behind the scenes of NeXT networking

by Marc Majka

If you're about to become a NeXT administrator or are just curious about what goes on behind the scenes of the network you use, read on. You'll find it useful to get a big picture of what the computers in your network are doing and why they're doing it. In this article, we examine the fundamentals of NeXT networking, and because NeXT computers support UNIX® networking standards, we also explore how you can use your NeXT computer to take advantage of UNIX networking.

Before you worry about wires and software, it's important to consider why you want to use a network. Computer networks support resource sharing and communication. Resource sharing permits computers to offer services-such as shared files and shared printers-to other computers on a network. Communication permits you to exchange information with other computer users and to interact with distant computers. The wires, boxes, and technical details are there to provide the how of computer networking.

There are four requirements for successful networking:

- Physical connectivity Computers must be interconnected by media that allow them to exchange data. Their network connections must have electronic interfaces that enable them to send and receive data using network hardware.

- Internetwork communications Computers that exchange information must agree on their data formats and methods of information exchange. They must be able to communicate with others that have different network hardware and system software, and they must be able to transfer data reliably across diverse networks.

- Resource-sharing software Computers that provide shared resources must support sharing with server software. Computers using shared resources must support sharing with client software. Servers use internetwork communications to provide data or to accept requests from clients.

- Administrative information management Networking hardware and software must provide administrators with the means to manage the information, communications, and shared resources of a network.

physical connectivity

Computers communicate using methods as diverse as dedicated network cable, fiber optic cable, telephone lines, and radio and satellite transmission. Many computer networks use more than one communication method. For example, a local Ethernet may be connected to a telephone network to provide communication with a local area network (LAN) on the other side of the country. Gateway devices that connect the networks convert data between the formats used on each network.

The NeXT Ethernet interface permits messages, called frames, to be sent to computers attached to the same Ethernet. Each interface has an identifying 48-bit binary number, called an address. Every message starts with the addresses of the sender and the destination. When an interface receives a message with its address as the destination, it passes the data in that message to higher-level system software. A broadcast address allows a computer to send a message to all computers on the same physical network.

At this low level of networking, network interfaces don't know whether messages reach their destinations. Messages may be lost or corrupted, or destination computers may be disconnected. Ethernet interfaces can respond to some error conditions, but it's up to higher-level software and applications to provide additional and improved services.

internetwork communications

Ethernet interfaces follow a set of rules called a communication protocol. Each level of service on a network has its own protocol that defines the interactions that level has with equivalent software on another computer. Ethernet interfaces provide the lowest-level communication service. High-level software doesn't need to know how lower levels work. Network services can be provided by any type of computer system, as long as it speaks the same language as the computers receiving its communications.

the Internet Protocol

The next level of service on a NeXT computer is provided by software that follows a set of rules called the Internet Protocol (IP). IP provides a service for sending messages, called datagrams, across diverse physical networks. A gateway computer that has two or more network interfaces runs IP gateway software that receives datagrams from one interface and resends them on another. Gateways can also route datagrams according to their destination. Computers that direct IP datagrams between several attached networks are called IP routers. Routing software uses a table to determine which interface should be used for specific destination addresses.

Because IP datagrams travel across networks that use different schemes for addressing hardware, every computer using IP must have an IP address, assigned by an administrator. The collection of worldwide connected IP networks is called the Internet. Internet IP addresses are coordinated by the Network Information Center (NIC), which you can reach in the following ways:

- By mail

  Government Systems, Inc.
  Attn: Network Information Center
  14200 Park Meadow Drive, Suite 200
  Chantilly, VA 22021
  U.S.A.

- By e-mail

  hostmaster@nic.ddn.mil

- By telephone

  From North America: 1-800-365-3642
  From all other countries: +1-703-802-4535

IP addresses are 32-bit binary numbers structured to make routing easy. These addresses are divided into two parts, one indicating a network and the other a host. The network is usually specified by the bits at the beginning of the address, and the host is usually specified by the last few bits. This allows routers to decide which addresses can be reached over each of their network interfaces by keeping track of the network numbers used on each attached network. IP addresses are usually written as four numbers separated by dots, rather than as binary numbers. Each of the four numbers is the decimal value of a group of 8 bits. For example, the IP address 192.42.172.1 represents the binary number 11000000001010101010110000000001.

Some networks have many hosts; others have few. IP addressing provides for this by adjusting the number of bits used to specify the host. Network numbers are organized into

the address classes listed in table 1.

table 1: IP address classes

class	**range of addresses**		**network bits**	**host bits**
A	1.*x.x.x*	127.*x.x.x**	8	24
B	128.0.*x.x*	191.255.*x.x*	16	16
C	192.0.0.*x*	223.255.255.*x*	24	8
D[2]	224.*x.x.x*	239.*x.x.x*		
E[3]	240.*x.x.x*	247.*x.x.x*		

* The class A network 127.*x.x.x* is reserved for the internal loopback interface.
[2] Class D networks are used for multicast addressing. Multicast is not discussed in this article.
[3] Class E networks are reserved for future use.

Administrators can split an NIC-assigned network number into subnets. The netmask is a 32-bit number that tells the IP software which part of its IP address identifies the network and which part identifies hosts. Bits with a binary 1 value in the netmask identify the network. For example, hosts on the 129.18.1.xxx subnet would have the netmask 11111111111111111111111100000000 (the first 24 bits specify the network; the last 8 specify hosts). This could also be written as 255.255.255.0. Or you might see these numbers written in hexadecimal notation (each hexadecimal digit represents 4 binary digits). In hexadecimal, the netmask 255.255.255.255 would be written as ffffff00.

IP uses the netmask to decide whether destinations are on the same network. If they are, IP uses a network interface to send the datagram to the destination's physical address. If a destination is on a different network, IP uses the network interface to send to the physical address of a router that will forward to the correct destination. Although NeXT computers can maintain their own routing tables, many just use a default route that identifies a router for all other networks. This allows them to pass the buck-and the datagram-to another computer that has a complete routing table. See figure 1 for an example of routing through an IP network.

figure 1: IP routingÐA sends directly to B, but C sends to the router to forward to D.

A36_IPRouting.tiff ¬

IP software may need to broadcast datagrams. The broadcast IP address uses a host value of all bits that have a value of 1 (for example,129.18.1.255). The address of all bits having a value of 1 (255.255.255.255) is also used for broadcast. Routers don't forward broadcast datagrams. If a computer relies on broadcast to obtain a service, that service must be provided on its local network.

The level of service IP provides doesn't guarantee that datagrams will arrive at their destination. Gateways may fail or send datagrams in the wrong direction. Datagrams may be lost, delayed, or duplicated. They may arrive at the destination in an order different

from that in which they were sent. Although some error conditions can be detected and reported to the sender by return datagrams, it's the role of higher-level protocols to deliver improved levels of service.

interprocess communication protocols

IP datagrams provide a communication service between computers, but it's the processes running on these computers that need to communicate. For example, a mail-handling process on one computer needs to contact the mail-handling process on another computer to transfer electronic mail. This means that processes need addresses of their own. Two main Internet protocols support interprocess communication by giving addresses to processes. They are the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

UDP and TCP each maintain a list of addresses, called ports. You won't find a bunch of little hardware connectors on the back of your NeXT computer labeled TCP port 1, TCP port 2, and so on, but the idea of a TCP or UDP port is similar to that of a hardware port: It serves as an end point for communications. A server process connects to a port and listens for requests. A client process connects to a port and sends requests to the server's port.

Port numbers are like telephone numbers. Certain port numbers are well known-they're always assigned to certain services. For example, the e-mail server (sendmail) always

listens on TCP port 25. When sendmail needs to forward e-mail to another computer, it can contact the e-mail server on the other computer by connecting to TCP port 25 at the remote machine's IP address. Ports not assigned to well-known services are available for any process needing to make an interprocess connection and are assigned as required. The port numbers of well-known services are stored in the NetInfo /services directory.

IP provides a datagram service between computers. UDP provides a datagram service between processes. UDP software accepts data from a process, adds source and destination UDP port numbers to the beginning of the UDP datagram, and passes the resulting message to IP for transmission to the destination. IP software at the destination passes the message to UDP, which places the message in a queue for the destination port. Like IP datagrams, UDP datagrams can be lost, delayed, or delivered out of sequence.

TCP adds the reliability and services IP doesn't provide. Processes can use TCP to make a connection from port to port. Once a connection has been established, each process can send a stream of bytes to the other and be certain that the bytes have been delivered and that they have been delivered in order. If the connection is lost, processes are notified. Data flow is regulated so a fast data provider won't overwhelm the input capabilities of a slower recipient. Processes can carry on two-way communications without worrying about the capability of lower-level network layers, as shown in figure 2.

figure 2: TCP interprocess communication

A37_TCP_interprocess_commun.tiff ¬

Because TCP and UDP connections are process to process, a server can accept many connections at a UDP or TCP port and provide services for many clients simultaneously. The TCP protocol is so useful that it is often the interprocess communication channel used between processes running on a single host. Because the TCP and IP protocols are so common on UNIX networks, they are often referred to as TCP/IP networks, even though several other protocols, such as UDP and the Internet Control Message Protocol (ICMP), are also used on those networks.

resource-sharing software

Before networks were common, computers had to provide all services directly. During the last few years, a quiet revolution has occurred in computing. Computers attached to networks have made their services available throughout their network communities. To today's computer users, the services available on the network as a whole are more important than those provided by a particular computer.

At one time, resources such as hard disks, fast processors, and printers were expensive. In those days, administrators dedicated special computers on their networks to be servers. A server was a machine that had all the expensive resources attached to it. The rest of the

computers on the network were clients.

The situation is different today. Resources such as printers and disk drives are much less expensive. A computer can have many resources and can be a server for any of them. The resources computers most often share are files (hard disks), printers, and CPU processing power. Communication between computer users most often takes the form of remote login, file transfer, and electronic mail.

It's no longer meaningful to ask which computer is the server. Any machine can act as a server for some resource. Today, a server is not a computer. It's a process that communicates with client processes on other computers, responding to requests and providing information. This relationship is reflected in the client/server model supported by TCP and UDP. A server process listens for requests from client processes (from the same or another computer). Client processes make requests for services or information from the server processes. In figure3, you can see a client process on one computer using a TCP connection to communicate with a server process on another computer. The database server, for example, might be netinfod and the e-mail server sendmail.

figure 3: client/server communication

A38_TCP_client_server_commun.tiff ¬

In the rest of this section, we examine the processes that support resource sharing and communication in a TCP/IP network environment. All the network services discussed here are included with the NeXT system software, along with many others. They support resource sharing and communication among NeXT computers and non-NeXT systems.

file transfer: FTP

The Internet File Transfer Protocol (FTP) is an excellent example of a simple, but useful, application based on TCP. FTP enables file transfer between two computers. The command-line interface (called ftp) allows you to make a connection to a remote FTP server. The ftp process contacts the remote server (called ftpd) at TCP port 21. After the connection has been established and you have supplied a valid login name and password for the remote system, ftp provides simple commands to send and receive files. NeXT computers can use FTP to transfer files to and from any computer that supports the FTP protocol.

remote login: TELNET

The TELNET protocol is an application supported by the client process called telnet and the server process telnetd. TELNET uses TCP port 23 at each end of a connection to provide you with an interactive UNIX shell on a remote host. NeXT computers can use TELNET to communicate with any computer that supports the protocol.

shared printers: lpd

The line printer server (lpd) accepts connections at TCP port 515. Processes contact lpd to print files or to obtain queue information. If a process prints on a remote printer, the local lpd contacts lpd on the remote computer to forward the print request. Printer information on NeXT computers is stored in NetInfo. Any NeXT or non-NeXT computer that shares a printer using lpd can be listed in NetInfo. A NeXT printer can be shared with other UNIX computers by placing an entry for the NeXT printer in their /etc/printcap files.

electronic mail: SMTP

The Internet Simple Mail Transfer Protocol (SMTP) is supported by sendmail, which accepts connections at TCP port 25. sendmail either passes messages to a mail-delivery program for direct delivery to users or forwards messages to other SMTP servers. sendmail operates according to instructions in a configuration file that contains rules for handling messages based on their addresses. Administrators can use sendmail to set up precise routing and handling and provide gateways to diverse email systems. However, configuration files are written in a format that can be difficult for administrators to learn. On a NeXT computer, the sendmail configuration file is /etc/sendmail/sendmail.cf. In NeXTSTEP™ Release 3, the configuration file name is stored in NetInfo.

Most NeXT systems don't require customization of their sendmail configuration files. NeXT provides three predefined configuration files that work for most network configurations:

· /etc/sendmail/sendmail.subsidiary.cf causes sendmail to deliver mail directly to

known users.

- /etc/sendmail/sendmail.sharedsubsidiary.cf causes sendmail to pass any messages it receives to the SMTP processes running on a computer named mailhost.

- /etc/sendmail/sendmail.mailhost.cf is for a server (mailhost) that receives messages from clients.

A NeXT computer can be an e-mail client of a non-NeXT computer if it can contact the server by making a connection to TCP port 25 on a computer named mailhost. A NeXT computer can be a server for non-NeXT mail clients as long as the client can make a connection to the server's sendmail process.

sendmail passes e-mail to a delivery program (usually /bin/mail) that appends new messages to files in the directory /usr/spool/mail. For example, to deliver a message to the user susan, the message would be appended to the file /usr/spool/mail/susan. The mail spool directory is like a post office, with boxes for each user. /NextApps/Mail.app fetches messages from the spool directory. It periodically checks for new e-mail and transfers new messages to a user's Mailboxes directory.

For a networkwide e-mail service to work correctly, the central e-mail server must be identified as mailhost, and the mail spool directory must be accessible to all the e-mail

client computers. The name mailhost may be defined in a network administrative database. (We say more about network information systems later, in the section on administrative information management.) Sharing the mail spool directory is a job for the Network File System (NFS®).

shared files: NFS

NFS is a client/server protocol that allows computers to share files and directories. NFS uses Remote Procedure Call (RPC), a protocol built on both TCP and UDP. NFS servers maintain lists of the directories they are willing to share, or export, and may place restrictions on their exports. For example, access may be extended only to a specific list of clients. Clients must first obtain authorization to import (or mount) a directory from a server. On NeXT computers, this authorization comes from the process named rpc.mountd. Mount requests can be sent to a server by the UNIX mount command or from an automatic NFS mounting process, such as autonfsmount, which does most of the mounting work on a NeXT computer.

Once a client machine has mounted a directory from a server, processes can access the directory or files within the directory as if they were on the client's own disk. The NeXT operating system software itself acts as an NFS client, with some performance-enhancing assistance from processes called biod.

remote applications: NeXTSTEP WindowServer

Any NeXTSTEP application program can access the NeXTSTEP window system, keyboard, and mouse. These resources are managed by the WindowServer process. Application programs contact WindowServer to request drawing operations. WindowServer draws on the screen on behalf of its application clients and informs them of keyboard and mouse events. Clients usually contact the server by using Mach messaging to send messages to other processes on the same host. However, any NeXTSTEP application can make a connection to the NeXTSTEP window server on another computer. The remote window server will allow network connections only if it's been configured as a Public Window Server in the Experts panel of the Preferences application. Applications used in this way must be started from a UNIX shell, and their full pathnames must be used. The option -NXHost remote-host-name causes the application to contact the window server on the remote host.

For example, to start Digital Librarian™ and direct it to use the window resources of a machine named astra, you'd use the following UNIX shell command:

```
/NextApps/Librarian.app/Librarian -NXHost astra
```

If the connection fails, the application prints an error message and then quits.

time synchronization: NTP

Some network services require computers to agree on the time of day. For example, NFS

uses the modification times associated with files to decide whether clients have new or old versions of their data. Because a clock is a resource, network protocols can be used to share time information on a network. NeXT computers can be configured to use Network Time Protocol (NTP) to synchronize their clocks. ntpd processes use TCP port 123 to communicate.

If a network has an NTP master server, clients adjust their clocks to coincide with the master's clock. ntpd clients request the time from the master and adjust the client computers' internal times to drift gradually toward the master's time. The drift is slow, so it's a good idea to manually set the clocks on your computers to be close before you set up NTP. It's possible to set up clone time servers that will provide time information to clients if the master is unavailable.

If the master is down or there is no master server at all, NTP clones synchronize their time as a group by finding an average of their time values and all drifting toward the average. NTP uses Coordinated Universal Time internally, so synchronization works across time zones.

If you haven't set up network time service, your computers keep time by their own clocks without synchronization.

administrative information management

TCP/IP networking and client/server applications provide communication and resource-sharing capabilities. But networks don't manage themselves. Administrators need to maintain up-to-date information about their networks and to control network communications and shared resources.

Before computer networks, an administrator had little information to manage. The easiest way to administer these few items was to keep lists in plain ASCII files. For example, /etc/passwd contained a list of user names, passwords, and other account information associated with each user. The information in these files (which came to be called the administrative flat files) changed infrequently.

Networking changed administration dramatically. It became necessary to keep track of much more information about other computers and services available on the network. There were many new flat files to maintain. Worse still, any time a change was made on the network, the flat files on every networked computer needed to be updated. The problem was that every computer had its own picture of the network, and that picture was usually out of date.

The solution to this problem is to view administrative information as a shared resource. Information servers provide a centralized database, and processes that require administrative information become clients of the information servers. NeXT computers can access information from three information services: NetInfo, the Domain Name Service

(DNS), and the Network Information Service (NIS, formerly called Yellow Pages). NetInfo is always used as an information source. You need to set up the DNS and NIS on NeXT computers if you want to make use of their services.

An information service is like a book of administrative information. A computer that uses multiple information sources consults multiple books. NeXT computers always consult their sources in a fixed order: first NetInfo, then the DNS, and finally NIS. It's the job of the process lookupd to be the information clearinghouse for a NeXT computer. Any time other processes need information, lookupd consults its sources. It checks its sources only as far as it must to find information. If it finds what it's looking for in NetInfo, it doesn't search any farther. If the required information isn't available in NetInfo, lookupd consults the other services in turn (see figure 4).

figure 4: information lookup

A39_InfoLookUp_1.tiff ¬A39_InfoLookUp.tiff ¬

Three network information sources, like three reference books, rarely contain exactly the same information. An administrator can choose which information to maintain and distribute with each information service. Some services carry only certain types of information, and some services can't be accessed by all UNIX systems. You can use these information services to suit the needs of your particular network environment.

The most effective administrative strategies centralize information to avoid duplication. In practice, it's impossible to avoid some duplication, especially when you have many information sources. When the information stored in one system is changed, it's important to make the same change to other systems that carry the same information. It's a good idea to regard one system as primary and institute procedures for updating the other systems whenever you change the primary system. No existing software does this automatically. You'll need to create policies and procedures suitable for your environment. In some cases, you may want to keep duplicate information in two or more information systems and create programs to maintain data consistency.

NetInfo

Administrative information in NetInfo is organized into a hierarchy of domains. Each domain is a collection of information that is available to a single NeXT computer or a group of NeXT computers. The information that applies to a single NeXT computer is stored in its local domain. Other domains contain information available to groups of NeXT computers. For example, you might create a domain for all the NeXT computers in the marketing department of your organization, another domain for sales, another for research and development, and so on. The top-level domain would contain information available to all the NeXT computers in your company.

NetInfo domains are information sources. Because NeXT computers can belong to

several domains, it's important to know that domains are searched from bottom to top, starting with the local domain and ending with the top-level domain.

A NeXT computer can be configured to use only the local NetInfo domain. Although the local NetInfo domain is required, a NeXT computer need not belong to a NetInfo hierarchy to function correctly. NetInfo hierarchies exist to make it easier for you to organize and maintain administrative information for groups of NeXT computers.

the Domain Name Service

The Internet Domain Name Service (Some people call this system BIND, which is the 4.3BSD UNIX Berkeley Internet Name Domain server for the Domain Name Service. On NeXT computers, the server is called named. Some references refer to the server as the resolver.) is primarily used for looking up IP addresses. Because people prefer to use names rather than IP addresses for their computers, NetInfo, the DNS, and NIS all provide IP address lookup. DNS servers can be configured to contact DNS servers at other Internet sites to look up nonlocal IP addresses. For example, if you use ftp to retrieve files from the computer named next.com, DNS could determine the IP address of next.com. Your ftp process could then use TCP to make a connection to the TCP server at the correct IP address.

For computer names to identify computers, their names must be unique. To that end, the Internet has been split into name domains (separate from NetInfo domains), each with an

administrator responsible for ensuring that names within that domain are unique. For example, the ca domain contains computers in Canada. The uk domain contains computers in the United Kingdom. Some domains are not based on geography, for example, the com (commercial), mil (military), and edu (educational) domains. The fully qualified name of a computer has the domain name appended after a "." as in the example of the computer named next.com. Note that the names are not case sensitive, so the name next.com is equivalent to the name NeXT.COM.

A domain administrator can split a domain into subdomains and assign the authority for names within each subdomain to other administrators. For example, the edu domain contains many subdomains, such as mit.edu, which is used at the Massachusetts Institute of Technology. Subdomains can be split into further subdomains. As long as domain names are unique and all the computers within a domain have unique names, the fully qualified name of any computer will be unique. For example, a computer named astra at the computer science department at Pine Tree University might have the unique fully qualified name astra.cs.pine.edu.

DNS servers consult a database for their own domain to answer requests from clients. They may also have a list of names and addresses of DNS servers for other domains. DNS clients need to know only their domain name and the address of a computer that runs a DNS server. The DNS server process on a NeXT computer is called named. Clients must have their domain name and the IP address of up to three computers that run DNS

servers listed in the file /etc/resolv.conf. Clients will try connecting to each in turn, which makes it likely that if one server is down they can contact another server.

Because NeXT NetInfo domains are linked together by the serves property of computers named in the /machines directories of NetInfo domains, the names and IP addresses of NeXT computers must always be kept in NetInfo. If you want to have NeXT computer names and IP addresses carried by the DNS as well, you need to update both NetInfo and DNS databases whenever you change the name or IP address of a NeXT computer.

Network Information Service and flat files

NIS, an administrative information system supported on a large number of UNIX systems, also uses the idea of information domains (separate from NetInfo or DNS domains). A computer can belong to no more than one NIS domain. Information in an NIS domain is split up into a set of maps. A map is a list of keys and values that contains the kinds of information held in the UNIX administrative flat files.

NIS maps are distributed by NIS master and slave server processes (both named ypserv). Most administrators update NIS maps by editing the associated flat files on an NIS master and using a make utility program to create new maps from the flat files. Maps must also be distributed to the slave servers. This is usually done by make when the maps are updated.

The NIS client process (ypbind) obtains information from an NIS server. In most cases, flat files are ignored on a NeXT NIS client. However, user account and user group information lookups cause ypbind to consult the /etc/passwd and /etc/group files. The NIS passwd and group maps are consulted only if entries of the form "+" or "+name" are found. A "+" causes ypbind to include the entire passwd or group map. The form "+name" causes ypbind to include the user or group name.

For example, when you log into a NeXT computer using NIS, the loginwindow process initiates a lookup for your login name. lookupd checks its own information cache first, then NetInfo, and then NIS (continuing at each step if your name hasn't been found). The NIS lookup first checks /etc/passwd, which may contain an entry for your login name or may contain a "+" entry. A "+" entry causes NIS to include your login name from the NIS passwd map. If any information source contains your login name (and you have supplied the correct password), you can log in. loginwindow gets the information it needs to check that your account is valid but doesn't know where the information was found.

conclusion

This article is just an introductory look into the world of networking that's available on your NeXT computer. TCP/IP networking gives you the tools with which to build a rich network environment. NetInfo, the DNS, and NIS let you manage that environment easily. With a good idea of what your network is doing, how it works, and how to interpret the networking terminology you'll find in the documentation, you can confidently explore the networking

utilities of your NeXT computer. You can also use your knowledge to help you create and manage shared resources and communication in mixed UNIX networks that use the same TCP/IP-based resource sharing and communications systems.

references

The following references provide more information on some of the topics we've explored. Much more information about networking is waiting for you in NeXT's on-line documentation, too.

Comer, Douglas. *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

Comer, Douglas, and David L. Stevens. *Internetworking with TCP/IP, Volume II: Design, Implementation, and Internals*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

Nemeth, Evi, Garth Snyder, and Scott Seebass. *UNIX System Administration Handbook*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

NeXTSTEP Network and System Administration, Reading, MA: Addison-Wesley Publishing Co., 1992 (forthcoming).

Stern, Hal. *Managing NFS and NIS*. Sebastopol, CA: O'Reilly & Associates, 1991.

networking glossary

address A unique identifying number supplied by a computer vendor or administrator. Communication protocols rely on idenifying a computer by an address.

broadcast A message sent to all destinations. Many protocols support broadcast messages.

Domain Name Service (DNS) A network information system primarily used for lookup of host names and IP addresses.

Ethernet address A 48-bit binary number that identifies an Ethernet interface.
Ethernet frame A message sent using the Ethernet protocol. Frames include the source and destination Ethernet address. Frames are 64 to 1518 octets (8 bits) long.

File Transfer Protocol (FTP) An application based on TCP, FTP enables file transfer between two computers. The command-line interface is ftp.

gateway A device that converts messages between different network formats.

Internet The collection of worldwide connected IP networks.

Internet Control Message Protocol (ICMP) A protocol that uses IP to transmit IP control and error information.

Internet Protocol (IP) A communication protocol for sending messages, called datagrams, across diverse physical networks.

IP address A 32-bit number that identifies computers using the Internet Protocol to communicate.

IP datagram A message sent by a computer using the Internet Protocol. Datagrams include the source and destination IP address and may range from 20 to 65535 octets (8 bits).

IP router A device (sometimes a general-purpose computer) that connects two or more networks. Routers copy Internet Protocol datagrams between networks based on their destination IP addresses.

NetInfo NeXT's network information service.

netmask A 32-bit number that tells IP software which part of its IP address identifies the network and which part identifies hosts.

Network File System (NFS) A client/server protocol that allows computers to share files and directories.

Network Information Service (NIS) A network information service, formerly called Yellow Pages, that is supported by many UNIX systems.

protocol A set of rules and data formats used for communications.

Simple Mail Transfer Protocol (SMTP) A protocol and message format specification for e-mail exchange. SMTP is supported by many UNIX and non-UNIX computer systems.

subnet A part of an IP network with its own IP network number.

Transmission Control Protocol (TCP) A reliable byte-stream interprocess communication protocol. TCP is widely used to support client/server and peer-to-peer communications on UNIX networks.

TELNET A protocol used for remote terminal connections.

User Datagram Protocol (UDP) An interprocess datagram delivery service.