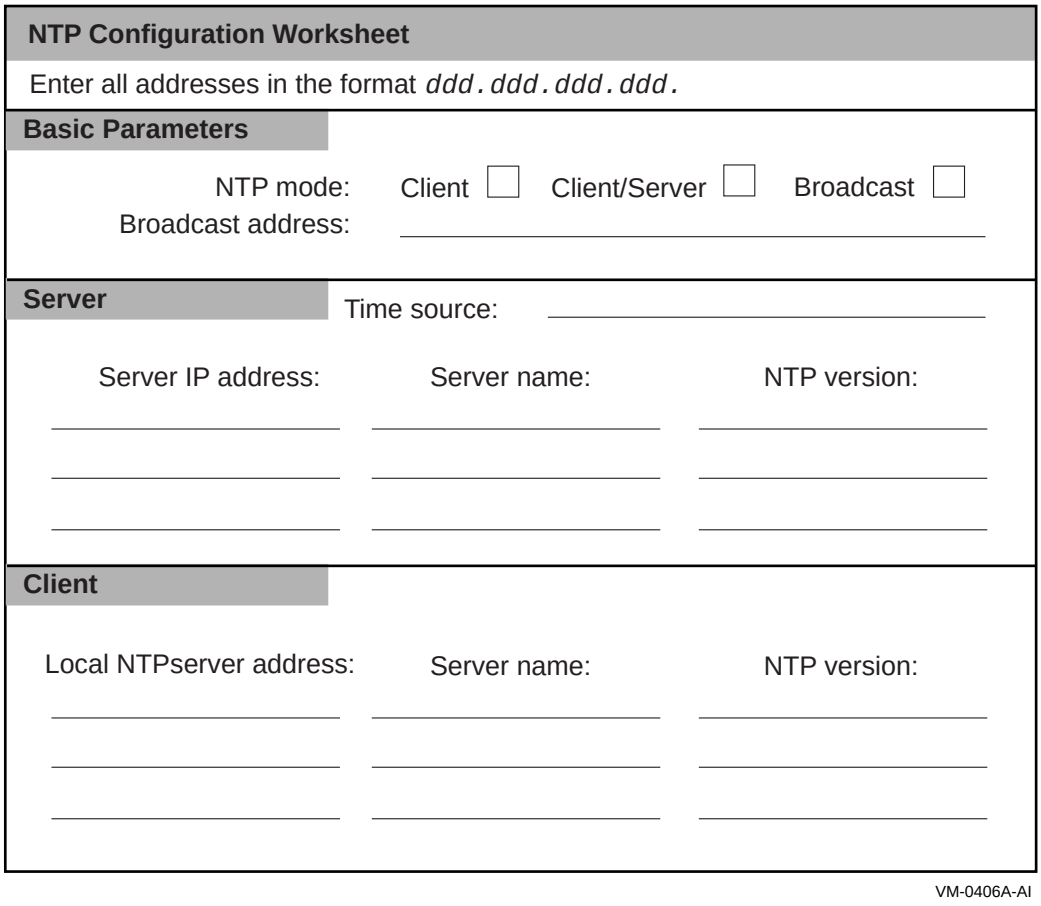


Table 5–15 describes the NTP parameters.

Table 5–15 NTP Parameters

Parameter	Enter on Worksheet...
NTP mode	Check: • Client If the system is to obtain time from a remote server, but will not provide time to another server. • Client/server If the system is to obtain time from a remote server, and will provide time to another sever. • Broadcast If the system will broadcast messages to clients at broadcast/multicast address
Broadcast address	If the NTP mode is broadcast, enter the broadcast/multicast address to which NTP time broadcasts are to be sent.

Table 5–15 (Cont.) NTP Parameters

Parameter	Enter on Worksheet...
NTP Server:	
Time Source	Enter if the time source is from an Internet NTP server or a local reference clock.
Server IP address	Enter the IP address of the Internet NTP server or the local reference clock.
Server name	The fully qualified host name of the Internet NTP server or the local reference clock.
NTP version	The version of NTP running on the Internet NTP server or the local reference clock.
NTP Client:	
Server address	Enter the IP address of the client's NTP server.
Server name	Enter the name of the client's NTP server.
Server version	Enter the NTP version running on the client's NTP server.

5.7 SNMP

If you plan to allow a remote SNMP management client to obtain information about your host or to set network parameters, you will need to configure SNMP.

You can configure your host to:

- Respond to a client's read requests (gets) for network information
- Send alert messages (traps) to a client as a result of events that might need to be monitored (for example, an authentication failure)
- Process client write requests (sets) on your host's MIB data items

Before running TCPIP$CONFIG, use the worksheet in Figure C–10 to record parameter information for the configuration procedure.

Figure 5–10 SNMP Configuration Worksheet

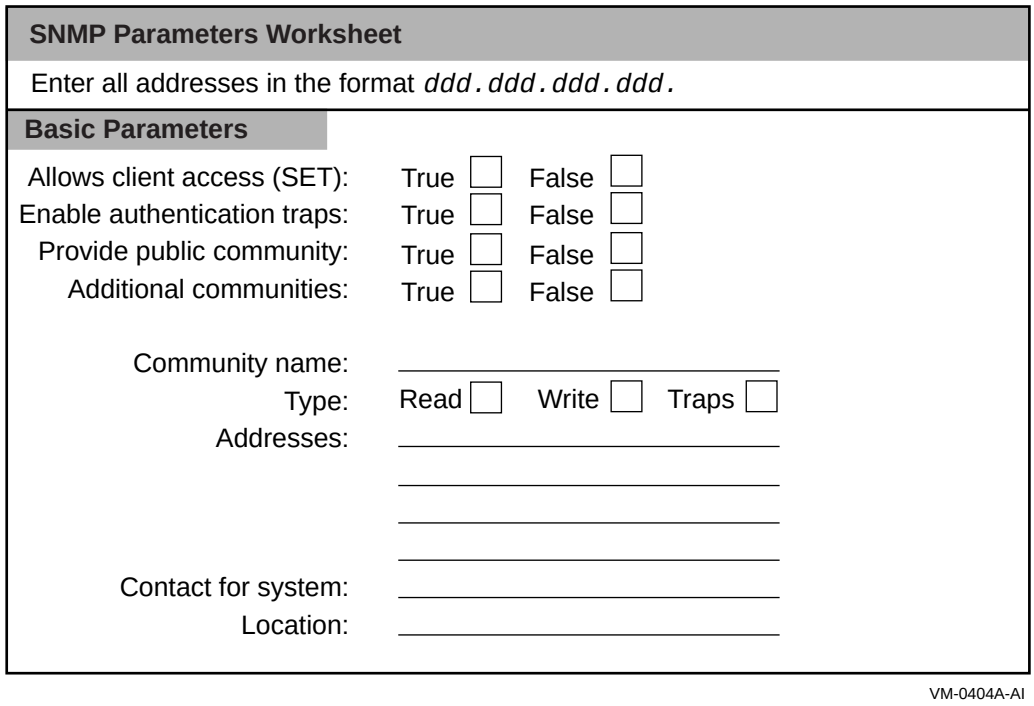


Table 5–16 describes the SNMP parameters.

Table 5–16 SNMP Parameters

Parameter	Enter on Worksheet...
Allow client access (SET)	Check YES, if you want to allow remote updates by clients.
Enable authentication traps	Check YES, if you want SNMP agents to send authentication traps
Provide public community	Check YES, if you will configure the public community to allow readonly access to clients on any host.
Provide additional communities	Check YES, if you want to provide additional communities. If you checked YES to enable authentication traps or allow enable client access, you will need to provide an additional community.
Community name	Enter the name of the community.
Type of access	Check Read if the master agent and subagents are to respond to a client's read requests (gets) for network information.
	Check Write if the maser agent and subagents are to process client write requests (sets) on your host's MIB data items.
	Check Trap, if the Master agent and subagents are allowed to send alert messages (traps) to a client as a result of unusual events.
Addresses	Enter the IP addresses of those systems that are to have access to the community
Contact for system	Enter the name of the person to contact about the system. Initial value can be 235 characters (or less).

Table 5–16 (Cont.) SNMP Parameters

Parameter	Enter on Worksheet...
Location	Enter the physical location of the system. Initial value can be 215 characters (or less).

After running TCPIP$CONFIG to configure SNMP, you can display the current SNMP configuration by entering the following command:

```
$TCPIP SHOW CONFIGURATION SNMP
```

5.8 User Accounts and Proxy Identities

You will need to set up accounts for local users, coordinate the establishment of corresponding accounts on remote systems, and create accounts for remote users who will be accessing server components on the local host.

When creating accounts for remote users, you can create one account for all remote users, an account for groups of remote users, or accounts for individual users. The strategy you use depends on your organization, system resources, and security needs.

Certain product components, for example, LPD/LPR, RMT/RCD, and NFS, act as servers for remote clients. You control access to your system and to these services by giving remote users proxy identities. A proxy identity maps a user account on a remote host to a OpenVMS account on the local host. The information you provide with each entry, along with the privileges you set for the account, let you specifically grant or deny access to your system.

When you configure the TCP/IP Services product, a proxy database file is created for you. Two types of proxies are required depending on the application used:

- **Communication proxy**

  A communication proxy provides an identity for remote users of RSH, RLOGIN, RMT/RCD, and LPD/LPR. Any user attempting to access these services must be registered in the proxy database. For each user, supply a proxy entry that contains the user name and the remote host name.

- **NFS proxy**

  An NFS proxy provides an identity for users of the NFS client, the NFS server. In addition to supplying host and user information in the proxy entry, you must also provide a user's UNIX style identity using a UID/GID pair. You can use NFS proxies to specify specify access to the NFS client, the NFS server, or both.

See the *DIGITAL TCP/IP Services for OpenVMS Management* manual for a more complete discussion about UNIX style identities and how the NFS server and client use the proxy database.

A

Network and Domain Registration Services

Before your BIND servers can resolve inquiries originating from outside your organization, you need to register:

- Your network with a network registry
- Your domain and name servers with your parent-top-level domain

A.1 Registering Your Network

Before your parent domain delegates your subdomain to you, it will require that you register your network and its `in-addr.arpa` subdomain.

To check that your network is registered, use one of the following `whois` services.

ARIN - `http://www.arin.net/whois/arinwhois.html`
APNIC - `http://www.apnic.net/reg.html`
RIPE - `http://www.ripe.net/db/whois.html`

You'll need to register before setting up your `in-addr.arpa` zones. There currently are three Internet Number registries:

ARIN - For the North and South America, the Caribbean and Sub-Saharan Africa geographical areas
APNIC - For Asia and Pacific geographical area
RIPE - For the European geographical area

A.1.1 American Registry for Internet Numbers (ARIN)

The American Registry for Internet Numbers is a non-profit organization established for the purpose of administration and registration of Internet Protocol (IP) numbers to the following geographical areas:

- North America
- South America
- Caribbean
- Sub-Saharan Africa.

See `http://www.arin.net/RSA.html` to review the *American Registry for Internet Numbers, Ltd. Registration Services Agreement.*

For questions, contact the ARIN Help Desk. The ARIN Help Desk is open from 8 a.m. to 7 p.m., Eastern Time. You may reach ARIN IP Registration Services by sending email to `hostmaster@arin.net`, telephone at (703) 227-0660, and by FAX at (703) 227-0676.

A.1.1.1 Registration Templates

ARIN templates are locate at http://www.arin.net/templates.html.

Use the templates listed in Table A–1 to register your IN-ADDR.ARPA domain, networks, and autonomous systems.

Table A–1 Registration Templates

If Registering This:	Use This Template:
Network	http://www.arin.net/templates/networktemplate.txt
Autonomous system	http://www.arin.net/templates/asntemplate.txt
Reverse-mapping	http://www.arin.net/templates/inaddrtemplate.txt
ISP Network	http://rs.arin.net/templates/isptemplate.txt

A.1.2 Asia Pacific Network Information Center (APNIC)

APNIC is the non-profit Internet Registry organization for the Asia Pacific region. Networks that will be connected/located within the geographic region maintained by the Asian-Pacific Network Information Center (http://www.apnic.net/) should use the APNIC template located at:

ftp://ftp.apnic.net/apnic/docs/isp-address-request

Follow the instructions that APNIC provides for submission of that template. To contact the APNIC:

APNIC
Office: Level 1, 33 Park Road
Milton Queensland 4064
Australia

Postal: APNIC
Box 2131
Milton Queensland 4064
Australia

Phone: +61-7-3367-0490
FAX: +61-7-3367-0482
Email: info@apnic.net
Web: http://www.apnic.net
Office hours: 9:00am - 6:00pm
Mon to Fri (Australian EST, UTC+10:00)

A.1.3 Reseaux IP Europeens (RIPE)

Networks that will be connected/located within the European geographic regions maintained by RIPE (http://www.ripe.net/) should use the European template located at:

ftp://ftp.ripe.net/ripe/forms/netnum-appl.txt

Please follow the instructions that RIPE provides for submission of that template.

To contact RIPE:

RIPE NCC
Singel 258
1016 AB Amsterdam

The Netherlands

Phone: +31 20 535 4444
FAX: +31 20 535 4445

A.2 Registering Your Domain Name

You need to notify your parent-top-level-domain of the servers that are authoritative for your domain name. Depending upon the domain of which you are a member, different registration methods may apply.

For members of the generic top-level domains like `com` and `edu`, you need to contact the InterNIC.

You can access the InterNIC's web site at `http://rs.internic.net` for up-to-date information or:

Network Solutions Inc.
ATTN: InterNIC Registration Services
505 Huntmar Park Drive
Herndon, VA 22072

B

Requests for Comments (RFCs)

The Internet Architecture Board (IAB) of the Internet Society (ISOC) produces numerous protocol standards and operational procedures known as *Requests for Comments* (RFCS). RFCs are reviewed by appropriate IAB task forces and those RFCs that are destined to become Internet standards progress through a standards track: *Proposed Standard, Draft Standard, and Internet Standard*.

You may obtain RFCs by e-mail or FTP from many RFC repositories around the world. To obtain information on how to retrieve RFCs, send an email message to: RFC-SERVER@ISI.EDU with the following information:

```
TO:           RFC-SERVER@ISI.EDU
SUBJECT:      getting rfcs.
MESSAGE BODY: help: ways_to_get_rfcs
```

Also, you may visit the following URL:

`http://www.rfc-editor.org/`

Table B–1 lists the RFCs associated with the DIGITAL TCP/IP Services for OpenVMS implementation.

Important

This list is provided for convenience only and does not necessarily imply full support for each RFC.

Table B–1 Relative RFCs for TCP/IP Services for OpenVMS

Protocol	Number	Title
ARP	RFC 826	Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware
IP-E	RFC 894	Standard for the Transmission of IP Datagrams over Ethernet Networks
IP-IEEE	RRC 1042	Standard for the Transmission of IP Datagrams over IEEE 802 Networks
BOOTP	RFC 951	Bootstrap Protocol
	RFC 1048	BOOTP Vendor Information Extensions
	RFC 1084	BOOTP Vendor Information Extensions

Table B–1 (Cont.) Relative RFCs for TCP/IP Services for OpenVMS

Protocol	Number	Title
CIDR	RFCS 1517, 1518, 1519, 1520	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
CSLIP	RFC 1144	Compressing TCP/IP Headers for Low-Speed Serial Links
DHCP	RFC 2131	Dynamic Host Configuration Protocol
	RFC 1534	Interoperation Between DHCP and BOOTP
DNS, BIND	RFC 819	Domain Naming Convention for Internet User Applications
	RFC 920	Domain Requirements
	RFC 974	Mail Routing and the Domain System
	RFC 1032	Domain Administrator's Guide
	RFC 1033	Domain Administrator's Operations Guide
	RFC 1034	Domain Names — Concepts and Facilities
	RFC 1035	Domain Names — Implementation and Specification
	RFC 1101	DNS Encoding of Network Names and Other Types
	RFC 1183	New DNS RR Definitions
	RFC 1400	Transition and Modernization of the Internet Registration Service
	RFC 1535	A Security Problem and Proposed Correction with Widely Deployed DNS Software
	RFC 1536	Common DNS Implementation Errors and Suggested Fixes
	RFC 1537	Common DNS Data File Configuration Errors
	RFC 1591	Domain Name System Structure and Delegation
	RFC 1597	Address Allocation for Private Internets
	RFC 1637	DNS NSAP Resource Records
Subnetting	RFC 950	Internet Standard Subnetting Procedure
EGP	RFC 904	Exterior Gateway Protocol Formal Specification
	RFC 911	RFC EGP Gateway Under Berkeley UNIX 4.2
	RFC 1009	Requirements for Internet Gateways
	RFC 1092	EGP and Policy Based Routing in the New NSFNET Backbone
BGP	RFC 1771	A Border Gateway Protocol 4 (BGP-4)
	RFC 1267	A Border Gateway Protocol 3 (BGP-3)
FDDI	RFC 1390	Transmission of IP and ARP over FDDI Networks
FTP	RFC 959	File Transfer Protocol
ICMP	RFC 792	Internet Control Message Protocol
	RFC 1256	ICMP Router Discovery Messages
IP	RFC 791	Internet Protocol
LPD	RFC 1179	Line Printer Daemon Protocol
NETBIOS	RFC 1001	Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods

Table B–1 (Cont.) Relative RFCs for TCP/IP Services for OpenVMS

Protocol	Number	Title
NTP	RFC 1305	Network Time Protocol (Version 3) Specification and Implementation
NFS	RFC 1094	NFS: Network File System Protocol Specification
OSPF	RFC 1583	OSPF Version 2
POP	RFC 822	Standard for the Format of ARPA Internet Text Messages
	RFC 1082	Post Office Protocol Version 3: Extended Service Offerings
	RFC 1321	The MD5 Message-Digest Algorithm
	RFC 1725	Post Office Protocol (Version 3)
PPP	RFC 1661	The Point-to-Point Protocol (PPP) for the Transmission of Multi-Protocol Datagrams over Point-to-Point Links (obsoletes RFC 1331)
RARP	RFC 903	Reverse Address Resolution Protocol
RIP	RFC 1058	Routing Information Protocol, Version 1
	RFC 1388	Routing Information Protocol, Version 2
RLP	RFC 887	Resource Location Protocol
Routing	RFC 1112	Host Extensions for IP Multicasting
	RFC 1256	ICMP Router Discovery Messages
RPC	RFC 1057	RPC: Remote Procedure Call Protocol Specification Version 2
SLIP	RFC 1055	Nonstandard for Transmission of IP Datagrams Over Serial Lines: SLIP
SMTP	RFC 821	Simple Mail Transfer Protocol
	RFC 822	Standard for the Format of ARPA Internet Text Messages
SNMP	RFC 1155	Structure and Identification of Management Information for TCP/IP-Based Internets
	RFC 1156	Management Information Base for Network Management of TCP/IP-Based Internets
	RFC 1157	A Simple Network Management Protocol
	RFC 1158	Management Information Base for Network Management of TCP/IP-based Internets: MIB-II
	RFC 1212	Concise MIB Definitions
	RFC 1213	Management Information Base for Network Management of TCP/IP-based Internets: MIB-II
	RFC 1214	OSI Internet Management: Management Information Base
	RFC 1215	Convention for Defining Traps for Use with the SNMP
	RFC 1442	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
	RFC 1514	Host Resource MIB
	RFC 1592	Simple Network Management Protocol Distributed Protocol Interface
Path MTU	RFC 1191	Path MTU Discovery

Table B–1 (Cont.) Relative RFCs for TCP/IP Services for OpenVMS

Protocol	Number	Title
Sun ONC RPC, XDR	RFC 1790	An Agreement between the Internet Society and Sun Microsystems, Inc. in the Matter of ONC RPC and XDR Protocols
TELNET	RFC 854	Telnet Protocol Specification
	RFC 855	Telnet Option Specification
	RFC 856	Telnet Binary Transmission
	RFC 857	Telnet Echo Option
	RFC 858	Telnet Suppress Go Ahead Option
	RFC 859	Telnet Status Option
	RFC 860	Telnet Timing Mark Option
	RFC 861	Telnet Extended Options: List Option
TFTP	RFC 1350	The TFTP Protocol
TP	RFC 868	Time Protocol
TCP	RFC 793	Transmission Control Protocol
UDP	RFC 768	User Datagram Protocol
XDR	RFC 1014	XDR: External Data Representation Standard

C

Configuration Worksheets

This appendix contains the worksheets for use in the planning process before you attempt to configure the TCP/IP Services for OpenVMS product. Read Chapter 5 for information on how to fill out the worksheets.

You will want to complete a copy of these forms for each system installation. They will also serve as documentation, which can be of help if you need to reconstruct your original TCP/IP configuration.

Figure C–1 Network Interface Configuration Worksheet

Network Interface Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Type of installation:	Single ☐ TCP/IP cluster member ☐
Adapter name:	_____
Host name:	_____
IP address source:	DHCP server ☐ User supplied ☐
Internet address:	_____
Network mask:	_____
Broadcast mask:	_____
Cluster host name:	_____
Cluster address/mask:	_____

VM-0389A-AI

Figure C–2 Routing Configuration Worksheet

Routing Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Type of routing: Static ☐ Dynamic ☐
Static default route: _____
Type of dynamic routing: ROUTED ☐ GATED ☐

ROUTED

Supply dynamic routing information Yes ☐ No ☐

GATED

Configuration file: _____
Routing protocols: RIP ☐ OSPF ☐ EGP ☐ BGP ☐ Hello ☐
Router discovery: Yes ☐ No ☐

VM-0390A-AI

Figure C–3 BIND Configuration Worksheet

BIND Configuration Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Default domain [1]: _____
BIND server for the domain

You can configure a maximum of three servers [1,2,3]

Server1 Host name: _____
Server1 IP address: _____
Server2 Host name: _____
Server2 IP address: _____
Server3 Host name: _____
Server3 IP address: _____

Domain Search List [2]: _____

[1] Use TCPIP$CONFIG procedure to set.
[2] Use management command to set.
[3] If configuring both client and server on the same system, use LOCALHOST IP address.

VM-0317A-AI

Figure C–4 BOOTP Configuration Worksheet

BOOTP Configuration Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

BOOTP Clients That Will Access the BOOTP Server

Client Name: _____ _____ _____
System image: _____ _____ _____
System image location: _____ _____ _____
Hardware address: _____ _____ _____
IP address: _____ _____ _____
Additional information: _____
_____

Hosts That Will Access the BOOTP Server

Host Name: _____ _____ _____
System image: _____ _____ _____
System image location: _____ _____ _____
Hardware address: _____ _____ _____
IP address: _____ _____ _____
Additional information: _____
_____

Gateways To Be Used For Downloading

Name: _____ _____
Address: _____ _____
Name: _____ _____
Address: _____ _____

VM-0314A-AI

Figure C–5 DHCP Server Parameters

DHCP Server Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Server Parameters

BOOTP address from pool: True ☐ False ☐
BOOTP compatibility: True ☐ False ☐
Default lease time: _____
Ping timeout: _____
Provisional time to live: _____
Restrict to MAC Addr: True ☐ False ☐

IP Ranges

Subnet address: _____
DHCP server: _____
IP ranges: _____ _____
_____ _____

Hostname Lists

Domain name: _____
DHCP server: _____
Hostname prefix: _____
Hostnames: _____ _____
_____ _____

VM-0315A-AI

Figure C–6 DHCP Client Parameters Worksheet

DHCP Client Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

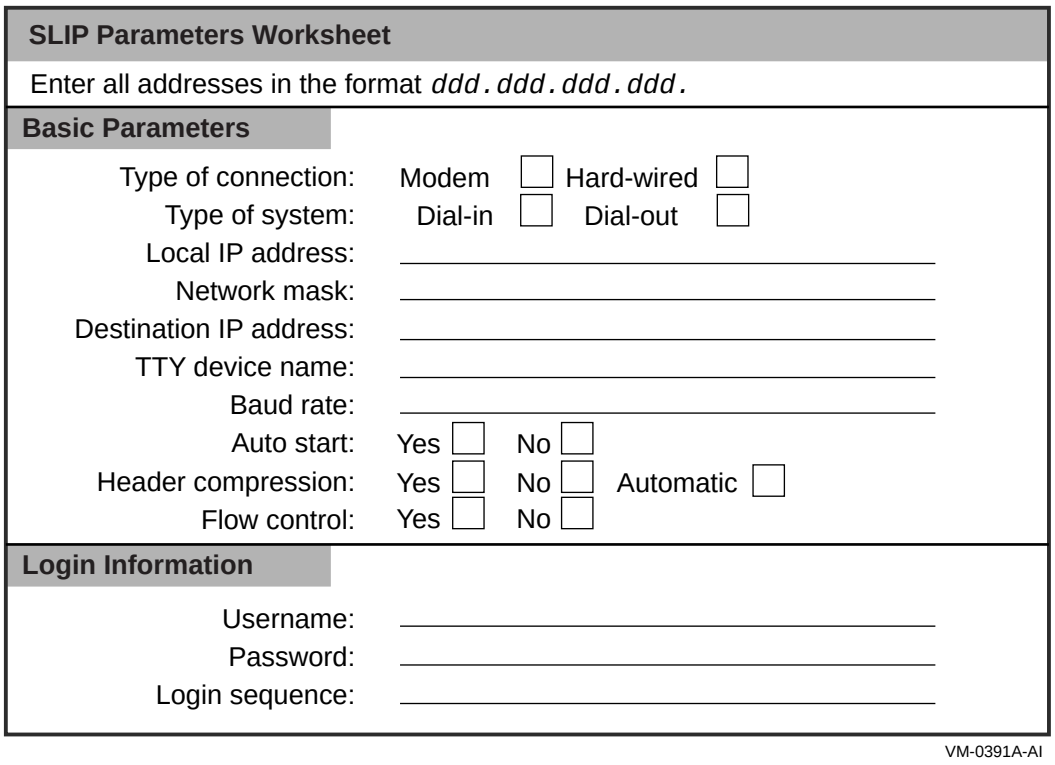
Basic Parameters

Configuration type:	Node ☐ Subnet ☐	Group ☐
Name:	_____	
Member of group:	_____	
Group members:	_____	_____
	_____	_____
Net or subnet IP address:	_____	
Hardware addr/Client ID:	_____	

Boot file:	_____	
Boot file server address:	_____	
Boot file size:	_____	
DNS domain name:	_____	
DNS server IP address:	_____	_____
Host IP address:	_____	_____
Home directory:	_____	
Routers:	_____	_____
	_____	_____
Send client's hostname:	True ☐ False ☐	
Subnet mask:	_____	
TFTP root directory:	_____	
Broadcast address:	_____	
Subnets are local:	True ☐ False ☐	
Supply masks:	True ☐ False ☐	
DHCP rebinding time:	_____	
DCHP renewal time:	_____	
DHCP lease time:	_____	

VM-0316A-AI

Figure C–7 SLIP Configuration Worksheet



SLIP Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

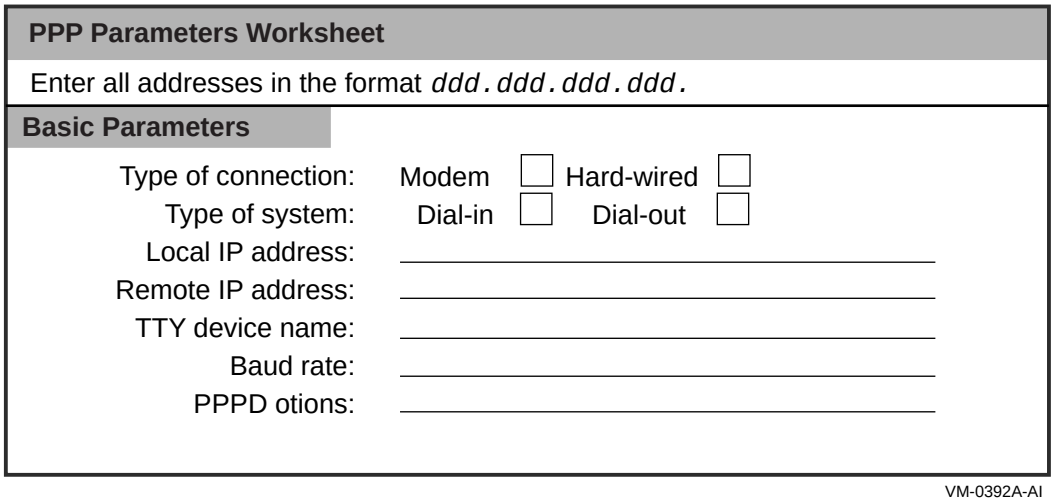
Type of connection: Modem ☐ Hard-wired ☐
Type of system: Dial-in ☐ Dial-out ☐
Local IP address: _____
Network mask: _____
Destination IP address: _____
TTY device name: _____
Baud rate: _____
Auto start: Yes ☐ No ☐
Header compression: Yes ☐ No ☐ Automatic ☐
Flow control: Yes ☐ No ☐

Login Information

Username: _____
Password: _____
Login sequence: _____

VM-0391A-AI

Figure C–8 PPP Configuration Worksheet



PPP Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Type of connection: Modem ☐ Hard-wired ☐
Type of system: Dial-in ☐ Dial-out ☐
Local IP address: _____
Remote IP address: _____
TTY device name: _____
Baud rate: _____
PPPD otions: _____

VM-0392A-AI

Figure C–9 NTP Configuration Worksheet

NTP Configuration Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

NTP mode: Client ☐ Client/Server ☐ Broadcast ☐

Broadcast address: _____

Server

Time source: _____

Server IP address:	Server name:	NTP version:

Client

Local NTPserver address:	Server name:	NTP version:

VM-0406A-AI

Figure C–10 SNMP Configuration Worksheet

SNMP Parameters Worksheet

Enter all addresses in the format *ddd.ddd.ddd.ddd.*

Basic Parameters

Allows client access (SET):	True ☐	False ☐
Enable authentication traps:	True ☐	False ☐
Provide public community:	True ☐	False ☐
Additional communities:	True ☐	False ☐

Community name: _____

Type: Read ☐ Write ☐ Traps ☐

Addresses: _____

_____

_____

_____

Contact for system: _____

Location: _____

VM-0404A-AI

Glossary

This glossary defines terms that explain the features and operation of the DIGITAL TCP/IP Services for OpenVMS product.

G.1 Definitions

absolute path name

A path name that starts with a slash (/); specifies a file that can be found by starting at the root of the file system and traversing the file tree.

absolute time

A specific date or time of day; specified in the following format: [dd-mmm-yyyy] [:hh:mm:ss:cc].

abstract syntax

The description of a data structure that is independent of host structures or codes.

Abstract Syntax Notation One (ASN.1)

The language used by ISO protocols for describing abstract syntax. Most notable use in TCP/IP is for Simple Network Management Protocol (SNMP) and the Management Information Base I & II (MIB-I & MIB-II). The rules of ASN.1 are independent of the encoding techniques used to represent them.

access control information

A character string with login information that validates connect or login at a remote host.

access control list (ACL)

A list that defines the kinds of access to be granted or denied to users.

access rights

A set of privileges that determines what users can do.

ACK

See **acknowledgment**.

acknowledgment (ACK)

A type of message sent to indicate that a block of data arrived at its destination without error. A control bit (acknowledgment flag) in the TCP header indicates that the acknowledgment number field is significant for each segment in a packet.

ACL

See **access control list**.

ACP

See **ancillary control process**.

active port

A port that is bound to a process.

address

A number or group of numbers that uniquely identifies a network node within its own network or internet. (*See also* **IP address** and **hardware address**.)

address mask

A 32-bit value used to identify which bits in an IP address correspond to the network and subnet portions of the address.

address resolution

The process of relating an IP address to a hardware address, when both refer to the same device, for example, conversion of an IP address into the corresponding Ethernet, Token Ring, or FDDI hardware address. This may require broadcasting on a local network. *See also* **Address Resolution Protocol**.

Address Resolution Protocol (ARP)

The TCP/IP protocol that dynamically binds an IP address to a hardware address such as an Ethernet or FDDI address; limited to physical network systems that support broadcast packets that can be heard by all hosts on a single physical network. *See also* **proxy ARP**.

addressing

The function that ensures that network systems are correctly identified at all times.

addressing authority

The authority, such as the American National Standards Institute (ANSI), responsible for assigning Network Interface layer addresses within an addressing domain.

addressing domain

A level in a hierarchy of Network Interface layer addresses.

adjacency

A single connection to an adjacent node; collection of state information representing a node in the local node's routing databases.

A relationship formed between selected neighboring routers for the purpose of exchanging routing information. Not every pair of neighboring routers becomes adjacent.

adjacency address

An address that identifies a local subnet access point and a subnet address of an adjacent system.

adjacent nodes

The nodes with direct lines between them; can communicate without an intermediate system. For example, all nodes on an Ethernet LAN are adjacent to each other.

administrative domain

A group of hosts, routers, and networks operated and managed by a single organization. Routing within an administrative domain is based on a consistent technical plan. An administrative domain is viewed from the outside, for purposes of routing, as a cohesive entity, of which the internal structure is unimportant. Information passed by other administrative domains is trusted less than information from one's own administrative domain.

advertisement lifetime

A field in the Router Discovery Protocol router advertisement message that indicates how long advertisement addresses are valid. A lifetime of zero indicates the one or more addresses are no longer valid.

aged packet

A data packet that is discarded because it exceeded the maximum number of hops while being forwarded through the network.

agent

A system that acts on behalf of another system. (1) Client/server model: Part of the system that initiates, prepares, and exchanges information preparation on behalf of a client or server application. (2) Network management: Portion of an entity that responds to management requests and/or preprogrammed trap.

agent access module

The portion of an agent responsible for the agent's end of SNMP.

agent access point

The instance of a connection between a client or director and a server or agent.

agent address

An address that specifies the information needed by a director to establish communications with the agent's management interface.

agent attributes

The attributes maintained by the agent; do not cross the internal management interface.

aggregate throughput

See **throughput**.

alias

A name, usually easy to remember, that is translated from a different name, usually difficult to remember. Most often used as an optional alternate name for a host. *See also* **host name**.

alias node identifier

An optional node name used by some or all nodes in an OpenVMS cluster, allows them to be treated as one node.

alternate address notation

The internet address notation that conveys the same information as the common notation, but consists of two parts: network and host.

American National Standards Institute (ANSI)

The organization that coordinates U.S. standards in many areas, including computers and communications.

American Standard Code for Information Interchange (ASCII)

The standard character set that assigns an octal sequence to each letter, number, and selected control characters.

ancillary control process (ACP)

The process that acts as an interface between user software and an I/O driver; provides functions supplementary to those performed in the driver, such as file and directory management.

anonymous (FTP)

A convention of the File Transfer Protocol that allows individuals who do not have explicit authorization to transfer files to and from a host without the need for an account and password. The individual usually logs in with a generic user ID and e-mail address as password.

ANSI

See **American National Standard Institute**.

API

See **Application Programming Interface**.

application

A program that provides functionality for end users of systems.

Application layer

The top-most layer in the Internet architecture model where the user interacts with an application such as Network File Service (NFS), File Transfer Protocol (FTP), and mail.

application process

A part of a distributed application running on a single host.

Application Programming Interface (API)

A standardized set of routines that makes system functions available to programmers.

architecture

The structure of a system, a description of which can be used to re-create the system.

ARP

See **Address Resolution Protocol**.

ASCII

See **American Standard Code for Information Interchange**.

assigned numbers

The numbers officially assigned as part of the Internet standards.

Asynchronous Transfer Mode (ATM)

The method for dynamic allocation of bandwidth using a fixed-size packet (called a cell). Also known as fast packet.

asynchronous transmission

The mode of transmission in which the time intervals between character transmissions differ. Each character is surrounded by start and stop bits to allow the receiving device to recognize the beginning and end of each character (also called start-stop transmission).

ATM

See **Asynchronous Transfer Mode**.

attribute

The controllable or observable part of an entity; a variable that network managers and applications programmers can manipulate for optimal performance.

attribute group

A named collection of attributes grouped together, such as all information relating to errors.

authentication

Verification of the identity of a person or process attempting to access a system.

authentication server

The software that searches the proxy database for valid user and group identification for remote personal computer users and returns them to PC-NFS.

authority

A name server is said to have authority for a zone. That is, the name server has complete information about a part of a domain space for which the name server is consider to be the authority. A name server may be the authority for one or more zones. Authority for a domain space may be delegated to one or more zones.

authoritative answer

In response to an `nslookup` or a resolver query, an answer is an authoritative answer if a server queries the authority for the zone and returns the answer. A server returns a nonauthoritative answer when the server answer comes from its own cache.

autonomous confederation

A group of independent computer systems that trust each other regarding routing and reachability information; members believe information provided by other members in preference to information received from systems that are not part of the confederation.

autonomous system

A collection of networks controlled by one administrative authority. The gateways within this system are expected to trust one another and to share and update routing information among themselves by any mutually agreeable protocol. A core gateway must also be designated to share routing information with other

autonomous systems by means of an External Gateway Protocol. *See also* **External Gateway Protocol.**

A set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs. Since this classic definition was developed, it has become common for a single AS to use several interior gateway protocols and sometimes several sets of metrics within an AS.

The use of the term "autonomous system" stresses that even when multiple internal gateway protocols and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it. The AS is represented by a number between 1 and 65534, assigned by the Internet Assigned Numbers Authority.

automounting

The process of mounting NFS file systems on a as-needed basis. The NFS file system automatically unmounts after a period of inactivity on the file system (Default is 5 minutes). You specify file systems to be automounted in the `automounts` map file.

auxiliary server

The DIGITAL TCP/IP Services for OpenVMS software that runs as a background process and listens for incoming requests for services. When it receives a request, it runs the appropriate server application; includes `inetd`, security, and logging options.

availability

The proportion of time a specific piece of equipment, system, or network is usable, compared to the total time it is expected to be.

backbone

The primary connectivity mechanism of a hierarchical distributed system. Usually a high-speed high-performance network that links together other networks into an internetwork. All systems with connectivity to an intermediate system on the backbone will connect to each other. This does not prevent systems from setting up private arrangements with each other to bypass the backbone for reasons of cost, performance, or security.

background mounting

In the UNIX environment, the default mount option is to retry remote mount requests in the foreground. If during a boot process, any server listed in `/etc/fstab` is not currently available, the local system will not finish booting until the server becomes available. With background mounting, a remote mount request is executed once in a foreground process. If the mount request fails, the requests is retried in a background process. This allows the local system to continue the boot procedure without waiting for the server to become available.

In the TCPIP Services for OpenVMS environment, background mounting provides a retry of the mount command for ??? Need some information here regarding what happens in OpenVMS world. ???

bandwidth

(1) Technically: The difference, in Hertz (Hz), between the highest and lowest frequencies of a transmission channel. (2) Typically: The amount of data that can be sent through a communications circuit.

baseband

A characteristic of any network technology that uses a single carrier frequency and requires all stations attached to the network to participate in every transmission; only one communication channel is provided at a time. *See also* **broadband**.

BBS

See **Bulletin Board System**.

Berkeley Internet Name Domain (BIND)

The implementation of a DNS server developed and distributed by the University of California at Berkeley. Host name and address lookup service for the Internet; implemented in a client/server model. The client software, referred to as the *resolver*, allows client systems to obtain host names and addresses from servers rather than from locally hosted databases.

Berkeley Software Distribution (BSD)

The derivation of the original UNIX operating system developed by the Computer Systems Research Group of the Department of Electrical Engineering and Computer Science at the University of California at Berkeley. The DIGITAL UNIX operating system is based on the BSD version of UNIX.

best-effort delivery

A characteristic of network technologies that will attempt to deliver data but will not try to recover if there is an error such as a line failure. Internet protocols IP and UDP provide best-effort delivery service to application programs.

BG driver

The DIGITAL TCP/IP Services for OpenVMS implementation of a network device driver. *See also* **device driver**.

BGP

See **Border Gateway Protocol**.

big endian

The format for storage or transmission of binary data in which the most significant bit (or byte) comes first. The reverse convention is called **little endian**.

BIND resolver

A set of library routines compiled into a client application like `telnet` or `ftp` that formulates a query to ask a name server to look up name and address information.

BIND server

The software that responds to queries from BIND resolvers for name and address lookups; can be local or distributed. *See also* **cache server**, **forwarder server**, **primary server**, and **secondary server**.

binding

Defining a remote file system to be a part of the local OpenVMS file system.

bits per second (bps or b/s)

The measure of the rate of data transmission.

block

A contiguous unit of user information grouped together for transmission, such as the user data within a packet, excluding the protocol overhead.

boot file

A database file that BIND servers use to determine their type, the zones for which they have authority and the location of other BIND database files.

BOOTP

The mnemonic for Bootstrap protocol. The protocol used for booting diskless systems remotely to a network. *See also* **remote boot**.

BOOTP database

A DIGITAL TCP/IP Services for OpenVMS database with entries for diskless network clients that depend on a boot server to download their operating system images.

Border Gateway Protocol (BGP)

The interautonomous system routing protocol used to exchange network reachability information between autonomous systems; runs over TCP.

One of a class of exterior gateway protocols, described in more detail in the BGP section of UNIX reference page `gated.proto`(4).

bottleneck

A point in the network where traffic is delayed or blocked. Bottlenecks are the limiting factors in network performance.

bound port

An I/O function specifying a port number and IP address for the device socket to bind a port to a process.

bps

See **bits per second**.

bridge

A device that connects two or more physical networks and then stores and forwards complete packets between them; can usually be made to filter packets (that is, to forward only certain traffic).

broadband

A characteristic of any network that multiplexes multiple, independent network carriers onto a single cable; usually using frequency division multiplexing. Broadband technology allows several networks to coexist on one single cable; traffic from one network does not interfere with traffic from another because the "conversations" happen on different frequencies.

broadcast

A delivery system where a copy of a packet is sent simultaneously to many hosts; can be implemented with hardware (for example, as in Ethernet) or with software (for example, as in Cypress). *See also* **multicast**.

broadcast address

The address that designates all hosts on a physical network. The broadcast address contains a hostid of all ones.

broadcast addressing

A type of multicast addressing in which all nodes receive a message simultaneously.

broadcast circuit

A circuit on which multiple nodes are connected. A message can be transmitted to multiple receivers, and all nodes are adjacent.

broadcast end node adjacency

An end node connected to the same broadcast circuit as the local node. *See also* **adjacency**.

broadcast router adjacency

An intermediate system (router) connected to the same broadcast circuit as the local node. *See also* **adjacency**.

broadcast mask

A mask used to interpret the IP address as a broadcast address.

broadcast storm

An incorrect packet broadcast on a network that causes most hosts to respond all at once, typically with wrong answers that start the process over again.

brouter

A bridge/router; a device that forwards messages between networks at both network and data link levels.

BSD

See **Berkeley Software Distribution**.

Bulletin Board System (BBS)

A message database where people can log in and leave broadcast messages for others grouped (typically) into topic groups.

buffer

A device or an area of memory used for temporary storage when transmitting data from one device to another. Compensates for a difference in rate of data flow or in time of occurrence of events. Used on routing nodes to temporarily store data that is to be forwarded from one node to another.

buffering level

The number of buffers provided at one time by the network software to handle data. Level can be single or multiple. Single buffering tends to be less efficient than multibuffering but uses less memory on the local system. Multibuffering provides better performance, and a network can send or process several buffers of data in quick succession.

bus

(1) A LAN topology in which all nodes connect to a single transmission medium. All nodes are equal, and all nodes hear all transmissions on the medium. Bus topologies are reliable because failure of a node does not affect the ability of other nodes to transmit and receive. (2) A flat, flexible cable consisting of many transmission lines or wires used to interconnect computer system components to provide communication paths for addresses, data, and control information.

cache

A portion of a computer's RAM reserved to act as a temporary memory for items read from a disk. These items become instantly available to the user.

cache server

A BIND server that has no authority for any zone; acquires information in the process of resolving clients' queries and stores it in its cache. *See also* **BIND server**, **forwarder server**, **primary server**, and **secondary server**.

canonical name

The main or official name for a host; other names for the same host are aliases. In a BIND configuration, you specify the canonical name in a CNAME record of the `named.hosts` file.

category phrase

A BIND configuration logging statement phrase which specifies the different categories for which to log messages. Categories include: `config`, `parser`, `queries`, `lame-servers`, `statistics`, `panic`, `update`, `ncache`, `xfer-in`, `xfer-out`, `db`, `eventlib`, `packet`, `cname`, `security`, `os`, `insist`, `maintenance`, `load`, `response-checks`, and `default`.

catenet

The network in which hosts are connected to networks with varying characteristics, and the networks are interconnected by gateways.

centralized management

A form of network management that manages from a single point in the network.

CFS

See **Container File System**.

CFSRTL

See **Container File System RTL**.

channel

The data path between two or more stations, including the communications control capability of the associated stations.

channel phrase

A BIND configuration logging statement that specifies output methods, format options and severity levels associated with a category of messages to be logged.

checksum

A computed value based on the contents of a packet. The value is sent with the packet when it is transmitted. The receiving host computes a new value based on the received data. If the originating and receiving values are the same, the receiver has a high degree of confidence that the data was received correctly.

circuit

A logical (virtual) link that provides a communications connection between adjacent nodes.

class name

The name of an entity class. For example, `node` is the global entity class.

client

A computer system or process that requests a service of another computer service or process.

client/server relationship

A model of interaction used in distributed processing products when a client process sends a request and waits for the results from a server process.

clock

The combined hardware interrupt timer and software register that maintain system time. In many systems, the hardware timer sends interrupts to the operating system; at each interrupt, the operating system adds an increment to a software register that contains the time value.

cluster alias

An optional node name and address used by some or all nodes in an OpenVMS cluster, allowing these nodes to be reachable on the network with the same address.

cluster failover environment

An environment that allows a system in a cluster to take on the responsibilities of a system that crashed or is otherwise unavailable. For example, you can configure a system to become a DHCP server when the primary DHCP server process crashes or the system that the primary DHCP server is running on becomes unavailable.

collision

The condition in which two data packets are transmitted over a medium at the same time, making both unintelligible.

common address notation

The common way of expressing an Internet address. The 32-bit address uses four fields that are separated by periods; each field ranges from 0 to 255.

communications link

The physical medium connecting two systems.

communications server

A special-purpose standalone system dedicated to managing communications activities for other computer systems.

concatenation

The process of joining two or more items together, as when input files are appended to a new output file.

configuration database

The DIGITAL TCP/IP Services for OpenVMS database with SMTP, SNMP, and TIME specifications.

congestion

The condition in which a network or part of a network is overloaded and has insufficient communication resources for the volume of traffic.

connection

A logical communication path between two processes which are using the TCP protocol. The communication path must exist before data can be sent in either direction. A three-way handshake occurs between the requesting and receiving process to establish a port through which the two processes communicate.

connection-oriented

The model of interconnection that consists of three phases: establish connection, transfer data, and release connection. TCP is a connection-oriented protocol.

connectionless

The model of interconnection in which communication takes place without first establishing a connection. UDP, IP, and IPX are connectionless protocols.

connectivity

The degree to which network nodes are interconnected. Full connectivity means all nodes have links to every other node.

container file

A data file on a DIGITAL UNIX NFS server with a UNIX directory structure and UNIX file attributes for a local, logical UNIX-style file system. Each UNIX regular file is stored as a separate data file using. The directory data files in the container file contain the UNIX file names and a pointer to the corresponding OpenVMS Files-11 data file.

Container File System (CFS)

A logical UNIX-style file system that resides on a Files-11 formatted disk and is represented as a set of Files-11 files. *See also* **container file**.

Container File System RTL (CFSRTL)

The OpenVMS Run-Time Library (RTL) that is used by the NFS server to process files in the UNIX-style container file system.

contention

The condition when two or more stations attempt to use the same channel at the same time.

contention control

The scheme of access control used by many networks. Control is distributed among the nodes of the network. Any node wanting to transmit can do so, accessing the network on a first-come, first-served basis. However, it is possible that two nodes are in contention, or start transmitting at the same time, in which case a collision occurs. Each node must then back off and retransmit after waiting a random period of time.

control cluster

A group of small (256-byte) buffers dynamically allocated from nonpaged pool memory; stores information related to device sockets, internal control structures, IP addresses, Internet routes, and Internet packet headers.

Coordinated Universal Time (UTC)

Greenwich Mean Time

cost

An OSPF (Open Shortest Path First) protocol metric. See metric and OSPF.

counters

The performance and error statistics kept for an entity by network management, such as lines and nodes.

CRC

See **Cyclic Redundancy Check**.

Cyclic Redundancy Check (CRC)

An error detection scheme whereby a number is derived from a set of data before it is transmitted. Once transmitted, the receiving node recalculates the number and compares it to the value originally transmitted. If the numbers are different, some type of transmission error has occurred.

daemon

A process that executes in the background waiting for some event to occur.

data cluster

A group of large (1792-byte) buffers that store data in the system space; transmit and receive operations service user processes by moving data to and from data clusters.

Data Encryption Key (DEK)

Used for encryption of message text and (with certain choices among a set of alternative algorithms) for computation of message integrity check (MIC) quantities.

Data Encryption Standard (DES)

A type of encryption scheme approved by the U.S. National Bureau of Standards.

data link

A logical connection between two systems on the same circuit on which data integrity is maintained.

Data Link layer

The layer in a network model that handles communication between physical hosts.

data octet

See **octet**.

data overrun

The data blocks received that arrived too quickly to be processed by the receiver and were, therefore, lost.

datagram

A self-contained package of data carrying enough information to be routed from source to destination without reliance on earlier exchanges between source and destination or the transporting network.

datagram fragment

The result of fragmenting a datagram. Fragments carry a portion of data from the larger original and a copy of the original datagram header. The header fragmentation fields are adjusted to indicate the fragment's relative position within the original datagram.

datagram reassembly time

The time allowed for reassembly of a fragmented datagram.

datagram service

The mode of delivery for a datagram which is delivered in such a way that the receiver can determine the boundaries of the datagram as it was entered by the source.

DCE

See **Distributed Computing Environment**.

DCL

See **Digital Command Language**.

decision

The routing process that determines the path, or route, along which a data packet travels to reach its destination; forwards packets on the lowest-cost path even if that one does not have the fewest hops. The path that the data takes through the network is transparent to users.

decoding

The process by which the transfer syntax representation of a data value is transformed into the local representation of that value.

dedicated serial connection

A permanent connect between two hosts using an RS232 serial port. SLIP or PPP can be used for TCP/IP communication between the two hosts.

default route

The route used to direct any data addressed to network host addresses for which no explicit route is specified.

delay

A HELLO metric. Valid values are from zero to 30000, inclusive. The value of 30000 is the maximum metric and means unreachable. See metric and HELLO.

delete access

The access right that grants users the ability to remove data from the domain.

DEK

See **data encryption key**.

DES

See **Data Encryption Standard**.

designated router

In OSPF, a designated router is a multiaccess network that has at least two attached routers. The designated router generates a link state advertisement for the multiaccess network and assists in running the protocol. The designated router is elected by the HELLO protocol.

destination address

The IP address that specifies where a datagram is to be sent; contains the network and host identifiers.

Any network or host.

destination port

A 2-octet value in the TCP and UDP header field that identifies the destination upper-level protocol for a packet's data.

device driver

The software associated with each physical device; serves as the interface between the operating system and the device controller.

device socket

The extension of the pseudodevice, used for communications; consists of the Internet pseudodevice and the socket. *See also* **pseudodevice**.

DHCP

See **Dynamic Host Configuration Protocol**

dialogue

The sequence of message exchanges between open systems that represents a single association and the set of underlying connections.

dialup

A temporary (as opposed to dedicated) network connection established through a telephone line with a modem.

dialup provider

A host that responds to incoming PPP connection requests. A PPP server.

Digital Command Language (DCL)

The command interface of the OpenVMS operating system.

DIGITAL TCP/IP Services for OpenVMS

The Digital Equipment Corporation software product implemented on OpenVMS as an ancillary control process (ACP) and a network device driver (BG driver) with executive-level components and user applications that use TCP/IP protocols.

distance

An EGP metric. See metric and EGP. Valid values are from zero to 255 inclusive.

Distributed Computing Environment (DCE)

An architecture of standard programming interfaces, conventions, and server functions (for example, naming, distributed file system, remote procedure call) for transparently distributing applications across networks of heterogeneous computers.

distributed database

A collection of several different data repositories that look like a single database to the user. The Domain Name Service (DNS) is a distributed database.

distributed management

A form of network management in which network managers and management software are dispersed across many systems.

distributed processing

The technology that enables the distribution throughout the network of computing power and storage facilities to user work areas, such as offices, laboratories, or machines on factory floors.

distributed system

A collection of computer systems, tied together by communications networks for the purpose of sharing resources; end users do not need to be aware of the physical location of the shared resources.

DNS

See **Domain Name Service**.

domain

An organizational unit with administrative responsibility for naming networks or hosts. An internet domain name consists of a sequence of names (labels) separated by periods (dots); for example, `tundra.mpk.ca.us`.

domain name

The name used to refer to a fully qualified domain or subdomain. For example, in `cat.food.iams.com`, `food.iams.com`, `iams.com`, and `.com` are all domain names. Each name specifies a different domain level.

Domain Name Service (DNS)

A distributed database system that allows TCP/IP applications to resolve a host name into a correct IP address.

dot address

See **dotted decimal notation**.

dotted decimal notation

The syntactic representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them; used to represent IP addresses in the Internet, as in: 192.67.67.20. Many Internet application programs accept dotted decimal notation in place of destination machine names.

downline loading

Transferring a copy of a system image from a load host to a target. Some systems, such as DEC WANrouter systems and DECserver terminal servers, automatically request a downline load of their image upon startup and reboot. One of the functions of a TFTP server.

drift

The change in a clock's time rate over a specified period.

A measure, in hertz per second, of how quickly the skew of a clock is changing. See also skew.

dynamic adaptive routing

The automatic rerouting of traffic based on a sensing and analysis of current actual network conditions; not including cases of routing decisions taken on predefined information.

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP), a superset of the BOOTP protocol, enables the automatic assignment of IP addresses to clients on networks from a pool of addresses. The IP address assignment and configuration occurs automatically whenever appropriate client systems (workstations and portable computers) attach to a network. The TCP/IP Services for OpenVMS implementation of DHCP is based on the JOIN product by Competitive Automation.

dynamic routing

A type of routing where a host or router talks to adjacent routers to learn what networks each router is connected to. Subsequently, the kernel's routing tables are updated when the router learns new information. There are many routing protocols including Interior Gateway Protocols (RIP, OSPF) and Exterior Gateway Protocols (EGP and BGP).

ephemeral port number

A port number temporarily assigned to a client process for the duration of a session. When the client process terminates, the port number can be assigned to another process. The port number is usually between 1024 and 5000.

EGP

See **Exterior Gateway Protocol**.

elective protocol

The classification in Internet standards for optional protocols.

electronic mail

The service whereby a computer user can exchange messages with other computer users (or groups of users) by means of a communications network; one of the most popular uses of the Internet.

email

See **electronic mail**.

encapsulation

A technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer below. As an example, in Internet terminology, a packet would contain a header from the physical layer, followed by a header from the Network layer (IP), followed by a header from the Transport layer (TCP), followed by the application protocol data.

encryption

A process of encoding information so the meaning of its content is no longer immediately obvious to anyone who obtains a copy of it.

end node

See **end system**.

end system

A nonrouting system; can receive data packets addressed to it and send data packets to other systems on the same subnet but cannot relay, route, or forward data packets to other systems.

entity

An individual, manageable piece of a network; has attributes that describe it, a name that identifies it, and an interface that supports management operations.

entity class

A collection of entities that share the same properties and have the same parent entity; each member of the class has a unique identifier within the class. Entity classes have class names.

entity group

An architecturally defined collection of entities. The entities in the group must have a common top entity and must all be of the same class.

entity hierarchy

A logical hierarchical tree structures of manageable entities in which child entities are below their parent entities. Children can be accessed only through their parents' agent.

entity identifier

An attribute that specifically identifies an entity. *See also* **attribute group**.

entity name

A label associated with some entities used to identify or locate them for management purposes.

entity type

The subgrouping of an entity that determines its relationship to other entities.

Ethernet

A baseband network medium. Commonly used to connect a local area network.

event

A measurable network- or system-specific occurrence for which a logging component maintains a record.

experimental protocol

The classification in Internet standards for protocols that are developed as part of an ongoing research project not related to an operational service offering; not intended for operational use.

export database

The DIGITAL TCP/IP Services for OpenVMS database with directory names that can be mounted from remote NFS clients.

exported file

A file in an exported directory or a subdirectory of an exported directory. *See also* **exporting**.

exported file systems

A file system that can be accessed by a remote system using the Network File System. The local system imports the remote file system. Both the remote and local system must be configured to grant and receive access to the file system.

exporting

Identifying a directory on an NFS server that can be remotely mounted by NFS clients.

extended LAN

Multiple LANs connected with data link relays or bridges.

Exterior Gateway Protocol (EGP)

The protocol that distributes routing information to the gateways that interconnect networks.

A class of routing protocols used to exchange routing information within an autonomous system. A detailed explanation of exterior gateway protocols is available in `gated.proto`(4).

One of a class of exterior gateway protocols, described in more detail in the EGP section of `gated.proto`(4).

FDDI

See **Fiber Distributed Data Interface**.

fetch/store operation

The operation of two commands that allow a system manager to fetch a value from a data item or to store a value into a data item.

Fiber Distributed Data Interface (FDDI)

The high-speed (100 mb/s) networking standard based on fiber optics, established by the American National Standards Institute (ANSI); uses 1300 nanometer light wavelength. FDDI networks are limited to approximately 200 km in length, with repeaters every 2 km or less.

file

A uniquely named collection of information with shared managerial and structural properties.

file attribute

The characteristic of a file, such as its size or creation date. The values of some file attributes may change during the lifetime of a file.

file data

The information that is stored within a file and comprises its contents (as opposed to its attributes).

file designation

System-specific information that identifies a file on its storage system.

file server

The host whose principal purpose is to store files and provide network access to them.

file specification

System-specific information that identifies a file on its storage system.

file system

A method for recording, cataloging, and accessing files on a volume.

File Transfer Protocol (FTP)

The protocol and software that permit a user on one host to access and transfer files to and from another host over a network. *See also* **Trivial File Transport Protocol**.

Files-11 ODS level 2 structure

The set of rules that govern the organization of the OpenVMS file system, external to the files themselves.

Finger utility

The utility that provides information about users on local and remote systems.

flow control

(1) The function of a receiving entity to limit the amount or rate of data that is sent by a transmitting entity. (2) The control of the rate at which hosts or gateways inject packets into a network or Internet, usually to avoid congestion. Flow control mechanisms can be implemented at various levels and allow communicating layers to match their data transfer and receive rates. Simplistic schemes, like ICMP source quench, simply ask the sender to cease transmission until congestion ends. More complex schemes vary the transmission rate continuously.

forwarder server

The name server that processes recursive requests that a slave server cannot resolve locally; has access to the Internet. *See also* **BIND server**, **cache server**, **primary server**, **secondary server**, and **slave server**.

forwarding information base

The table that GATED uses internally to store routing information it learns from routing protocols is a routing table, also known as a routing information base, or RIB. The routing table is used to collect and store routes from various protocols.

forwarding table

The table in the kernel that controls the forwarding of packets is a forwarding table, also known as a forwarding information base, or FIB.

FQDN

See **fully qualified domain name**.

fragment

A piece of a packet that results from a router dividing an IP datagram into smaller pieces for transmission across a network that cannot handle the original datagram size. Fragments use the same format as datagrams; fields in the IP header declare whether a datagram is a fragment and, if so, where the data in the fragment occurred in the original datagram. IP software at the receiving end must reassemble the fragments. *See also* **maximum transmission unit**.

fragmentation

The IP process of breaking up packets into smaller packets for transmission; allows a packet originating in a network that allows a large packet size to traverse a network that limits packets to a smaller size. The destination host reassembles the fragments. *See also* **maximum transmission unit**.

frame

A Data Link layer packet that contains the header and trailer information required by the physical medium

FTP

See **File Transfer Protocol**.

full-duplex circuit

A circuit designed for transmission in both directions at the same time. *Contrast with* **half-duplex circuit**.

full-duplex transmission

Data transmission in both directions at the same time. *Contrast with* **half-duplex transmission**.

fully qualified domain name (FQDN)

The full site name of a system such as `warren.enet.dec.com`, rather than just its host name —`warren`.

function code

A parameter in a $QIO system service call that defines the specific function of that $QIO.

GATED

A routing daemon that can be configured to route one or more of the following protocols: RIP, BGP, EGP, HELLO and OSPF.

gateway

A communications device or program that passes data between networks having similar functions but dissimilar implementations. The term "router" is now used in place of the original definition of "gateway."

1. An intermediate destination by which packets are delivered to their ultimate destination. 2. A host address of another router that is directly reachable via an attached network. As with any host address it may be specified symbolically.

gateway client

Another term for an access system.

gateway routing daemon

See **GATED**

GID

See **group identification**.

gigabit

One billion bits

gigabyte

One billion bytes

group identification (GID)

The identification code for a group of UNIX users.

half-duplex circuit

A circuit designed for transmission in either direction, but only one direction at one time. *Contrast with* **full-duplex circuit**.

half-duplex transmission

Data transmission in either direction, but only one direction at a time. *Contrast with* **full-duplex transmission**.

handshaking sequence

The exchange of connection information between two communicating entities; takes place to enable the successful completion of a connection. Used, for example, in establishing a TCP connection between client and server applications.

hardware address

The address that identifies the connection device between the network controller of a host and the network cable. *See also* **address**.

hard link

A mechanism that allows you to assign more than one name to a file. Both the new name and the file being linked must be in the same file system. *See* **link**.

header

The portion of a packet that precedes the actual data and contains control information such as source and destination address and error checking.

header compression

A technique used by PPP and SLIP protocols to reduce the number of bytes per frame when sending over a slow serial link. The use of header compression is negotiated between the client and servers processes to reduce the size of the IP and TCP headers.

HELLO

One of a class of interior gateway protocols, described in more detail in the HELLO section of gated.proto(4).

heterogeneous network

A network consisting of different network protocols or different operating system software, such as OpenVMS and UNIX.

hierarchical routing

Routing based on domains. Interdomain routers are responsible only for getting data to the right domain and intradomain routers take responsibility for routing within the domain.

hop count

The number of connections between two hosts, based on the number of different routers needed to traverse the distance between the two hosts.

hop

A term used in routing. Number of hosts separating a source and final destination (including the final destination) on a network.

host

A computer system that acts as a source or destination of network messages sometimes called "node."

The IP address of any system, usually specified as a dotted quad four values in the range of 0 to 255, inclusive, separated by dots (.). For example 132.236.199.63 or 10.0.0.51. It can also be specified as an eight digit hexadecimal string preceded by 0x. For example, 0x0a000043. In addition, if the options noresolv statement is not specified, this can be a symbolic host name. For example, gated.cornell.edu or nic.ddn.mil. The numeric forms are preferred over the symbolic form.

host address

See host number.

hosts database

The DIGITAL TCP/IP Services for OpenVMS database that is created by default; allows users to use host names; contains host names, IP addresses of the hosts, and any alias names for the hosts.

host name

The name given to a network host. *See also* **fully qualified domain name** and **alias**.

host number

The part of an IP address that identifies which host on the network is being addressed.

Host-to-Host Communication layer

Also called Transport layer. The second-highest level in the Internet architecture model; provides end-to-end communication services, including mechanisms such as end-to-end reliability and network control. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) reside in this layer.

IAB

See **Internet Architecture Board**.

IBM TN3270

The TELNET options that allow TELNET users to connect to hosts that support 3270 terminals.

ICMP

See **Internet Control Message Protocol**.

idempotent

IETF

The Internet Engineering Task Force. A large international community of network designers, operators, vendors and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Membership is open to everyone. See `http://www.ietf.org/` for more information.

IGP

See **Interior Gateway Protocol**.

imported file

A file within a local NFS server that has been copied or linked to a remote NFS client.

`inetd`

A UNIX internet daemon. A server process listens for client requests for specific services. When `inetd` receives a request for a service, it starts the appropriate server process.

initial sequence number

The first sequence number used for sending or receiving on a connection.

inode

A UNIX file structure used to address a file block. There is a unique inode allocated for each active file with a name comprised of a device/i-number pair.

interface

The boundary between two parts of a system across which communication is possible; may be defined through hardware or software.

The host address of an attached network interface. This is the address of a broadcast, nbma, or loopback interface, and the remote address of a point-to-point interface. As with any host address it can be specified symbolically.

The connection between a router and one of its attached networks. A physical interface may be specified by a single IP address, domain name, or interface name. (Unless the network is an unnumbered point- to-point network.) Multiple levels of reference in the configuration language allow identification of interfaces using wild card, interface type name, or delete word address. Be careful with the use of interface names as future versions might allow more than one address per interface. Dynamic interfaces can be added or deleted and indicated as up or down as well as changes to address, netmask and metric parameters.

Interior Gateway Protocol (IGP)

The protocol used to propagate network reachability and routing information within an autonomous system; RIP is among the most popular.

One of a class of routing protocols used to exchange routing information within an autonomous system. A detailed explanation of interior gateway protocols is available in gated.proto(4).

interface list

A list of one or more interface names, including wildcard names (names without a number) and names that may specify more than one interface or address, or the token all for all interfaces. See gated.conf(4) for more information.

intermediate system

An OSI system that performs Internet layer forwarding. A routing system receives data packets from a system on one subnet and passes them on to a system on another subnet; it receives data packets from a source end system, or from the previous intermediate system on the route, and passes them on to the destination end system, or to the next intermediate system on the route.

internet

A shortened form of internetwork; a network of networks; interconnected TCP/IP networks that function as one large virtual network. Differs from the Internet by their lack of connectivity with the global Internet.

Internet

The worldwide network of networks and gateways that use the TCP/IP protocol suite and function as one virtual network; provides universal connectivity and three levels of network services: unreliable, connectionless packet delivery; reliable, full-duplex stream delivery; and application level services such as electronic mail that build on the first two. The Internet connects many universities, government research labs, military installations, and private businesses.

Internet architecture

A four-layered communications model that consists of the following: Application layer, Transport layer, Internet layer, and Network Interface layer.

Internet Architecture Board (IAB)

The technical body that oversees the development of the Internet suite of protocols (commonly referred to as "TCP/IP"). It has a research task force and an engineering task force, each responsible for investigating a particular area.

Internet Autonomous System

A system that consists of a set of gateways, each of which can reach any other gateway in the same system using paths by means of gateways only in that system. The gateways of a system cooperatively maintain a routing database using an interior gateway protocol.

Internet Control Message Protocol (ICMP)

An extension to the Internet Protocol; used by gateways to communicate with the network software in hosts.

Internet header length

An IP header field that indicates the number of 32-bit words making up the Internet header.

Internet layer

The layer in the TCP/IP network model where data is transferred between hosts across networks. Also referred to as Network Interface layer.

Internet number

See **IP address**.

Internet Protocol (IP)

A connectionless best-effort packet switching protocol that resides in the Internet layer and has two major functions: internet addressing and fragmentation of messages.

Internetwork

A collection of many different computing systems which communicate with each other. The computing systems can include different hardware architectures, operating systems and network technologies.

interoperability

The ability of software and hardware on multiple machines from multiple vendors to communicate meaningfully.

InterNIC Registration Services

The Internet Network Information Center; organization that provides the Internet community with registration, directory, database and information services.

I/O status block (IOSB)

A data structure associated with the $QIO system service. The IOSB holds information about how the I/O request completes.

IP

See **Internet Protocol**.

IP address

An address that identifies the connection between the network controller of a node using TCP/IP and the network cable. The 32-bit address is composed of two parts: network number and host number.

IP datagram

The basic unit of information passed across the Internet; contains source and destination addresses, the data, and fields that define the length of the datagram, the header checksum, and flags to say whether the datagram can be (or has been) fragmented. An IP datagram is to the Internet what a hardware packet is to a physical network. *See also* **datagram**.

IP forwarding

A configurable kernel options that controls whether a host forwards IP datagrams. Generally, hosts do not forward IP datagrams.

IP trailer protocol

A protocol in which the protocol header follows the data.

KA9Q

A popular implementation of TCP/IP and associated protocols for amateur radio systems.

Kbps

See **Kilobits per second**.

kernel

The software that provides the standard API for application programs. Generally speaking, the kernel embodies the policy and structure of an operating system. In a narrower sense, the kernel provides a programmatic interface to any hardware resources available. In a UNIX system, the kernel is a program that contains the device drivers, the memory management routines, the scheduler, and system calls; always running while the system is operating.

Kilobits per second (Kbps or Kb/s)

The measure of data transmission rate.

LAN

See **local area network**.

layer

(1) The grouping of related communication functions that provide a well defined service to a client independently of the protocols and other means used to provide it. (2) A software protocol levels that make up network architectures; each layer performs certain functions for the layers above and below it.

limited use protocol

A classification in Internet standards for protocols that are for use in limited circumstances; possibly due to their experimental state, specialized nature, limited functionality, or historic state.

line printer daemon (LPR/LPD)

The DIGITAL TCP/IP Services for OpenVMS remote printing services for UNIX and OpenVMS client hosts.

line speed

The maximum rate at which data can be reliably transmitted over a line; varies with the capability of the modem or hardware device that performs the transmitting.

link

A directory entry referring to a file; one file may have several links to it.

little endian

The format for storage or transmission of binary data in which the least significant byte comes first. The reverse convention is called **big endian**.

load broker

A TCP/IP Services component that provides configurable, calculated methods for distributing BIND services among systems in a cluster.

local address

The address of a host within a subnet.

The host address of an attached interface. This is the address of a broadcast, nbma, or loopback interface, and the local address of a point-to-point interface. As with any host address it may be specified symbolically.

local area network (LAN)

A self-contained group of computers and communications devices (such as modems, routers, servers, and repeaters) that offers a high-speed, reliable communications channel. LANs span a limited distance such as a building or group of buildings, but can be connected to wide area networks (WANs) with gateways. *Contrast with* **wide area network (WAN)**.

local data

Any data stored locally by a system.

local network

A network directly attached to a host or gateway.

local node

A node at which the user is located.

local subnet

A subnet directly attached to a host or gateway.

lock manager

An NFS component that allows an NFS client to lock portions of files that reside on an NFS server.

logical connectivity

The ability of nodes to communicate.

logical link

A temporary connection between processes on source and destination nodes (or between two processes on the same node).

Logical Link Control

The upper portion of the Data Link layer that presents a uniform interface to the user of the data link service, usually the Internet layer.

loop node

A local node that is associated with a particular address and is treated as if it were a remote node. All traffic to the loop node is sent over the associated address; used for loopback testing.

loopback

A program that sends packets to a remote host on the Internet and looks for replies; works by means of the echoing facility provided by the ICMP protocol and is a way to determine if an Internet host is reachable from your host. *See also* **packet internet groper**.

LPR/LPD

See **remote line printing** or **line printer daemon**.

mail bridge

A mail gateway that forwards electronic mail between two or more networks while ensuring that the messages it forwards meet certain administrative criteria; specialized form of mail gateway that enforces an administrative policy with regard to what mail it forwards.

mail exchange record (MX record)

The Domain Name Service resource record type indicating which host can handle mail for a particular domain or host.

mail exchanger (MX)

The DIGITAL TCP/IP Services for OpenVMS implementation of a mail exchanger that allows hosts in a local network to forward mail to systems that might not be directly connected to the local network.

mail exploder

The part of an electronic mail delivery system that allows a message to be delivered to a list of addressees. Users send messages to one address (e.g., hacks@somehost.edu) and the mail exploder handles delivery to the individual mailboxes.

mail gateway

A host that connects two or more electronic mail systems (especially dissimilar mail systems on two different networks) and transfers messages between them.

mail path

A series of hosts used to direct electronic mail from one user to another.

Management Information Base (MIB)

A database used by the Simple Network Management Protocol (SNMP) to check network statistics and configurations. An SNMP management station can query a MIB or set it in an SNMP agent (for example, router). Standard, minimal MIBs have been defined (MIB I, MIB II), and vendors often have custom entries. In theory, any SNMP manager can talk to any SNMP agent with a properly defined MIB.

Management Information Base II (MIB II)

Data that can be accessed by a network management protocol; for DIGITAL TCP/IP Services for OpenVMS, the database maintained by a gateway running SNMP.

management station

The workstation of a human network manager running SNMP.

mask

A means of subdividing networks using address modification. A mask is a dotted quad specifying the bits of the destination that are significant. Except when used in a route filter, gated only supports contiguous masks.

mask length

The number of significant bits in the mask.

master file directory (MFD)

The root of an OpenVMS file system on a particular physical device.

master server

The name server that is the authority for a specific domain space. *See also* **BIND server**.

maximum transmission unit (MTU)

The largest possible unit of data that can be sent on a given physical medium. *See also* **fragmentation**.

MBUFs

See **memory buffers**.

memory buffers (MBUFs)

The portions of memory that act as queues for data arriving at a port before the process is ready to claim that data.

message

A message block, or a series of message blocks, that constitute a logical grouping of information; each is delimited by communications control characters.

metric

One of the units used to help a system determine the best route. Metrics may be based on hop count, routing delay, or an arbitrary value set by the administrator depending on the type of routing protocol. Routing metrics may influence the value of assigned internal preferences. (See preference.)

The following sample table shows the range of possible values for each routing protocol metric and the value used by each protocol (See gated.proto(4)) to reach a destination:

SAMPLE ROUTING PROTOCOL METRICS

Protocol	Metric Represents	Range	Unreachable
RIP	distance (hop-count)	0-15	16
HELLO	delay (milliseconds)	0-29999	30000
OSPF	cost of path	0-?????	Delete
EGP	distance (unused)	0-65535	255
BGP	unspecified	0-65534	65535

MFD

See **master file directory**.

MIB

See **Management Information Base**.

MIB II

See **Management Information Base II**.

MIME

Multipurpose Mail Extensions; a specification for the transfer of nontext files with regular Internet e-mail.

mode

A protection placed on a file.

modem (modulator/demodulator)

A device that translates digital signals (electrical impulses) generated by a computer into analog signals (tones) that can be transmitted over telephone lines, and vice versa.

mount

An NFS process that makes a remote directory available to local users.

mount point

A directory on an NFS client which is associated with a remote file system. The directory must exist before NFS can use it as a mount point.

MTU

See **maximum transmission unit**.

multiaccess networks

Those physical networks that support the attachment of multiple (more than two) routers. Each pair of routers on such a network is assumed to be able to communicate directly.

multicast

A transmission of network traffic intended for multiple hosts (but not all connected hosts) within a network or internet.

multicast address

An address that designates a subset of nodes that are all listening for packets destined to this address.

multicast addressing

An addressing mode in which a data packet is targeted to a group of nodes that are of the same type, for example, all level 1 routers or all level 2 routers.

multihomed host

A host that has two or more hardware connections to a network; requires multiple IP addresses.

multiplexing

Using a single connection to carry several data streams and the mechanism for assigning these streams to that connection.

multipoint circuit

A circuit that connects multiple systems.

multiprocessing system

A network consisting of multiple processors.

MX record

See **mail exchange record**.

NAK

See **negative acknowledgment**.

name resolution

The process of mapping a host name to its corresponding address. *See* **Domain Name Service**.

named

The BIND Name Server daemon.

namespace

A commonly distributed set of names in which all names are unique.

negative acknowledgment (NAK)

The response to receipt of a corrupted packet of information. *See also* **acknowledgment**.

neighbor

Another router with which implicit or explicit communication is established by a routing protocol. Neighbors are usually on a shared network, but not always. This term is mostly used in OSPF and EGP. Usually synonymous with peer.

neighboring routers

Two routers that have interfaces to a common network. On multiaccess networks, routers are dynamically discovered by OSPF's HELLO protocol.

network

A group of computer systems that can communicate with each other; can be composed of computers in a single building (local area networks or LANs), or computers thousands of miles apart (wide area networks or WANs). The Internet is a worldwide collection of computer networks that can intercommunicate.

Any packet-switched network. A network may be specified by its IP address or network name. The host bits in a network specification must be zero. Default may be used to specify the default network (0.0.0.0).

The IP address of a network. Usually specified as a dotted quad, one to four values in the range of 0 to 255 inclusive separated by dots (.). For example, 132.236.199, 132.236, or 10. It may also be specified as a hexadecimal string preceded by 0x with an even number of digits between two and eight. For example, 0x??????, 0x???? or 0x0a. Also allowed is the symbolic value default that has the value 0.0.0.0, the default network. If options noresolv statement is not specified, this can also be a symbolic network name. For example,nr-tech-prod, cornellu-net, and arpanet. The numeric forms are preferred over the symbolic form.

network address

A unique identifier of a specific system on a network, usually represented as a number or series of numbers. *See also* **IP address**.

network architecture

The specification of a network's functions and its parts, together with the ways in which the network is organized; specifies the layers of different functions in the network, ranging from data transmission at the lowest levels to user applications at the highest levels.

network byte order

The order in which bytes of information are sent or received by network applications as oppossed to how the bytes are stored in memory by different operating systems and hardware architectures. The standard network byte order is Big Endian.

network class

A definition of the type of network addressing scheme being used; high-order bits in the network number designate the network class of the IP address.

network database

The DIGITAL TCP/IP Services for OpenVMS database that allows users to refer to networks by name rather than network number; contains network names, IP addresses for the networks, and any alias names for the networks.

network delay

The time it takes to get a unit of data from the source of a transmission to the destination; usually refers to delay from the network and not by system-dependent application processing delays at source and destination nodes.

A HELLO metric. Valid values are from zero to 30000, inclusive. The value of 30000 is the maximum metric and means unreachable. See metric and HELLO.

network diameter

The distance (number of hops) between the two nodes in the network with the greatest reachability distance. The reachability distance is the path with fewest number of hops between two nodes.

Network File System (NFS)

A protocol developed by Sun Microsystems that allows a computer system to access files over a network as if they were on its local disks.

Network Information Service (NIS)

A set of services in the Network File System that propagate information out from masters to recipients; used for the maintenance of system files on complex networks.

Network Interface

A device driver that communicates with the IP layer of the TCP/IP protocol suite and the network interface card.

Network Interface layer

The layer in the TCP/IP architecture model that provides the mechanism for connecting the hosts to the networks.

network management

See **MIB II** and **Simple Network Management Protocol (SNMP)**.

network mask

A mask used to determine the subnet in the IP address; each bit that is turned on (binary one) in the mask is interpreted as part of the network and subnet address. Synonymous with subnet mask.

A means of subdividing networks using address modification. A mask is a dotted quad specifying the bits of the destination that are significant. Except when used in a route filter, gated only supports contiguous masks.

network

Any packet-switched network. A network may be specified by its IP address or network name. The host bits in a network specification must be zero. Default may be used to specify the default network (0.0.0.0).

network

The IP address of a network. Usually specified as a dotted quad, one to four values in the range of 0 to 255 inclusive separated by dots (.). For example, 132.236.199, 132.236, or 10. It may also be specified as a hexadecimal string preceded by 0x with an even number of digits between two and eight. For example, 0x??????, 0x???? or 0x0a. Also allowed is the symbolic value default that has the value 0.0.0.0, the default network. If options noresolv statement is not specified, this can also be a symbolic network name. For example,nr-tech-prod, cornellu-net, and arpanet. The numeric forms are preferred over the symbolic form.

network meltdown

The state of complete network overload; the network equivalent of thrashing. *See also* **broadcast storm**.

network number

The part of an IP address that designates the network to which the destination host belongs.

network performance

The description of how a network performs, as measured against the expectations or requirements of users, customers, designers, or implementors, or as claimed by sales and marketing personnel. The criteria for network performance include parameters such as throughput, response time, and resource utilization.

network status notification

Information about the state of logical and physical links over which two tasks communicate. A nontransparent task can use this information to take appropriate action under conditions such as third-party disconnections and a partner's exiting before I/O completion.

network task

A nontransparent task that can process multiple inbound connection requests; that is, it has a declared network name or object number.

Network Time Protocol (NTP)

The protocol that ensures accurate local timekeeping with reference to radio and atomic clocks located on the Internet; capable of synchronizing distributed clocks within milliseconds over long time periods.

NFS

See **Network File System**.

NFS client

The software that requests remote file services from an NFS server. Client system users access files that physically reside on an NFS server system.

NFS server

The software that provides remote file services to NFS clients.

NFS server (OpenVMS server)

A computer system that offers services to NFS clients within an Internet environment; can be a single host, a whole OpenVMS cluster system, or members of an OpenVMS cluster system.

NIS

See **Network Information Service**.

nobody

A UNIX convention used when file ownership is not known; maps to an account with a UID and GID of –2.

node

(1) A system on a network; also referred to as a host. (2) One member in an OpenVMS cluster system.

node address

The required unique numeric identification of a specific node in the network.

node name

The alphanumeric identification associated with the node address for one-to-one mapping.

nonadjacent nodes

Nodes without direct lines between them; can communicate only if intermediate systems forward the data along the path between the source and the destination.

nonauthoritative answer

A name server's answer is said to be nonauthoritative when the server answer comes from its own cache.

nontransparent task

A form of device-dependent I/O that uses system services for network-specific functions; can initiate and complete a logical link connection, exchange messages between two tasks, and terminate the communication process. Application that has direct access to network-specific information and operations, such as optional user data on connects and disconnects and interrupt messages, to monitor the communications process; can receive and process multiple inbound connection requests.

normalization

The estimation of the change in a counter value over a specified time period.

nslookup

The DIGITAL TCP/IP Services for OpenVMS utility that allows you to interactively query domain name servers (BIND servers) and helps you set up and manage the BIND server software.

NTP

See **Network Time Protocol**.

NTP packet

A message sent over the network that conforms to the Network Time Protocol format. This format includes space for recording the current time. See also "poll".

null modem

A simple form of modem connection where only the data interchange circuits, and not the modem control circuits, are used.

occluded mounting

A TCP/IP Services/NFS method of mouting an NFS file system onto a client mount point that is higher or lower in the directory structure than an exisitng active mount.

octet

A single 8-bit unit of data; used in networking (rather than bytes) because some systems have bytes that are not 8 bits long.

OPCOM

See Operator Communication Manager

OPCOM messages

Messages broadcast by the Operator Communication Manager (OPCOM). These messages are displayed on operator terminals and written to the operator log file. The messages might be general messages that you send, user requests, operator replies, or system events.

OPCOM process

The system process that manages Operator Communication Manager (OPCOM) operations.

open network

A network made up of nonproprietary, interoperable systems.

Operator Communication Manager

A system administration tool for communicating with users and operators on the system.

OSPF (Open Shortest Path First)

One of a class of interior gateway protocols, described in more detail in the OSPF section of gated.proto(4).

open system

A nonproprietary, interoperable system with communications software.

Open System Interconnection (OSI)

A suite of protocols, designed by ISO committees, to be the international standard of computer network architecture.

OpenVMS cluster

A configuration of OpenVMS processors.

OpenVMS cluster alias

An alias that allows remote hosts to address the cluster members as one host, as well as any cluster member individually.

OpenVMS file system

The OpenVMS files and directories on a mounted OpenVMS volume. These files and directories reside on a Files-11 On-Disk Structured (ODS-2) disk.

origination

The beginning point of communications on a circuit.

overmounting

The process of NFS mounting another directory over an exisiting mount point. The original file system is dismounted from the mount point and the new file system is mounted.

packet

A unit of data sent across a network.

Packet Internet Groper (PING)

A program used to test reachability of a destination by sending an ICMP echo request and waiting for a reply. *See also* **loopback**.

packet looping

A condition in which a packet revisits a node. *See also* **aged packet**.

packet size

The amount of data in a packet.

packet switching

A communication paradigm in which packets are individually routed between hosts, with no previously established communication path.

path

The physical lines between source nodes and destination nodes; can comprise a sequence of connected nodes. The path that the data takes through the network is transparent to users.

path cost

The sum of the circuit costs along a path between two nodes.

An OSPF (Open Shortest Path First) protocol metric. See metric and OSPF.

path length

The total distance (the number of circuits) between a source node and a destination node, measured in hops. Each line between systems, including routing nodes and end nodes, equals one hop. *See also* **network diameter**.

path name

A unique designation that identifies a directory or subdirectory. UNIX path names are composed of a series of fields separated by slashes (/); each field designates a file name that is uniquely contained in the previous field (directory).

path MTU

The smallest MTU of any data link that packets traverse between two hosts. The path MTU depends upon the route being used at the time. Therefore, the sending path MTU may differ from the receiving path MTU.

path MTU discovery

A mechanism to determine the path MTU at any one time.

path splitting

The ability to split the transmission load destined for a single node over several paths of equal path cost. Any destination node receiving data that has been split over several paths must support out-of-order packet caching.

PC-NFS Daemon

The server software that handles authentication and printing requests from personal computer implementations of NFS.

peer

Another router with which implicit or explicit communication is established by a routing protocol. Peers are usually on a shared network, but not always. This term is mostly used by BGP. Usually synonymous with neighbor.

physical address

A unique address of each physical connection of a node to the physical medium.

physical connection

The Physical layer communications path between two systems.

physical connectivity

The Physical layer connectivity that is a result of nodes being attached to each other via active lines and nodes.

PING

See **Packet Internet Groper**.

point-to-point circuit

A circuit that connects only two nodes. A point-to-point configuration requires a separate physical connection between each pair of nodes. Point-to-point systems communicate directly with other systems. *Contrast with* **multipoint circuit**.

point-to-point line

A line that connects two systems by using a single circuit.

Point-to-Point Protocol (PPP)

A method for transmitting datagrams over serial point-to-point lines where a line is established between a remote host (usually over a telephone line) and another host acting as a gateway to a remote host.

poll

The sending of an NTP packet from a host to an NTP time server to request the current time. The server responds by recording the current time in the packet, then sending it back to the originating host. See also "NTP packet".

polling

Connecting to another system to check for things such as mail or news.

POP

See **Post Office Protocol**.

port

The end point of a communication link between two processes.

A UDP or TCP port number. Valid values are from 1 through 65535 inclusive.

port number

A 16-bit number used to identify applications using TCP or UDP. The number is stored in the transport layer protocol headers to identify the application.

port assignment

Portmapper Service

A service that client programs can use to determine the port number that another service uses. Clients use the Portmapper Service for NFC, PC-NFS, and RPC applications.

post

To send a message to a mailing list or newsgroup. Distinguished in context from "mail."

Post Office Protocol (POP)

The TCP/IP-based protocol for client stations to read mail from a server.

PPP

See **Point-to-Point Protocol**.

PPP client

A host requiring a temporary PPP connection to a dialup provider or a terminal server.

PPP dialup provider

A host that answers modem calls from PPP clients, assigns IP addresses and establishes PPP connections initiated by PPP clients.

preference

A preference is a value between 0 (zero) and 255 used to select between many routes to the same destination. The route with the best (numerically lowest) preference is selected as the active route. The active route is the one installed in the kernel forwarding table and exported to other protocols. Preference zero is usually reserved for routes to directly attached interfaces. A default preference is assigned to each source from which gated receives routes. (See Preference.)

prefix

A contiguous mask covering the most significant bits of an address. The prefix length specifies how many bits are covered.

primary server

A BIND name server that maintains the database for a zone; secondary servers copy their information from primary servers. *See also* **BIND server**, **cache server**, **forwarder server**, and **secondary server**.

printcap database

The DIGITAL TCP/IP Services for OpenVMS database that maps local queues to printers on remote hosts; specifies local queues for LPD printing from remote hosts. Equivalent to the UNIX `/etc/printcap` file.

privileged port

A port in which the remote host has done some level of checking against the application using the port; privileged port numbers range from 1 to 1023.

Reviewer Note

Is this terminology correct? Well-known ports are 1 - 1023. UNIX has reserved ports 1-1023.

process

The context within a system in which a specific computing session occurs; provides the context in which an application executes.

protocol

A set of rules that controls the communications between computers. Also, a set of conventions between communicating processes regarding the format and contents of messages to be exchanged.

Protocols can describe low-level details of machine-to-machine interfaces, such as the order in which the bits from a byte are set across a wire, or high-level exchanges between applications programs such as the way in which two programs transfer a file across the Internet.

protocol data unit (PDU)

The unit of data sent across a network. Also called a **packet**.

protocol machine

The set of data structures and routines that implements a specific protocol and controls the progress of a communication between peer entities.

protocol overhead

The part of communications data or processing not directly consumed by the users but necessary to successfully bring about the transfer of user information.

protocol port

An abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. Internet protocols identify ports using small positive integers. Usually the operating system allows an application program to specify which port it wants to use. Some ports are reserved for standard services such as electronic mail.

protocol transparency

The quality in a communications device or system that allows various higher-level protocols to coexist on the same wire. The protocols are transparent to the device or system.

protocol sequence

An ordered list of protocol identifiers.

protocol stack

The set of functions, one at each layer of the protocol stack, that work together to form a set of network services; each layer of the protocol stack uses the services of the module beneath it.

protocol transparency

The degree to which users of underlying protocols are aware of the specifics of those protocols.

proxy

The mechanism whereby one system acts on behalf of another system in responding to protocol requests. DIGITAL TCP/IP Services for OpenVMS uses a proxy mechanism to provide an OpenVMS identity (account) for each UNIX client by adding the name and identification codes of the client to a proxy database.

proxy ARP

The technique in which one machine, usually a router, answers Address Resolution Protocol (ARP) requests intended for another machine. By "faking" its identity, the router accepts responsibility for routing packets to the "real" destination. Proxy ARP allows a site to use a single IP address with two physical networks. Creating a subnet would normally be a better solution.

proxy database

The database that provides OpenVMS identities for remote NFS clients and UNIX-style identities for local NFS client users; provides proxy accounts for remote processes.

pseudodevice

A software device used to implement special-purpose transports and not directly associated with hardware.

pseudo-interface

A method of extending subnet routing using a network interface. Each network interface has one name and at most nine pseudo-interface names. Each network interface and pseude-interface has its own IP address, network mask and broadcast mask.

public domain

Intellectual property available to people without paying a fee.

quality of service (QoS)

The OSI equivalent of TOS.

RARP

See **Reverse Address Resolution Protocol**.

RCD

See **RMT/RCD**.

RCP

See **remote copy program**.

reachable node

The node to which the local node has a usable communications path.

read access

The access right that grants the ability to view data.

reassembly

The process of piecing together datagram fragments to reproduce the original datagram based upon the fragmentation data in the IP header of the datagram.

reassembly time

A settable routing parameter which specifies the length of time allowed for the reassembly of a message received in fragments. If the reassembly time expires before all fragments are received, the fragments are discarded.

Record Management Services (RMS)

The OpenVMS data management subsystem that defines the rules that govern the internal organization of and the methods of accessing file data, including how files are named and cataloged in directories.

reliability

The ability of a protocol to recover data that is damaged, lost, duplicated, or delivered out of order.

relative path name

A path name that does not start at the root; default directory is merged with the relative path name to form the absolute path name.

relay queue

remote boot (BOOTP)

The software that supports the downloading of system images and other types of files to requesting clients.

remote copy program (RCP)

The program based on the Berkeley UNIX (see BSD) *rcmd* protocol that permits files to be copied from one computer to another by an extension to the syntax of the UNIX *cp* (copy) command. (RCP) does not provide the word-length adaptability and flexibility that the FTP protocol does.

remote line printing (LPR/LPD)

The remote printing services for UNIX and OpenVMS client hosts.

remote node

A node in the network other than the local node.

remote file system

A file system that resides on a network host other than the local node.

remote procedure call (RPC)

A programming interface for implementing the client/server model of distributed computing. In general, a request is sent to a remote system to execute a designated procedure, using arguments supplied, and the result returned to the caller. *See also* **Sun RPC**.

remote shell

A program that sends a command, shell, script, or command procedure to a remote host for execution.

remote task

A task either executing at a remote host or originating there.

repeater

A bidirectional device that amplifies or synchronizes signals into standard voltages, currents, and timing; propagates electrical signals from one Ethernet to another without making routing decisions or providing packet filtering; Physical layer intermediate system. *See also* **bridge** and **router**.

Request for Comments (RFC)

A series of documents, begun in 1969, that describes the Internet suite of protocols and related experiments. Very few RFCs describe Internet standards, but all Internet standards are written as RFCs.

resolver

A mechanism or process to correlate a network host name into an appropriate network address in support of network applications—a network name resolver. *See* **BIND resolver**.

reserved port

An assigned port that provides services to unknown callers by providing a service contact point; reserved port numbers range from 1 to 255.

resynchronization

A process that enables the recovery of user information lost or corrupted during transfer across an association. Sets the association back to the state it was in at a specified point in the transfer.

retransmission

A method of error recovery in which stations receiving messages acknowledge the receipt of correct messages and, on receipt of incorrect messages, either do not acknowledge or acknowledge in the negative. The lack of acknowledgment or receipt of a negative acknowledgment indicates to the sending station that it should transmit the failed message again.

Reverse Address Resolution Protocol (RARP)

The TCP/IP protocol that provides the reverse function of ARP. This protocol maps a physical (hardware) address to an IP address. Often used by diskless nodes when they first initialize to find their Internet address.

reverse domain

An Internet domain that BIND servers use to map IP addresses to domain names.

RFC

See **Request for Comments**.

RFC 822

The TCP/IP standard format for electronic mail message headers; often referred to as "822 messages". The name comes from RFC 822 that contains the specification; previously known as 733 format.

RIB (routing information base)

routing database

RIP

See **Routing Information Protocol**.

rlogin

Remote login: The Berkeley 4.3 BSD service that allows users of one machine to connect to other systems across the Internet and interact as if their terminals are connected the machines directly.

RMS

See **Record Management Services**.

RMT/RCD

Remote command that allows remote users to access magnetic tapes and CD drives.

root

The top level directory in a UNIX-style file system; also used to indicate a user (the superuser) who has special privileges. *See* **superuser**.

root mode

The file protection placed on a container file when it is created.

root name

The element of a path name that identifies the target file system.

root server

An Internet name server that knows about all of the top-level domains on the Internet network; the master servers for the Internet root zone.

round-trip delay

The total time during communications that implement a protocol with positive acknowledgments, for a message to be transmitted, arrive at its destination, and its corresponding acknowledgment to be sent and subsequently received by the sender of the original message.

The time it takes for a host to send an NTP packet to another host and get an NTP packet back from that host in reply.

round-trip time (RTT)

A variable computed during TCP sessions that indicates the total time required to send a TCP segment to a remote host and receive a reply.

route

The path over the network that information takes to get from one source to its destination.

route aggregation

route through

Data packets not destined for the local node.

routes database

The DIGITAL TCP/IP Services for OpenVMS database that specifies Internet gateways.

ROUTED

See **Route Daemon**.

Route Daemon (`routed`)

A program that runs under 4.2BSD/4.3BSD UNIX systems (and derived operating systems) to propagate routes among machines on a local area network using the Routing Information Protocol; pronounced "route-d."

One of a class of interior gateway protocols, described in more detail in the RIP section of gated.proto(4).

router

A node that can send and receive data and also forward data to other nodes.

router advertisement

A Router Discovery Protocol message sent out by Router Discovery Servers to announce their existence to hosts. The router advertisement contains a list of all router addresses on a given interface and their preference for use as a default router.

Router Discovery Protocol

An IETF standard protocol used to inform hosts of the existence of routers. It is used in place of or in addition to statically configuring default routes in hosts. The protocol has a server portion which runs on routers and a client portion which runs on hosts.

router id

A 32-bit number assigned to each router running the OSPF protocol. This number uniquely identifies the router within the autonomous system.

router_id

An IP address used as unique identifier assigned to represent a specific router. This is usually the address of an attached interface.

router solicitation

A Router Discovery Protocol message sent out by a host to request router advertisement responses from a router.

routing

A Network layer function, implemented in intermediate systems, that determines the path along which data travels to its destination and the movement of that data. *See also* **decision**.

routing database

The database that contains routing information, including destination host names, IP addresses for the hosts, gateway host names, and IP addresses for the gateways. There are two route databases: the static route database that is maintained on disk and the volatile database in memory.

The repository of all of gated's retained routing information, used to make decisions and as a source for routing information that is propagated.

routing domain

A set of hosts and routers within a single administrative domain that operates according to the same routing procedures.

Routing Information Protocol (RIP)

The protocol that enables gateways to broadcast their current routing database to hosts and networks that are connected directly to them. DIGITAL TCP/IP Services for OpenVMS software implements the RIP through its dynamic routing Zserver.

One of a class of interior gateway protocols, described in more detail in the RIP section of gated.proto(4).

Routing Protocol

A protocol sent between routers by which routers exchange information on how to route to various parts of the network. The TCP/IP family of protocols has many of this type of protocol, such as RIP, EGP, BGP, OSPF, and dual IS-IS.

routing socket

A data structure used by processes to communicate routing information to the kernel. A process can add and delete routes, dump the routing table and read messages from the kernel. The only type of socket supported in the AF_ROUTE domain is a raw socket.

routing table

The repository of all of gated's retained routing information, used to make decisions and as a source for routing information that is propagated.

RPC

See **remote procedure call** and **Sun RPC**.

rshell

Remote shell; a remote utility that gives the user with a shell session on a remote host.

RTL

See **Run-Time Library**.

RTT

See **round-trip time**.

Run-Time Library (RTL)

A collection of OpenVMS procedures available to native mode images at run time; provide support routines for high-level language compilers.

SCALE

A TCP window scaling option; allows window information to be interpreted as being scaled by 1 to 16 powers of 2, thus increasing the size of the effective window.

secondary server

A master BIND server that receives authoritative database information from a primary server. *See also* **BIND server**, **cache server**, **forwarder server**, and **primary server**.

segment

A unit of data exchanged by the TCP modules.

segment length

The amount of sequence number space occupied by a segment, including controls that occupy sequence space.

sequence number

A 32-bit field in the TCP header that contains the sequence number of a sequenced control flag, the first byte of data, or empty segments (The sequence number of the next data octet to be sent).

serial device

A device that uses serial transmission; that is, transmits data one bit at a time on a single channel as opposed to parallel transmission which transmits one or more bits at a time on one or more channels. Typically, terminals and printers are serial devices.

Serial Line Internet Protocol (SLIP)

A protocol designed to allow a host to connect to another host over serial lines, such as telephone circuits or RS-232 cables.

server

A process that offers a service to another process over the network and accepts requests from other processes, known as clients.

service

(1) A task that an application can carry out. (2) The interface provided by a service element or layer for accessing one or more function.

service interface

The boundary at which a layer provides a service to the adjacent higher layer in the network architecture; may vary between implementations.

service parameter

The means by which a service user and a service provider exchange information.

service provider

In network architecture, the service element or layer that provides a set of services to the layer immediately above.

service specification

An international standard that describes the functions and service parameters of every service of a service provider.

service user

An application program, service element, or Network layer that uses the services of a service provider.

services database

The DIGITAL TCP/IP Services for OpenVMS database created by default that contains one entry for each service configured.

Simple Mail Transfer Protocol (SMTP)

An Internet standard protocol for transferring electronic mail messages from one machine to another; specifies how two mail systems interact and the format of control messages they exchange to transfer mail.

Simple Network Management Protocol (SNMP)

The network management protocol of choice for TCP/IP-based internets; allows remote monitoring and management of network devices (particularly routers and servers) from across an Internet.

simplex

An interface may be marked as simplex either by the kernel, or by interface configuration. A simplex interface is an interface on a broadcast media that is not capable of receiving packets it broadcasts.

The `gated` daemon takes advantage of interfaces that are capable of receiving their own broadcast packets to monitor whether an interface appears to be functioning properly.

skew

A measure, in hertz, of the difference between the actual frequency of a clock and what its frequency should be to keep perfect time. See also "drift".

slave server

A name server that has no access to the Internet and relies on forwarder servers to resolve queries that it cannot resolve locally. As slave servers receive information from forwarder servers, they store that information in their cache. *See also* **cache server**, **forwarder server**, **primary server**, and **secondary server**.

slew

To adjust gradually the time of a clock until it tells the correct time. Compare with step.

SLIP

See Serial Line Internet Protocol

SMI

See **Structure of Management Information**.

SMTP

See **Simple Mail Transfer Protocol**.

SNMP

See **Simple Network Management Protocol**.

socket

The end point of communication to which an IP address and port may be bound. When writing an application, it is a data structure that is part of the Internet pseudodevice created every time an OpenVMS process assigns a communication channel. The other part of the Internet pseudodevice is the device socket.

socket pair

The client IP address and port number and the server IP address and port number that uniquely identifies a TCP connection.

socket API

An application programming interface for implementing TCP/IP protocols. Sometimes called Berkely sockets indicating where the API was developed.

source

The IP header field that contains the IP address of the datagram's point of origin.

source port

A 2-octet value in the TCP or UDP header field that identifies the upper-level application or protocol associated with the data in the segment.

spanning tree

A logical arrangement created by bridges in an extended LAN in which all LANs are connected and there are no loops.

split horizon

When a router (or group of routers work together) accepts routing information from multiple external networks, but does not pass on information learned from one external network to others. This is an attempt to prevent false routes to a network from being propagated because of gossip or counting to infinity.

splitting

The process of mapping one transport connection to several network connections.

stateless

A characteristic of a server designed to simplify crash recovery after a server crashes and reboots. The server does not keep track of the status of ongoing client interactions. Servers that do not keep track of client status are called stateless servers.

static routing

A routing method by which a system manager manually adds routes to the kernel's routing table. This method is generally used on small networks. On Open VMS systems, you use the SET ROUTE command to add static routes and on UNIX systems, you use the `route` command.

step

To change the time of a clock to the correct time with no intermediate adjustments. Compare with "slew".

stratum

The distance a host running the NTP time daemon is from an external source of Coordinated Universal Time (UTC). A stratum 1 server has direct access to an external source of UTC, such as a radio clock synchronized to a standard time signal broadcast. In general, a stratum n server is n-1 network hops away from a stratum 1 server. For example, a stratum 4 server is 3 hops away from a stratum 1 server. Also, a stratum n server is at a higher stratum than a stratum n-1 server. For example, a stratum 3 server is at a higher stratum than a stratum 2 server, and at a lower stratum than a stratum 4 server. See also "time daemon".

stream-oriented

The type of transport service that allows its client to send data in a continuous stream; guarantees that all data will be delivered to the other end in the same order as sent and without duplicates. Also known as a reliable transport service.

Structure of Management Information (SMI)

The rules used to define the objects that can be accessed by means of a network management protocol. *See also* **Management Information Base**.

subnet

An organization of hosts within a network into logical groups. A network can be comprised of several subnets. The portion of a network, which might be a physically independent network, that shares a network address with other portions of the network and is distinguished by a subnet number. A subnet is to a network what a network is to an internet.

subnet address

A part of the Internet addressing scheme. If a site uses a single IP address for multiple physical networks, there is one subnet address for each physical network. Each such address is composed of the network part of the full address and part of the local part (host).

subnet field

A bit field in an IP address that denotes the subnet number. The bits making up this field are not necessarily contiguous in the address.

subnet mask

A method of representing the portion of the IP network address that is devoted to subnet address. Each bit that is turned on (binary one) in the mask is interpreted as part of the network and subnet address. Synonymous with network mask. *See* **address mask**.

Sun Remote Procedure Call (RPC)

An easy and popular paradigm for implementing the client/server model of distributed computing. In general, the local system (client) sends a request to a remote system (server) to execute a designated procedure, using supplied arguments, and the remote system returns the result to the local system.

superuser

A UNIX user who has been granted special privileges; has an effective UID of 0.

symbiont

(1) A process that transfers record-oriented data to and from a mass storage device; for example, from disks to printers. (2) Synonym for daemon.

symbolic link

In the UNIX file system, a symbolic link is a file that contains a pointer to another file or directory. The link (also called a soft link) may be created across a different UNIX file system. Any changes to the file can be seen when you access the file through the file name or through the symbolic link. If you delete the file, the symbolic link will point to a non-existent file.

synchronous transmission

Data transmission in which characters are transmitted at a fixed rate. The transmitter and receiver are synchronized, gaining greater efficiency than in asynchronous transmission. Synchronous transmissions send a predetermined group of "sync" characters ahead of a long stream of data. The sync characters enable the communicating devices to synchronize with each other in accordance with a time clock at each end. *Contrast with* **asynchronous transmission**.

syntax

The rules for formatting/interpreting data.

TAC

See **terminal access controller**.

target system

The intended destination of messages.

TCP

See **Transmission Control Protocol**.

TCP/IP

An Internet suite of protocols. *See also* **Transmission Control Protocol** and **Internet Protocol**.

TELNET

An Internet protocol for remote terminal connection. TELNET allows a user at one site to interact with remote timesharing systems at another site as if the user's terminal were directly connected to the remote host.

terminal access controller (TAC)

A program and hardware that connects terminals to the Internet, usually using dialup modem connections.

terminal emulator

A program that allows a computer to emulate a terminal; a workstation thus appears as a terminal to the host.

terminal server

A device that handles terminal operations for host nodes on a LAN; can be used to connect terminal users to nodes on the same LAN and to users on nodes located off the LAN. Off-loads the terminal connection and I/O responsibilities from host nodes, and reduces the number of direct terminal connections to each host, thus saving substantial power, packaging, and cabling expense.

terminating packet

A packet whose destination is the local node.

TFTP

See **Trivial File Transport Protocol**.

thread

(1) A request from an NFS client to the NFS server. (2) A single unit of execution within a program.

throughput

A measure of how much data is sent, or can be sent, between two points in a specified unit of time; often used in either of two contexts:

- Rated throughput, which refers to the bandwidth or capacity of a component
- Real throughput, which refers to actual measured throughput.

time

A time value, usually a time interval. It may be specified in any one of the following forms:

```
number
  A non-negative decimal number of seconds.  For example, 27, 60, or
  3600.

number:number
  A non-negative decimal number of minutes followed by a seconds value
  in the range of zero to 59, inclusive.  For example, 0:27, 1:00, or
  60:00.

number:number:number
  A non-negative decimal number of hours followed by a minutes value in
  the range of zero to 59, inclusive, followed by a seconds value in
  the range of zero to 59, inclusive.  For example, 0:00:27, 0:01:00,
  or 1:00:00.
```

time to live (TTL)

A field in the IP header that indicates how long this packet should be allowed to be forwarded to other routers before being discarded.

The Time To Live (TTL) of an IP packet. Valid values are from one (1) through 255, inclusive.

time daemon

The program running on a host that synchronizes the host's hardware clock to Coordinated Universal Time in accordance with the protocols known as the Network Time Protocol.

timeo

A timeout option for the NFS `mount` command.

TN3270

TELNET options that allows TELNET users to connect to hosts that support 3270 model terminals.

token ring

A type of LAN that has stations wired in a ring, where each station constantly passes a special message (a "token") on to the next; technically referred to as IEEE 802.5.

topology

The architecture of a network. A network topology shows the computers and the links between them within a network.

TOS (type of service)

The type of service is for internet service quality selection. The type of service is specified along the abstract parameters precedence, delay, throughput, reliability, and cost. These abstract parameters are to be mapped into the actual service parameters of the particular networks the datagram traverses. The vast majority of IP traffic today uses the default type of service.

traffic

The measurement of data flow, volume, and velocity over a communications link.

transceiver

Transmitter-receiver; a physical device required in baseband networks that takes the digital signal from a computer or terminal and imposes it on the baseband medium; connects a host interface to a LAN, such as Ethernet.

transient information

Network management information carried in an operation; is meaningful only while the operation is being performed.

transit network

A network that passes traffic between networks in addition to carrying traffic for its own hosts; must have multiple connections to the internet.

Transmission Control Protocol (TCP)

A Transport layer protocol that provides the reliable, full-duplex, stream service on which many application protocols depend. TCP allows a process on one host to send a stream of data to a process on another. It is connection-oriented in the sense that before transmitting data, participants must establish a connection.

Transmission Control Protocol/Internet Protocol (TCP/IP)

The acronym for the suite of application and transport protocols that run over IP, for example, FTP, TELNET, and UCP as well as TCP and IP themselves.

Transport layer

The layer in the TCP/IP architecture model where network traffic is passed between an application on one host and an application on another host.

Trivial File Transport Protocol (TFTP)

The Internet protocol for file transfer with minimal capability and minimal overhead. The simple design of the facility is intended for use in application environments that do not require complex interactions among clients and servers. TFTP is a simple service running on top of UDP, using timeout and retransmission to ensure that data arrives. The sending side transmits a 512-byte, fixed-size file, and awaits an acknowledgment for each block before sending the next. The receiver acknowledges each block. *See also* **File Transfer Protocol**.

TTL

See **Time to Live**.

tunneling

The encapsulation of protocol A within protocol B such that A treats B as though it were a Network Interface layer. Used to get data between administrative domains that use a protocol not supported by the internet connecting those domains.

UAF

See **user authorization file**.

UCP

See **Management Control Program**.

Management Control Program

The DIGITAL TCP/IP Services for OpenVMS network management control software; includes a command-line interface.

UDP

See **User Datagram Protocol**.

UID

See **user identification**.

UNIX-style file system

An OpenVMS organization of files based on the UNIX operating system.

UNIX-to-UNIX Copy Program (UUCP)

A program that allows one UNIX system to copy files to or from another UNIX system.

upline dumping

A TFTP server function allowing a TFTP client to transfer data or a program image to the TFTP server's public directories. The opposite function of downline loading.

user authorization file (UAF)

An OpenVMS file that contains account names and their associated attributes.

User Datagram Protocol (UDP)

An Internet transport protocol. A connectionless, unreliable Transport layer protocol for the exchange of requests and replies between networked hosts. UDP, like TCP, uses IP for message delivery from one host to another; however, unlike TCP, UDP provides for exchange of datagrams without acknowledgments or guaranteed delivery of data. Each UDP message contains the data sent by a user process, a destination port number, and a source port number.

user identification (UID)

A unique number that identifies a user of a UNIX system. The number along with an associated group identification number (GID) determines file access privileges. Accounting statistics and other collected information also tracts by UID.

UUCP

See **UNIX-to-UNIX Copy Program**.

virtual circuit

The network service that allows two processes to communicate as if they were directly connected, regardless of the structure of the underlying subnet.

WAN

See **wide area network**.

well-known port

A port number assigned for use by a specific network application for connections made with either UDP or TCP. Every implementation of TCP/IP which provide well-known services provides them with the well-known port numbers between 1 and 1023. The Internet Assigned Numbers Authority (IANA) manages the well-known port numbers.

wide area network (WAN)

A network, usually constructed with serial lines, which covers large geographic areas.

wildcarding

A method for generalizing parts of a OpenVMS file designation to encompass a set of files by substituting a symbol to represent one or more characters. OpenVMS wildcarding symbols are % (for one character) and * (for a character string of any length, including zero).

window

A 2-octet field in a TCP header indicating the number of data octets (relative to the acknowledgment number in the header) that the sender is currently willing to accept.

write access

An Access right that grants users the ability to change data.

zone

A subdivision of the Internet hierarchy that starts at a domain and extends down to leaf domains (individual host names) or to domains where other zones begin; usually represents an administrative boundary. *Contrast with* **domain**.

zone file

A master name server file that describes the domain names for which the server has authority.

G.2 Acronyms

Table 1 shows DIGITAL TCP/IP Services for OpenVMS acronyms and other acronyms related to open networking.

Table 1 Acronyms

Acronym	Meaning
ACK	acknowledgment
ACL	access control list
ACP	ancillary control process
ANSI	American National Standards Institute
API	application programming interface
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
ATM	asynchronous transfer mode
BBS	Bulletin Board System
BGP	Border Gateway Protocol
BIND	Berkeley Internet Name Domain
BOOTP	Bootstrap Protocol
bps	bits per second
BSD	Berkeley Software Distribution
CFS	container file system
CFSRTL	container file system run-time library
CSLIP	Compressed Serial Line Internet Protocol
DCE	Distributed Computing Environment
DCL	Digital Command Language
DEK	data encryption key
DES	data encryption standard
DNS	Domain Name Service
eSNMP	extensible Simple Network Management Protocol
EGP	External Gateway Protocol
FDDI	Fiber Distributed Data Interface
EOF	end of file
EOL	end of line
FQDN	fully qualified domain name
FTP	File Transfer Protocol
GID	group identification (UNIX)
IAB	Internet Architecture Board
ICMP	Internet Control Message Protocol
IGP	Internal Gateway Protocol

Table 1 (Cont.) Acronyms

Acronym	Meaning
InterNIC	Internet Network Information Center
IP	Internet Protocol
ISDN	Integrated Services Digital Networks
IVP	installation verification procedure
Kbps	kilobits per second
LAN	local area network
LPD	line printer daemon
LPR	remote line printing
MBUF	memory buffer
MFD	master file directory
MIB	Management Information Base
MIBII	Management Information Base II
MTU	maximum transmission unit
MX	mail exchanger
NAK	negative acknowledgment
NFS	Network File System
NIS	Network Information Service
NOC	Network Operations Center
NTP	Network Time Protocol
PDU	protocol data unit
PING	packet internet groper
POP	Post Office Protocol
PPP	Point-to-Point Protocol
PSDN	Packet Switching Data Network
PWIP	PATHWORKS Internet Protocol
RARP	Reverse Address Resolution Protocol
RCP	remote copy
REXEC	remote execute
RFC	Request for Comments
RLOGIN	remote login
RIP	Routing Information Protocol
RMS	Record Management Services
RPC	remote procedure call
RSH	remote shell
RTL	run-time library
RTT	round-trip time
SLIP	Serial Line Internet Protocol
SMI	structure of management information

Table 1 (Cont.) Acronyms

Acronym	Meaning
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
TAC	terminal access controller
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transport Protocol
TP	Time Protocol
TTL	time to live
UAF	user authorization file
UCP	Management Control Program
UDP	User Datagram Protocol
UID	user identification (UNIX)
UTC	Coordinated Universal Time
UUCP	UNIX-to-UNIX Copy Program
WAN	wide area network
WKS	Well Known Server
XDR	external data representation

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Z