IN THIS CHAPTER

Networking and the Internet

9

The communications facilities linking computers are continually improving, allowing faster and more economical connections. The earliest computers were unconnected stand-alone machines. To transfer information from one system to another, you had to store it in some form (usually magnetic tape, paper tape, or punch cards—called IBM or Hollerith cards), carry it to a compatible system, and read it back in. A notable advance occurred when computers began to exchange data over serial lines, although the transfer rate was slow (hundreds of bits per second). People quickly invented new ways to take advantage of this computing power, such as e-mail, news retrieval, and bulletin board services. With the speed of today's networks, it is normal for a piece of e-mail to cross the country or even travel halfway around the world in a few seconds.

It would be difficult to find a computer facility that does not include a LAN to link the systems. GNU/Linux systems are typically attached to an *Ethernet* (page 1466) network. Wireless networks are becoming prevalent as well. Large computer facilities usually maintain several networks, often of different types, and almost certainly have connections to larger networks (company- or campuswide and beyond).

The Internet is a loosely administered network of networks (an *internetwork*) that links computers on diverse LANs around the globe. An internet (small *i*) is a generic network of networks that may share some parts in common with the public Internet. It is the Internet that makes it possible to send an e-mail message to a colleague thousands of miles away and receive a reply within minutes. A related term, *intranet*, refers to the networking infrastructure within a company or other

institution. Intranets are usually private; access to them from external networks may be limited and carefully controlled, typically using firewalls (page 358).

Over the past decade many network services have emerged and become standard. On GNU/Linux systems, as on UNIX computers, special processes called *daemons* (page 1463) support such services by exchanging specialized messages with other systems over the network. Several software systems have been created to allow computers to share their filesystems with one another, making it appear as though remote files are stored on local disks. Sharing remote filesystems allows users to share information without knowing where the files physically reside, without making unnecessary copies, and without learning a new set of utilities to manipulate them. Because the files appear to be stored locally, you can use standard utilities (such as cat, vi, lpr, mv, or their graphical counterparts) to work with them.

Developers have been creating new tools and extending existing ones to take advantage of higher network speeds and work within more crowded networks. The rlogin, rsh, and telnet utilities, designed long ago, have largely been supplanted by ssh (secure shell—page 374). The ssh utility allows a user to log in on or execute commands securely on a remote computer. Users rely on such utilities as scp and ftp to transfer files from one system to another across the network. Communication utilities, including e-mail utilities, and chat programs, such as talk, Internet Relay Chat (IRC), ICQ, and AOL Instant Messenger (AIM), have become so prevalent that many people with very little computer experience use them on a daily basis to keep in touch with friends and family.

An *intranet* is a network that connects computing resources at a school, company, or other organization but, unlike the Internet, typically restricts access to internal users. An intranet is very similar to a LAN but is based on Internet technology. An intranet can provide database, e-mail, and Web page access to a limited group of people, regardless of their geographic location.

The fact that an intranet is able to connect dissimilar machines is one of its strengths. Think of all the machines that are on the Internet: Macs, PCs running different versions of MS Windows, various machines running UNIX and GNU/Linux, and so on. Each of these machines can communicate via IP (page 360), a common protocol. So it is with an intranet: Different machines can all talk to one another.

Another key difference between the Internet and an intranet is that the Internet will transmit only one protocol suite: the IP protocol suite. An intranet can be set up to use a number of protocols, such as IP, IPX, Appletalk, DECnet, XNS, or various other protocols developed by vendors over the years. Although these protocols cannot be transmitted directly over the Internet, you can set up special gateway boxes at remote sites that tunnel or encapsulate these protocols into IP packets in order to use the Internet to pass them.

You can use an *extranet* (or *partner net*) to improve your security. A closely related term is virtual private network (VPN). These terms describe ways to connect

remote sites securely to a local site, typically by using the public Internet as a carrier and using encryption as a means of protecting data in transit.

As with the Internet, the communications potential of intranets is boundless. You can set up a private chat between people at remote locations, access a company database, see what is new at school, or read about the new university president. Companies that developed products for use on the Internet are investing more and more time and money developing intranet software applications as the intranet market explodes. Following are some words you may want to become familiar with before you read the rest of this chapter. Refer to the Appendix G on page 1453 for definitions.

ASP	bridge	extranet	firewall	gateway
hub	internet	Internet	intranet	ISP
packet	router	sneakernet	switch	VPN

Types of Networks and How They Work

Computers communicate over networks by using unique addresses assigned by system software. A computer message, called a *packet, frame,* or *datagram,* includes the address of the destination computer and the sender's return address. The three most common types of networks are *broadcast, point-to-point,* and *switched.* Once popular token-based networks (such as FDDI and Token Ring) are rarely seen anymore.

Speed is important to the proper functioning of the Internet. Newer specifications (cat 6 and cat 7) are being standardized for 1000BaseT (10 gigabits per second, called gigabit Ethernet, or GIG-E) and faster networking. Some of the networks that form the backbone of the Internet run at speeds up to almost 10 gig*abytes* per second (OC192) to accommodate the ever-increasing demand for network services. Table 9-1 lists some of the common specifications in use today.

Specification || table 9-1

DS0	64 kilobits per second
ISDN	Two DS0 lines plus signaling (16 kilobits per second) or 128 kilobits per second

Specification (Continued) || table 9-1

T-1	1.544 megabits per second (24 DS0 lines)
T-3	43.232 megabits per second (28 T-1s)
OC3	155 megabits per second (100 T-1s)
OC12	622 megabits per second (4 OC3s)
OC48	2.5 gigabits per seconds (4 OC12s)
OC192	9.6 gigabits per second (4 OC48s)

Broadcast

On a *broadcast network*, such as Ethernet, any of the many systems attached to the network cable can send a message at any time; each system examines the address in each message and responds only to messages addressed to it. A problem occurs on a broadcast network when multiple systems send data at the same time, resulting in a collision of the messages on the cable. When messages collide, they can become garbled. The sending system notices the garbled message and resends it after waiting a short but random amount of time. Waiting a random amount of time helps prevent those same systems from resending the data at the same moment and experiencing another collision. The extra traffic that results from collisions can put quite a load on the network; if the collision rate gets too high, the retransmissions result in more collisions, and the network becomes unusable.

Point-to-Point

A point-to-point link does not seem like much of a network at all because only two endpoints are involved. However, most connections to WANs are through point-to-point links, using wire cable, radio, or satellite links. The advantage of a point-to-point link is that because only two systems are involved, the traffic on the link is limited and well understood. A disadvantage is that each system can typically be equipped for a small number of such links, and it is impractical and costly to establish point-to-point links that connect each computer to all the rest.

Point-to-point links often use serial lines and modems but can use personal computer parallel ports for faster links between GNU/Linux systems. The use of a

modem with a point-to-point link allows an isolated system to connect inexpensively into a larger network.

The most common types of point-to-point links are the ones used to connect to the Internet. When you use DSL[1] (digital subscriber line), you are using a point-to-point link to connect to the Internet. Serial lines, such as T-1, T-3, ATM links, and ISDN, are all point to point. Although it might seem like a point-to-point link, a cable modem is based on broadcast technology and in that way is similar to Ethernet.

Switched

A *switch* is a device that establishes a virtual path between source and destination hosts such that each path appears to be a point-to-point link, much like a railroad roundhouse. The telephone network is a giant switched network. The switch brings up and tears down virtual paths as hosts need to communicate with each other. Each host thinks that it has a direct point-to-point path to the host it is talking to. Contrast this with a broadcast network, where each host also sees traffic bound for other hosts. The advantage of a switched network over a pure point-to-point network is that each host requires only one connection: the connection to the switch. Using pure point-to-point connections, each host must have a connection to every other host. Scalability is provided by further linking switches.

LAN: Local Area Network

Local area networks (LANs) are confined to a relatively small area—a single computer facility, building, or campus. Today most LANs run over copper or fiberoptic cable, but other, wireless technologies, such as infrared (similar to most television remote control devices) and radio wave, are becoming more popular.

If its destination address is not on the local network, a packet must be passed on to another network by a router (page 357). A router may be a general-purpose computer or a special-purpose device attached to multiple networks to act as a gateway among them.

Ethernet

A GNU/Linux system connected to a LAN usually connects to the network by using Ethernet. A typical Ethernet connection can support data transfer rates from 10 megabits per second to 1 gigabit per second, with speed enhancements planned.

1. The term DSL incorporates the xDSL suite of technologies, including ADSL, XDSL, SDSL, and HDSL.

Owing to computer load, competing network traffic, and network overhead, file transfer rates on an Ethernet are always slower than the maximum, theoretical transfer rate.

An Ethernet network transfers data by using copper or fiberoptic (glass) cable or wireless transmitters and receivers. Originally each computer was attached to a thick coaxial cable (called *thicknet*) at tap points spaced at six-foot intervals along the cable. The thick cable was awkward to deal with, so other solutions, including a thinner coaxial cable known as *thinnet*, or 10Base2,[2] were developed. Today most Ethernet connections are either wireless or made over unshielded twisted pair (referred to as UTP, Category 3 (cat 3), Category 5 (cat 5), Category 5e (cat 5e) 10BaseT, or 100BaseT) wire—similar to the type of wire used for telephone lines and serial data communications.

A *switched Ethernet* network is a special case of a broadcast network that works with a *network switch*, or just *switch*, which is a special class of hub that has intelligence. Instead of having a dumb repeater (hub) that broadcasts every packet it receives out of every port, a switch learns which devices are connected to which of its ports. A switch sorts packets so that it sends traffic to only the machine the traffic is intended for. A switch also has buffers for holding and queuing packets.

Some Ethernet switches have enough bandwidth to communicate simultaneously, in full-duplex mode, with all the devices that are connected to it. A nonswitched (hub-based) broadcast network can run in only half-duplex mode. Full-duplex Ethernet further improves things by eliminating collisions. Each host can transmit and receive simultaneously at 10/100/1000 megabits per second for an effective bandwidth between hosts of 20/200/2000 megabits per second, depending on the capacity of the switch.

Wireless

Wireless networks are becoming increasingly common. They are used in offices, homes, and public places, such as universities and airports. Wireless access points provide functionality similar to an Ethernet hub. They allow multiple users to interact, using a common radio frequency spectrum. A wireless, point-to-point connection allows you to wander about your home or office with your laptop, using an antenna to link to a LAN or to the Internet via an in-house base station. GNU/Linux has drivers for many of the common wireless boards. A wireless access point connects a wireless network to a wired network so that no special protocol is required for a wireless connection. Refer to the *Linux Wireless LAN HOWTO* and www.hpl.hp.com/personal/Jean_Tourrilhes/Linux.

2. Ethernet cables are classified as ***X*base*Y***, where ***X*** is the data rate in megabits per second, **base** means baseband (as opposed to radio frequency), and ***Y*** is the category of cabling.

WAN: Wide Area Network

A wide area network (WAN) covers a large geographic area. The technologies (such as Ethernet) used for LANs were designed to work over limited distances and for a certain number of host connections. A WAN may span long distances over dedicated data lines (leased from a telephone company) or radio or satellite links. WANs are often used to interconnect LANs. Major Internet service providers rely on WANs to connect to customers within a country and around the globe.

Some networks do not fit into either the LAN or the WAN designation: A MAN (metropolitan area network) is one that is contained in a smaller geographic area, such as a city. Like WANs, MANs are typically used to interconnect LANs.

Internetworking through Gateways and Routers

A LAN connects to a WAN through a *gateway,* a generic term for a computer or a special device with multiple network connections that passes data from one network to another. The purpose of the gateway is to convert the data traffic from the format used on the LAN to that used on the WAN. Data that crosses the country from one Ethernet to another over a WAN, for example, is repackaged from the Ethernet format to a different format that can be processed by the communications equipment that makes up the WAN backbone. When it reaches the end of its journey over the WAN, the data is converted by another gateway to the format appropriate for the receiving network. For the most part these details are of concern only to the network administrators; the end user does not need to know anything about how the data transfer is carried out.

A *router* is the most common form of a gateway. Routers play an important role in internetworking. Just as you might study a map to plan your route when you need to drive to an unfamiliar place, a computer needs to know how to deliver a message to a system attached to a distant network by passing through intermediary systems and networks along the way. You can imagine using a giant network road map to choose the route that your data should follow, but a static map of computer routes is usually a poor choice for a large data network. Computers and networks along the route you choose may be overloaded or down, without providing a detour for your message.

Routers communicate with one another dynamically, keeping one another informed about which routes are open for use. To extend the analogy, this would be like heading out on a car trip without consulting a map to find a route to your destination; instead you head for a nearby gas station and ask directions. Throughout the journey, you would continue to stop at one gas station after another, getting directions at each to find the next one. Although it would take a while to make the stops, each gas station would advise you of bad traffic, closed roads, alternative routes, and shortcuts.

The stops the data makes are much quicker than those you would make in your car, but each message leaves each router on a path chosen based on the most current information. Think of it as a GPS (global positioning system) setup that automatically gets updates at each intersection and tells you where to go next, based on traffic and highway conditions.

Figure 9-1 shows an example of how LANs might be set up at three sites interconnected by a WAN (the Internet). In network diagrams such as this, Ethernet LANs are drawn as straight lines, with devices attached at right angles; WANs are represented as clouds, indicating that the details have been left out; wireless connections are drawn as zigzag lines with breaks, indicating that the connection may be intermittent.

In Figure 9-1 a gateway or a router relays messages between each LAN and the Internet. Three of the routers in the Internet are shown (for example, the one closest to each site). Site A has a server, a workstation, a network computer, and a PC sharing a single Ethernet LAN. Site B has an Ethernet LAN that serves a printer and four GNU/Linux workstations. A firewall permits only certain traffic between the Internet router and the site's local router. Site C has three LANs linked by a single router, perhaps to reduce the traffic load that would result if they were combined or to keep workgroups or locations on separate networks. Site C includes a wireless access point that enables wireless communication with nearby computers.

Firewall

A firewall in a car separates the engine compartment from the passenger compartment, protecting the driver and passengers from engine fires, noise, and fumes. Computer firewalls separate computers from malicious and unwanted users.

A *firewall* prevents certain types of traffic from entering or leaving a network. A firewall might prevent traffic from your IP address from leaving the network and prevent anyone except users from select domains from using ftp to retrieve data from the network. The implementations of firewalls vary widely, from GNU/Linux machines with two *interfaces* (page 1473) running custom software to a *router* (page 1490) with simple access lists to esoteric, vendor-supplied firewall appliances. Most larger installations have at least one kind of firewall in place. A firewall is often accompanied by a proxy server/gateway (page 397) to provide an intermediate point between you and the host you are communicating with.

In addition to those found in multipurpose computers, firewalls are becoming increasingly common in consumer appliances. Firewalls are built into cable modems, wireless gateways, routers, and stand-alone devices.

When your need for privacy is critical, you can meet with a consulting firm that will discuss your security needs, devise a strategy, produce a written implementation

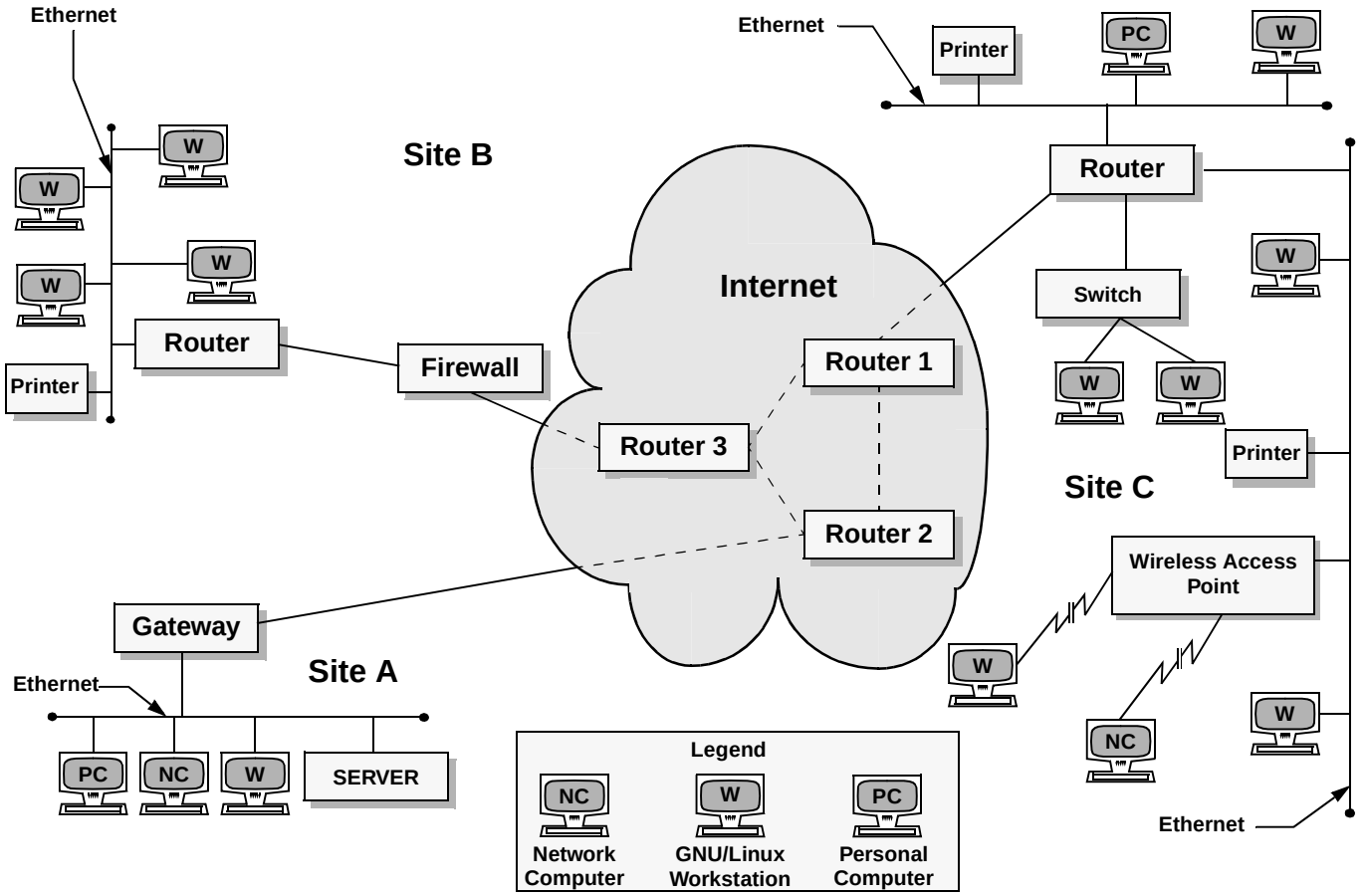

Figure 9-1 A slice of the Internet

policy, and design a firewall for you from scratch. Typically a single GNU/Linux machine can include a minimal firewall. A small group of GNU/Linux machines may have a cheap, slow GNU/Linux machine with two network interfaces and packet-filtering software functioning as a dedicated firewall. One of the interfaces connects to the Internet, modems, and other outside data sources, whereas the other connects, normally through a hub or switch, to the local network's machines. Refer to page 924 for information on setting up a firewall and to Appendix C for a discussion of security.

Network Protocols

To exchange information over a network, computers must communicate using a common language, or *protocol* (page 1486). The protocol determines the format of the message packets. The predominant network protocols used by GNU/Linux systems are TCP and IP, referred to as TCP/IP[3] (Transmission Control Protocol and Internet Protocol). Network services that need highly reliable connections, such as ssh and scp, tend to use TCP/IP. Another protocol used for some system services is UDP (User Datagram Protocol). Network services that do not require guaranteed delivery, such as RealAudio and RealVideo, operate satisfactorily with the simpler UDP.[4]

IP: Internet Protocol

Layering was introduced to facilitate protocol design: Layers distinguish functional differences between adjacent protocols. A grouping of layers can be standardized into a protocol model. IP is a protocol and has a corresponding model for what distinguishes protocol layers. The IP model differs from the ISO seven-layer protocol model (also called the OSI model) often illustrated in networking textbooks. IP uses a simplified five-layer model.

1. The first layer, called the *physical layer*, describes the physical medium (copper, fiber, wireless) and the data encoding used to transmit signals on that medium (pulses of light, electrical waves, or radio waves, for instance).
2. The second layer, called the *data link layer*, covers media access by network devices and describes how to put data into packets, transmit the data, and check it for errors. Ethernet is at this layer, as is 802.11 wireless.

3. All references to IP imply *IPv4* (page 1474).

4. Voice and video protocols are delay sensitive, not integrity sensitive. The human ear and eye accept and interpolate loss in an audio stream but cannot deal with variable delay. The guaranteed delivery that TCP provides introduces delay on a busy network when packets get retransmitted. This delay is not acceptable for video and audio transmissions, whereas less than 100 percent integrity is acceptable.

3. The third layer, called the *network layer*, frequently uses IP and addresses and routes packets.
4. The fourth layer, called the *transport layer*, is where TCP and UDP exist. This layer provides a means for applications to communicate with each other. Common functions of the transport layer include guaranteed delivery, delivery of packets in the order of transmission, flow control, error detection, and error correction. The transport layer is responsible for dividing data streams into packets. This layer also performs port addressing, which allows it to distinguish among different services using the same transport protocol. Port addressing keeps the data from multiple applications using the same protocol (for example TCP) separate.
5. Anything above the transport layer is the domain of the application and is part of the fifth layer. Unlike the ISO model, the Internet model does not distinguish among application, presentation, and session layers. All the upper-layer characteristics, such as character encoding, encryption, GUI, and so on, are part of the application. Applications choose the transport characteristics they require and choose the corresponding transport layer protocol to send and receive data.

TCP: Transmission Control Protocol

TCP is most frequently run on top of IP in a combination referred to as TCP/IP. TCP provides error recovery and guaranteed delivery in packet transmission order and works with multiple ports so that it can handle more than one application. TCP is a *connection-oriented protocol* (page 1461), also known as a streams-based protocol. Once established, a TCP connection looks like a stream of data, not individual IP packets. The connection is assumed to remain up and be uniquely addressable. Every piece of information you write to the connection always goes to the same destination and arrives in the order it was sent. Because TCP is connection oriented and establishes what you can think of as a *virtual circuit* between two machines, TCP is not suitable for one-to-many transmissions (see UDP, following). TCP has builtin mechanisms for dealing with congestion (or flow) control over busy networks and throttles back (slows the speed of data flow) when it has to retransmit dropped packets. TCP can also deal with acknowledgments, wide area links, high delay links, and other situations.

UDP: User Datagram Protocol

UDP runs at layer 4 of the IP stack, just as TCP does, but is much simpler. Like TCP, UDP works with multiple ports/multiple applications and has checksums for error detection but does not automatically retransmit packets that fail the checksum. UDP is a packet- (or datagram-) oriented protocol: Each packet must carry its own

address and port information. Each router along the way examines each packet to determine the destination one hop at a time. You can broadcast or multicast UDP packets to many destinations at the same time by using special addresses.

PPP: Point-to-Point Protocol

PPP provides serial line point-to-point connections that support IP. PPP compresses data to make the most of the limited bandwidth available on serial connections. PPP, which replaces SLIP (Serial Line IP), acts as a point-to-point layer 2/3 transport that many other types of protocols can ride on. PPP is used mostly for IP-based services and connections, such as TCP or UDP.[5] For more information, refer to "Internet Configuration Wizard" on page 1023.

Xremote and LBX

Two protocols that speed up work over serial lines are Xremote and LBX. Xremote compresses the X Window System protocol so that it is more efficient over slower serial lines. LBX (low-bandwidth X) is based on the Xremote technology and is a part of the X Window System release X11R6.

Host Address

Each computer interface is identified by a unique address, or host number, on its network. A system that is attached to more than one network has multiple interfaces, one for each network and each with a unique address.

Each packet of information that is broadcast over the network has a destination address. All hosts on the network must process each broadcast packet to see whether it is addressed to that host.[6] If the packet is addressed to a given host, that host continues to process it. If not, the host ignores it.

The network address of a machine is an IP address, which is represented as one number broken into four segments separated by periods (for example, 192.168.184.5). Domain names and IP addresses are assigned through a highly distributed system coordinated by ICANN (Internet Corporation for Assigned Names and Numbers—www.icann.org) via many registrars (see www.internic.net). ICANN is funded by the various domain name registries and registrars and IP address registries, which supply globally unique identifiers for hosts and services on the Internet. Although you may not deal with any of these agencies directly, your Internet service provider does.

---

5. SLIP was one of the first serial line implementations of IP and has slightly less overhead than PPP, but PPP supports multiple protocols (such as Appletalk and IPX), whereas SLIP supports only IP.

6. Contrast broadcast packets with unicast packets: Ethernet hardware on a computer filters out unicast packets that are not addressed to that machine; the operating system on that machine never sees them.

How a company uses IP addresses is determined by the system or network administrator. For example, the leftmost two sets of numbers in an IP address might represent a large network (campus- or companywide); the third set might specify a subnetwork (perhaps a department or single floor in a building); and the rightmost number, an individual computer. The operating system uses the address in a different, lower-level form, converting it to its binary equivalent, a series of 1s and 0s. See the following Optional section for more information. Refer to "private address space" on page 1486 in the Glossary for information about addresses you can use on your LAN without registering them.

Static versus Dynamic IP addresses

A static IP address is one that remains the same. A dynamic IP address is one that can change each time you connect to your ISP. A dynamic address remains the same during a single login session. Any server (mail, Web, and so on) must have a static address so that clients can find the machine that is the server. End user machines usually work well with dynamic addresses. During a given login session, they can function as a client (your Web browser, for example) because they have a constant IP address. When you log out and log in again, it does not matter that you have a new IP address, because your computer, acting as a client, establishes a new connection with a server. The advantage of dynamic addressing is that it allows inactive addresses to be reused, reducing the total number of IP addresses needed. Refer to "DHCP Client" on page 1028 for more information about dynamic IP addressing.

Optional

IP Classes

To facilitate routing on the Internet, IP addresses are divided into *classes*. Classes, labeled *class A* through *class E*, allow the Internet address space to be broken into blocks of small, medium, and large networks that are designed to be assigned based on the number of hosts within a network.

When you need to send a message to an address outside your network, your system looks up the address block/class in its routing table and sends the message to the next router on the way to the final destination. Every router along the way does a similar lookup to forward the message. At the destination, local routers direct the message to the specific address. Without classes and blocks, your host would have to know every network and subnetwork address on the Internet before it could send a message. This would be impractical because of the number of addresses on the Internet.

IP Classes || table 9-2

Class	Start Bits	Address Range	All Bits (including start bits) 0–7	8–15	16–23	24–31
Class A	0	001.000.000.000-126.000.000.000	0--netid--	==========hostid==========		
Class B	10	129.000.000.000-191.255.000.000	10-------netid------		======hostid=====	
Class C	110	192.000.000.000-223.255.255.000	110----------netid----------			=hostid==
Class D (Multicast)	1110	224.000.000.000-239.255.255.000	1110			
Class E (Reserved)	11110	240.000.000.000-255.255.255.000	11110			

Each of the four numbers in the IP address is in the range of 0–255 because each segment of the IP address is represented by 8 bits (an *octet*), each bit capable of taking on two values; the total number of values is $2^8 = 256$. When you start counting at 0, 1–256 becomes 0–255.[7] Each IP address is divided into a net address (*netid*) portion (which is part of the class) and a host address (*hostid*) portion. See Table 9-2.

The first set of addresses, defining class A networks, is for extremely large corporations, such as General Electric (3.0.0.0) and Hewlett-Packard (15.0.0.0), or for ISPs. One start bit (0) in the first position designates a class A network, 7 bits hold the network portion of the address (netid), and 24 bits hold the host portion of the address (hostid, Table 9-2). This means that GE can have 2^{24}, or approximately 16 million hosts on its network. Unused address space and *subnets* (page 1495) lower this number quite a bit. The 127.0.0.0 subnet is reserved (page 368), as are 128.0.0.0 and several others.

Two start bits (10) in the first two positions designate a class B network, 14 bits hold the network portion of the address (**netid**), and 16 bits hold the host portion of the address, for a potential total of 65,534 hosts.[8] A class C network uses 3 start bits (100), 21 netid bits (2 million networks), and 8 hostid bits (254 hosts). Today a new large customer will not receive a class A or B network but is likely to receive a class C or several (usually contiguous) class C networks, if merited.

Several other classes of networks exist. Class D networks are reserved for *multicast* (page 1480) networks. When you run **netstat –nr** on your GNU/Linux system, you can see whether your machine is a member of a multicast network. A 224.0.0.0 in the Destination column that netstat displays indicates a class D, multicast address

7. Internally, the IP address is represented as a set of four unsigned 8-bit fields, or a 32-bit unsigned number, depending on how programs are using it. The most common format in C is to represent it as a union of an unsigned 32-bit long integer, four unsigned chars, and two unsigned short integers.

(Table 9-2). A multicast is like a broadcast, but only hosts that subscribe to the multicast group receive the message. To use Web terminology, a broadcast is like a push. A host pushes a broadcast on the network, and every host on the network must check each packet to see whether it contains relevant data. A multicast is like a pull. A host will see a multicast only if it registers itself as subscribed to a multicast group or service and pulls the appropriate packets from the network.

Table 9-3 shows some of the computations for IP address 131.204.027.027. Each address is shown in decimal, hexadecimal, and binary. Binary is the easiest to work with for bitwise, (binary) computations. The first three lines show the IP address. The next three lines show the *subnet mask* (page 1495) in three bases. Next, the IP address and the subnet mask are ANDed together bitwise to yield the *subnet number* (page 1495), which is shown in three bases. The last three lines show the *broadcast address* (page 1458), which is computed by taking the subnet number and turning the hostid bits to 1s. The subnet number is the name/number of your local network. The subnet number and the subnet mask determine what range the IP address of your machine must be in. They are also used by routers to segment traffic; see *network segment* (page 1482). A broadcast on this network goes to all hosts in the range 131.204.27.1 through 131.204.27.254 but will be acted on only by hosts that have a use for it.

Subnets

Each host on a network must process each broadcast to determine whether the information in the broadcast packet is useful to that host. If a lot of hosts are on a network, each host must process many packets. To maintain efficiency, most networks, particularly shared media networks, such as Ethernet, need to be split into subnetworks, or *subnets*.[9] The more hosts on a network, the more dramatically network performance is impacted. Organizations use router and switch technology called VLANs (virtual local area network) to group similar hosts into broadcast domains (subnets) based on function. It's not uncommon to see a switch with different ports being part of different subnets.

---

8. A 16-bit (class B) address can address 2^{16} = 65,536 hosts, yet the potential number of hosts is two less than that because the first and last addresses on any network are reserved. In a similar manner an 8-bit (class C) address can address only 254 hosts ($2^8 - 2 = 254$). The 0 host address (for example, 194.16.100.0 for a class C or 131.204.0.0 for a class B) is reserved as a designator for the network itself. Several older operating systems use this as a broadcast address. The 255 host address (for example, 194.16.100.255 for a class C or 131.204.255.255 for a class B) is reserved as the IP broadcast address. An IP packet (datagram) that is sent to this address is broadcast to all hosts on the network.

The **netid** portion of a subnet does not have the same limitations. Often you are given the choice of reserving the first and last networks in a range as you would a **hostid**, but now this is rarely done in practice. More often, the first and last network in the netid range are used to provide more usable address space. Refer to "Subnets" on this page.

9. This is also an issue with other protocols, particularly Appletalk.

Computations for IP address 131.204.027.027 || table 9-3

	----------Class B----------		netid	hostid	
IP Address	131	.204	.027	.027	decimal
	8C	CC	1B	1B	hexadecimal
	1000 1100	1100 1100	0001 1011	0001 1011	binary
Subnet Mask	255	.255	.255	.000	decimal
	FF	FF	FF	00	hexadecimal
	1111 1111	1111 1111	1111 1111	0000 0000	binary
IP Address bitwise AND	1000 1100	1100 1100	0001 1011	0001 1011	decimal
Subnet Mask	1111 1111	1111 1111	1111 1111	0000 0000	hexadecimal
= Subnet Number	1000 1100	1100 1100	0001 1011	0000 0000	binary
Subnet Number	131	.204	.027	.000	decimal
	83	CC	1B	00	hexadecimal
	1000 1100	1100 1100	0001 1011	0000 0000	binary
Broadcast Address (Set host bits to 1)	131	.204	.27	.255	decimal
	83	CC	1B	FF	hexadecimal
	1000 0011	1100 1100	0001 1011	1111 1111	binary

A *subnet mask* (or *address mask*) is a bit mask that identifies which parts of an IP address correspond to the network address and subnet portion of the address. This mask has 1s in positions corresponding to the network and subnet numbers and 0s in the host number positions. When you perform a bitwise AND on an IP address and a subnet mask (Table 9-3), the result is an address that contains everything but the host address (**hostid**) portion.

There are several ways to represent a subnet mask: A network could have a subnet mask of 255.255.255.0 (decimal), FFFFFF00 (hexadecimal), or /24 (the

number of bits used for the subnet mask). If it were a class B network (of which 16 bits are already fixed), this yields 2^8 (24 total bits – 16 fixed bits = 8 bits, $2^8 = 256$) networks[10] with $2^8 – 2$ (256 – 2 = 254) hosts[11] on each network. If you do use a subnet mask, use netconfig to let the system know about it.

For example, when you divide the class C address 192.25.4.0 into eight subnets, you get a subnet mask of 255.255.255.224, FFFFFFE0, or /27 (27 1s). The eight resultant networks are 192.25.4.0, 192.25.4.32, 192.25.4.64, 192.25.4.96, 192.25.4.128, 192.25.4.160, 192.25.4.192, and 192.25.4.224. You can use a Web-based subnet mask calculator to calculate subnet masks (page 1401). To use this calculator to determine the preceding subnet mask, use an IP host address of 192.25.4.0. Go to www.telusplanet.net/public/sparkman/netcalc.htm for a nice subnet calculator.

CIDR: Classless Inter-Domain Routing

CIDR (pronounced *cider*) allows groups of addresses that are smaller than a class C block to be assigned to an organization or ISP and further subdivided and parceled out. In addition, it helps to alleviate the potential problem of routing tables on major Internet backbone and peering devices becoming too large to manage.

The pool of available IPv4 addresses has been depleted to the point that no one gets a class A address anymore. The trend is to reclaim these huge address blocks, if possible, and recycle them into groups of smaller addresses. Also, as more class C addresses are assigned, routing tables on the Internet are filling up and causing memory overflows. The solution is to aggregate[12] groups of addresses into blocks and allocate them to ISPs which in turn subdivide these blocks and allocate them to customers. The address class designations (A, B, and C) described in the previous section are used less today, although subnets are still used. When you request an address block, your ISP usually gives as many addresses as you need and no more. The ISP aggregates several contiguous smaller blocks and routes them to your location. This aggregation is CIDR. Without CIDR, the Internet as we know it would not function.

For example, you might be allocated the 192.168.5.0/22 IP address block, which could support 2^{10} hosts (32 – 22 = 10). Your ISP would set its routers so that any packets going to an address in that block would be sent to your network. Internally,

10. The first and last networks are reserved in a manner similar to the first and last host, although the standard is flexible. You can configure your router(s) to reclaim the first and last networks in a subnet. Different routers have different techniques for reclaiming these networks.

11. Subtract 2 because the first and last host addresses on every network are reserved.

12. *Aggregate* means to join. In CIDR the aggregate of 208.178.99.124 and 208.178.99.125 is 208.178.99.124/23 (the aggregation of two class Cs).

your own routers might further subdivide this block of 1024 potential hosts into subnets, perhaps into four networks. Four networks require an additional two bits of addressing ($2^2 = 4$). You could set up your router to have four networks with this allocation: 192.168.5.0/24, 192.168.6.0/24, 192.168.7.0/24, and 192.168.8.0/24. Each of these networks could have 254 hosts. CIDR lets you arbitrarily divide networks and subnetworks into ever smaller blocks along the way. Each router has enough memory to keep track of the addresses it needs to direct and aggregates the rest. This scheme uses memory and address space efficiently. You could take 192.168.8.0/24 and further divided it into 16 networks with 14 hosts each. The 16 networks require four more bits ($2^4 = 16$), so you'd have 192.168.8.0/28, 192.168.8.16/28, 192.168.8.32/28, and so on to the last subnet of 192.168.8.240/16, which would have the hosts 192.168.8.241 through 192.168.8.254.

Hostnames

People generally find it easier to work with symbolic names than with numbers, and GNU/Linux provides several ways to associate hostnames with IP addresses. The oldest method is to consult a list of names and addresses that are stored in the **/etc/hosts** file:

```
$ cat /etc/hosts
127.0.0.1     localhost
130.128.52.1  gw-tcorp.tcorp.com   gw-tcorp
130.128.52.2  bravo.tcorp.com      bravo
130.128.52.3  hurrah.tcorp.com     hurrah
130.128.52.4  kudos.tcorp.com      kudos
```

The address 127.0.0.1 is reserved for the special hostname **localhost**, which serves as a hook for the system's networking software to operate on the local machine without going out onto a physical network. The names of the other systems are shown in two forms: in a *fully qualified domain* (FQDN) format that is meant to be unique and as a nickname that is unique locally but usually not unique over all the systems attached to the Internet.

As more hosts joined networks, storing these name-to-address mappings in a regular text file proved to be inefficient and inconvenient. The file grew ever larger and impossible to keep up-to-date. GNU/Linux supports NIS (Network Information Service, page 390) and NIS+, which were developed for use on Sun computers. Each of these network services stores information in a database. These solutions make it easier to find a specific address but are useful only for host information within a single administrative domain. Hosts outside the domain cannot access the information.

The solution is DNS (Domain Name Service, page 388). DNS effectively addresses the efficiency and update issues by arranging the entire network naming space as a hierarchy. Each domain in the DNS manages its own name space (addressing and name resolution), and each domain can easily query for any host or IP address by following the tree up or down the name space until the appropriate domain is found. By providing a hierarchical naming structure, DNS distributes name administration across the entire Internet.

IPv6

The explosive growth of the Internet has uncovered deficiencies in the design of the current address plan, most notably lack of addresses. Over the next few years, a revised protocol, named IPng (IP Next Generation), or IPv6 (IP version 6),[13] will be phased in (it may take longer; the phase-in is going quite slowly). This new scheme is designed to overcome the major limitations of the current approach and can be phased in gradually because it is compatible with the existing address usage. IPv6 makes it possible to assign many more unique Internet addresses (2^{128}, or 340 *undecillion* [10^{36}]) and offers support for security and performance control features.

IPv6

- Enables autoconfiguration. With IPv4 autoconfiguration is available via optional DHCP. With IPv6 autoconfiguration is mandatory, making it easy for hosts to configure their IP addresses automatically.
- Reserves 24 bits in the header for advanced services, such as resource reservation protocols, better backbone routing, and improved traffic engineering.
- Makes multicast protocols mandatory and uses them extensively. In IPv4 multicast, which improves scalability, is optional.
- Aggregates address blocks more efficiently because of the huge address space. This aggregation obsoletes *NAT* (page 1481), which decreased scalability and introduced protocol issues.
- Provides a simplified packet header that allows hardware accelerators to work better.

A sample IPv6 address is fe80::a00:20ff:feff:5be2/10. Each group of four hexadecimal digits is equivalent to a number between 0 and 65536 (16^4). A pair of adjacent colons indicates a hex value of 0x0000, and leading 0s need not be shown.

13. IPv5 referred to an experimental real-time stream protocol named ST; thus the jump from IPv4 to IPv6.

With eight sets of hexadecimal groupings, you have $65{,}536^8 = 2^{128}$ possible addresses. In an IPv6 address on a host with the default autoconfiguration, the first characters in the address are always fe80. The last 64 bits hold an interface ID designation which is often the *MAC address* (page 1478) of the Ethernet controller on the system.

Communicate over a Network

Many commands that you can use to communicate with other users on a single computer system have been extended to work over a network. Three examples of extended utilities, all of which were introduced in Chapter 3, are electronic mail programs (such as pine), information-gathering utilities (such as finger), and communications utilities (such as talk). These utilities are examples of the UNIX philosophy: Instead of creating a new, special-purpose tool, modify an existing one.

Many utilities understand a convention for the format of network addresses: **user@host** (spoken as *user at host*). When you use an **@** sign in an argument to one of these utilities, the utility interprets the text that follows as the name of a remote host. When it does not include an **@** sign, a utility assumes that you are requesting information from or corresponding with someone on your LAN.

The prompts shown in the examples in this chapter include the hostname of the machine you are using. When you frequently use more than one system over a network, you may find it difficult to keep track of which system you are using at any particular moment. If you set your prompt to include the hostname of the current system, it will always be clear which system you are using. To identify the computer you are using, run hostname or **uname –n**:

```
$ hostname
kudos
```

See pages 579 bash tcsh and 749 zsh for information on how you can change your prompt.

finger: Displays Information about Remote Users

The finger utility displays information about one or more users on a system. This utility was designed for local use, but when networks became popular, it was obvious that finger should be enhanced to reach out and collect information remotely. In the following examples, finger displays information about all the users logged in on the system named **bravo**:

```
[kudos]$ finger @bravo
[bravo.tcorp.com]
Login     Name            Tty   Idle  Login Time   Office    Office Phone
root      root             *1    1:35  Oct 22  5:00
alex      Alex Watson       4          Oct 22 12:23 (kudos)
alex      Alex Watson       5       19  Oct 22 12:33 (:0)
jenny     Jenny Chen        7     2:24  Oct 22  8:45 (:0)
hls       Helen Simpson    11       2d  Oct 20 12:23 (:0)
```

A user's login name in front of the **@** sign causes finger to display information from the remote system for the specified user only. If there are multiple matches for that name on the remote system, finger displays the results for all of them.

```
[kudos]$ finger alex@bravo
[bravo.tcorp.com]
Login     Name            Tty   Idle  Login Time   Office    Office Phone
alex      Alex Watson       4          Oct 22 12:23 (kudos)
alex      Alex Watson       5       19  Oct 22 12:33 (:0)
```

The finger utility works by querying a standard network service, the **fingerd** daemon, that runs on the system being queried. Although this service is supplied with Red Hat Linux, some sites choose not to run it to minimize the load on their systems, reduce security risks, or maintain privacy. When you use finger to obtain information about someone at such a site, you will see an error message or nothing at all. It is the remote **fingerd** daemon that determines how much information to share with your system and in what format. As a result, the report displayed for any given system may differ from the preceding examples.

The fingerd Daemon || security

The finger daemon (fingerd) gives away system account information that can aid a malicious user. Some sites disable finger or randomize user account IDs to make a malicious user's job more difficult. Disable finger by giving the following command as root: chkconfig finger off.

The information for remote finger looks much the same as it does when finger runs on your local system, with one difference: Before displaying the results, finger reports the name of the remote system that answered the query (**bravo**, as shown in brackets in the preceding example). The name of the host that answers may be different from the system name you specified on the command line, depending on how the finger daemon service is configured at the remote end. In some cases several hostnames may be listed if one finger daemon contacts another to retrieve the information.

Sending Mail to a Remote User

Given a user's login name on a remote system and the name of the remote system or its domain, you can use an e-mail program, such as pine (page 87), to send a message over the network or the Internet, using the **@** form of an address:

```
jenny@bravo
```

or

```
jenny@tcorp.com
```

Although the **@** form of a network address is recognized by many GNU/Linux utilities, you may find that you can reach more remote computers with e-mail than with the other networking utilities described in this chapter. The reason for this disparity is that the mail system can deliver a message to a host that does not run IP, even though it appears to have an Internet address. The message may be routed over the network, for example, until it reaches a remote system that has a point-to-point, dial-up connection to the destination system. Other utilities, such as talk, rely on IP and operate only between networked hosts.

Mailing List Servers

A mailing list server (listserv[14]) allows you to create, manage, and administrate an e-mail list. An electronic mailing list provides a means for people interested in a topic to participate in an electronic discussion and for a person to disseminate information periodically to a potentially large mailing list. One of the most powerful features of most list servers is the ability to archive e-mail postings to the list, create an archive index, and allow users to retrieve postings from the archive based on keywords or discussion threads. Typically you can subscribe and unsubscribe from the list with or without human intervention. The owner of the list can restrict who can subscribe, unsubscribe, and post messages to the list. Popular list servers include LISTSERV (www.lsoft.com), Lyris (www.lyris.com), Majordomo (www.greatcircle.com/majordomo), Mailman (www.list.org), and ListProc (www.listproc.net). Red Hat maintains several mailing lists (https://listman.redhat.com) and list archives (www.redhat.com/mailing-lists). Use a browser to search on `linux mailing list` to find (many) other lists.

14. Although the term *listserv* is sometimes used generically to include many different list server programs, it is a specific product and a registered trademark of L-soft International, Inc.: LISTSERV (www.l-soft.com).

Network Utilities

To make use of a networked environment, it made sense to extend certain tools, some of which have already been described. Networks also created a need for new utilities to control and monitor them; this led to ideas for new tools that took advantage of network speed and connectivity. This section describes concepts and utilities for systems attached to a network; without a network connection, they are of little use.

Trusted Hosts

Some commands, including rcp and rsh, work only if the remote system trusts your local computer (that is, the remote system knows your local computer and believes that it is not pretending to be a system that it is not). The **/etc/hosts.equiv** file lists trusted systems. For reasons of security, Superuser account does not rely on this file to identify trusted Superusers from other systems.

Host-based trust is largely obsolete. The rcp, rlogin, and rsh commands[15] are deprecated in favor of ssh (page 374) and scp (page 376). Because there are many ways to subvert trusted host security, including subverting DNS systems and *IP spoofing* (page 1474), authentication based on IP address is widely regarded as insecure and obsolete. In a small homogeneous network of machines with local DNS control, it can be "good enough." The ease of use in these situations may outweigh the security concerns.

Do Not Share Your Login Account || security

You can use a .rhosts file to allow another user to log in as you from a remote system without knowing your password. *This setup is not recommended.* Do not compromise the security of your files or the entire system by sharing your login account. Use ssh and scp instead of rsh and rcp whenever possible.

15. The daemons necessary to set up a trusted host server are not included in default configurations of recent Red Hat releases, but the programs needed to access such a server are. This allows backward compatibility without propagating old technology. The **rsh-server*.rpm** package provides the **rlogind** and **rshd** daemons as well as files necessary to set up an rcp, rlogin, and/or rsh server.

ssh: Logs in or Runs a Command on a Remote Computer

You can use the secure ssh utility to log in on a remote system over the network. You might choose to use a remote system to access a special-purpose application, use a device that is available only on that system, or because you know that the remote system is faster or not as busy as your local computer. While traveling, many people use ssh on a laptop to log in on a system at headquarters. From a GUI you are able to use many systems simultaneously by logging in on each, using a different terminal emulator window.

You can log in on a remote machine that is running the **sshd** daemon and that you have an account on. For information on configuring ssh, see page 1029. All communication under ssh, including your name and password, is encrypted. When your login name is the same on the local and remote machines, give the command **ssh** *hostname*, where *hostname* is the name of the machine that you want to log in on:

```
[bravo]$ ssh kudos
alex@kudos's password:
Last login: Sat Sep 14 06:51:59 from bravo
Have a lot of fun...
You have new mail.
[kudos]$
...
[kudos]$ logout
Connection to kudos closed.
[bravo]$
```

After you supply your password, you are running a shell on the remote machine. When you log out, the connection is broken, and you resume using your local computer. To log in with a user name different from the one you are using on the local machine, give the command **ssh** *user@hostname*, where *user* is your login name on the remote machine named *hostname*:[16]

```
[bravo]$ ssh watson@kudos
watson@kudos's password:
...
[kudos]$
```

The ssh utility also allows you to run a command on a remote system without logging in on that system. When you need to run more than one command, it is usually easier to log in and run the commands on the remote machine. The next example runs ls on the **memos** directory on the remote system **kudos**. The example assumes that the user running the command has a login on **kudos** and that **memos** is in the user's home directory on **kudos**:

16. The **–l** (ell) option performs the same function: **ssh –l watson kudos**.

```
[bravo]$ ssh kudos ls memos
alex@kudos's password:
memo.0921
memo.draft
[bravo]$
```

Suppose that a file named **memo.new** is on your local machine and that you cannot remember whether it contains certain changes or whether you made these changes to the file named **memo.draft** on the system named **kudos**. You could copy **memo.draft** to your local system and run diff (page 67) on the two files, but then you would have three similar copies of the file spread across two systems. If you are not careful about removing the old copies when you are done, you may be confused again in a few days. Instead of copying the file, you can use ssh:

```
[bravo]$ ssh kudos cat memos/memo.draft | diff memos.new -
```

When you run ssh, standard output of the command run on the remote machine is passed to the local shell as though the command had been run in place on the local machine. Unless you quote characters that have special meaning to the shell, they are interpreted by the local machine. In the preceding example the output of the cat command on **kudos** is sent through a pipe on **bravo** to diff (running on **bravo**), which compares the local file **memos.new** to standard input (**–**). The following command line has the same effect but causes diff to run on the remote system:

```
[bravo]$ cat memos.new | ssh kudos diff - memos/memo.draft
```

Standard output from diff on the remote system is sent to the local shell, which displays it on the screen (because it is not redirected). Refer to page 1321 in Part III for more information on ssh.

Optional

The ssh utility can tunnel other protocols. You can secure protocols including POP, X, IMAP, and WWW using ssh as a virtual private network (VPN) between the two systems. Assume that you have a POP client on your local machine, the POP server is on a remote network that is protected by a firewall, and that you can access the remote network only using ssh. You can tunnel the POP protocol, which uses port 110, through an ssh tunnel. In this example, **kudos** is the firewall gateway machine, **pophost** is the POP server, and 1550 is a local port that you selected to use on your end of the tunnel.

```
$ ssh -N -L 1550:pophost:110 kudos
```

The **–N** option causes ssh not to execute any remote commands: ssh works only as a private network to forward ports. You can forward ports in either direction and in combination by using the –L and –R flags. See the ssh man page for details.

Once you are authenticated, you can set your POP client so that the POP server is **localhost** and the POP port is 1550. Then, when the client fetches e-mail, it makes a connection to port 1550 on the local machine which is forwarded through the ssh tunnel to **kudos** and then to **pophost** port 110 where the real daemon is running.

scp: Copies a file from/to a Remote Computer

The scp (secure copy) utility copies a file from one computer to another on a network. Using ssh to transfer files, scp uses the same authentication mechanism as ssh and therefore provides the same security. The scp utility asks you for a password when it is needed for security. The format of an scp command is

scp [fromhost:]source-file [tohost:][destination-file]

You can copy from or to your local machine or between two remote machines. When you specify a simple, or relative, filename, it is assumed to be relative to your home directory on a remote machine and relative to your working directory on your local machine. An absolute pathname describes a path from the root directory on any machine. Make sure that you have read permission to the file you are copying and write permission for the directory you are copying it into. In the following example, Alex uses scp to copy **rain.jpg** from his working directory on **bravo** (which happens to be his home directory) to his home directory on **kudos**:

```
[alex@bravo alex]$ scp rain.jpg kudos:
alex@kudos's password:
rain.jpg        100% |**********************| 30161   00:00
```

As the transfer progresses, the percent and number of bytes transferred increase and the time remaining decreases. The asterisks provide a visual representation of the progress of the transfer.

Use the **–r** option to copy a directory recursively. See the scp man page for more information.

telnet: Logs in on a Remote Computer

You can use the TELNET protocol to interact with a remote computer. The telnet utility, a user interface to this protocol, is older than ssh and is not secure but may work where ssh is not available (there is more non-UNIX support for TELNET

access than there is for ssh access). In addition, many legacy devices, such as terminal servers and network devices, do not support ssh.

```
[bravo]$ telnet kudos
Trying 130.128.52.2...
Connected to kudos.tcorp.com
Escape character is '^]'.

Welcome to SuSE Linux 7.3 (i386) - Kernel 2.4.10-4GB (2).
kudos login: watson
Password:
You have old mail in /var/mail/watson.
Last login: Mon Feb 25 14:46:55 from bravo.tcorp.com
watson@kudos:~>
.
.
.
watson@kudos:~> logout
Connection closed by foreign host.
[bravo]$
```

When you connect to a remote UNIX or GNU/Linux system through telnet, you are presented with a regular `login:` prompt. Unless you specify differently, the ssh utility assumes that your login name on the remote system matches that on the local system. Because telnet is designed to work with non-UNIX/Linux systems, it makes no such assumptions.

telnet Is Not Secure || security

Whenever you enter sensitive information, such as your password, while you are using telnet, it is transmitted in cleartext and can be read by someone who is listening in on the session.

Another difference between these two utilities is that telnet allows you to configure many special parameters, such as how RETURNs or interrupts are processed. When using telnet between two UNIX/Linux systems, you rarely need to change any parameters.

When you do not specify the name of a remote host on the command line, telnet runs in an interactive mode. The following example is equivalent to the previous telnet example:

```
[bravo]$ telnet
telnet> open kudos
Trying 130.128.52.2...
Connected to kudos.tcorp.com
Escape character is '^]'.
...
```

Before connecting you to a remote system, telnet tells you what your *escape character* is; in most cases it is ^] (the ^ represents the CONTROL key on your keyboard). When you press CONTROL-], you escape to telnet's interactive mode. Continuing the preceding example:

```
[kudos]$ CONTROL-]
telnet> ?
```

(displays help information)

```
telnet> close
Connection closed.
[bravo]$
```

When you enter a question mark in response to the telnet> prompt, telnet displays a list of its commands. The **close** command ends the current telnet session, returning you to your local system. To get out of telnet's interactive mode and resume communication with the remote system, press RETURN in response to a prompt.

It has been possible to use telnet to access special remote services at sites that have chosen to make such services available. However, many of these services, such as the U.S. Library of Congress Information System (LOCIS), have moved to the Web, so you can now obtain the same information by using a Web browser.

ftp: Transfers Files over a Network

You can use the ftp[17] (file transfer protocol) utility to transfer files between systems on a network. This interactive utility allows you to browse through a directory on the remote system to identify files you may want to transfer:

```
[kudos]$ ftp bravo
Connected to bravo.tcorp.com.
220 bravo.tcorp.com FTP server (Version wu-2.6.1-20) ready.
Name (bravo:alex): watson
331 Password required for watson.
Password:
230 User watson logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bin
200 Type set to I.
ftp> cd memos
250 CWD command successful.
```

17. The sftp (secure file transfer program) utility, which is included with Red Hat Linux, is similar to ftp but works over an encrypted ssh connection. See the sftp man page for more information.

```
ftp> put memo.921
local: memo.921 remote: memo.921
200 PORT command successful.
227 Entering Passive Mode (192,168,0,1,225,45)
150 Opening BINARY mode data connection for memo.921. (8401 bytes)
100% |*****************************| 8401  3.38 KB/s  00:00 ETA
226 Transfer complete.
8401 bytes received in 00:02 (3.38 KB/s)
ftp> quit
221-You have transferred 56064 bytes in 1 files.
221-Total traffic for this session was 56485 bytes in 1 transfers.
221-Thank you for using the FTP service on bravo.tcorp.com.
221 Goodbye.
[kudos]$
```

The remote system prompts you for a login name and password. By default the system expects that your login name is the same on both systems; just press RETURN if it is. In this case it is not, so Alex enters **watson** before pressing RETURN. Then he enters his password.

Although it is not necessary in this case, Alex gives a **bin** (binary) command as a matter of habit; he always establishes binary transfer mode as soon as he logs in.[18] With ftp in binary mode, you can transfer ASCII and binary files. ASCII mode can guarantee the successful transfer of ASCII files only.

Binary mode transfers an exact, byte-for-byte image of a file. ASCII mode performs end-of-line conversions between different systems and is consequently slower than binary mode. DOS/MS Windows, Macintosh, and UNIX/Linux each use different characters to indicate the end of a line of text. For example, Microsoft operating systems use a RETURN (CONTROL-M) followed by a NEWLINE (CONTROL-J) to mark the end of a line, whereas UNIX/Linux uses a NEWLINE by itself. Use ASCII mode to transfer plain text files (sometimes indicated by a **.txt** filename extension) only. Transfer Microsoft Word and other word processing documents in binary mode, as they are not plain text files. Unless you specifically need to convert the end-of-line characters, use binary mode.

Before transferring the file, Alex uses ftp's **cd** command to change directories *on the remote system* (use **lcd** to change directories on the local system). Then the **put** command, followed by the filename, transfers the file to the remote system in the remote working directory (**memos**).

The ftp utility makes no assumptions about filesystem structure, because you can use ftp to exchange files with non-UNIX/Linux systems (whose filenaming conventions may be different).

18. The ncftp utility, which is included with Red Hat Linux, takes care of the binary issue and others automatically. It is a front end for standard ftp that runs in place of ftp. See the ncftp man page for more information.

Anonymous FTP

Systems often provide ftp access to anyone on a network by providing a special login: **anonymous** (you can usually use the login name **ftp** in place of **anonymous**). The anonymous FTP user is usually restricted to a portion of a filesystem that has been set aside to hold files that are to be shared with remote users. Traditionally any password is acceptable for anonymous FTP; by convention you are expected to give your e-mail address. Some sites reject your connection if they cannot identify the name of your computer or if you supply a password that doesn't match the name of your site. Alex can enter **alex@tcorp.com** in response to the password prompt.

While using ftp, you can type **help** at any `ftp>` prompt to see a list of commands. For using Mozilla to perform an anonymous FTP transfer, see "Downloading a File" on page 408. Refer to page 1180 in Part III for more information on ftp.

ping: Tests a Network Connection

The ping[19] utility (http://ftp.arl.mil/~mike/ping.html) sends an ECHO_REQUEST packet to a remote computer. This packet causes the remote system to send back a reply. This is a quick way to verify that a remote system is available, as well as to check how well the network is operating, such as how fast it is or whether it is dropping data packets. The protocol ping uses is ICMP (Internet Control Message Protocol). Without any options ping tests the connection once per second until you abort the execution with CONTROL-C.

```
[kudos]$ ping tsx-11.mit.edu
PING tsx-11.mit.edu (18.86.0.44) from 10.0.1.5 : 56(84) bytes of data.
64 bytes from TSX-11.MIT.EDU (18.86.0.44): icmp_seq=0 ttl=48 time=500.199 msec
64 bytes from TSX-11.MIT.EDU (18.86.0.44): icmp_seq=1 ttl=48 time=518.703 msec
64 bytes from TSX-11.MIT.EDU (18.86.0.44): icmp_seq=2 ttl=48 time=516.304 msec
64 bytes from TSX-11.MIT.EDU (18.86.0.44): icmp_seq=3 ttl=48 time=95.807 msec
CONTROL-C

--- tsx-11.mit.edu ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/mdev = 95.807/407.753/518.703/180.243 ms
```

In this example the remote system named **tsx-11.mit.edu** is up and available to you over the network.

19. The name ping mimics the sound of a sonar burst used by submarines to identify and communicate with each other. The word ping also expands to Packet Internet Groper.

By default ping sends packets containing 64 bytes (56 data bytes and 8 bytes of protocol header information). In the preceding example four packets were sent to the system **tsx-11.mit.edu** before the user interrupted ping by pressing CONTROL-C. The four-part number in parentheses on each line is the remote system's IP address. A packet sequence number (called **icmp_seq**) is also given. If a packet is dropped, a gap occurs in the sequence numbers. The round-trip time is listed last, in microseconds; this represents the time that elapsed from when the packet was sent from the local system to the remote system until the reply from the remote system was received by the local system. This time is affected by the distance between the two systems, as well as by network traffic and the load on both computers. Before it terminates, ping summarizes the results, indicating how many packets were sent and received, as well as the minimum, average, maximum, and mean deviation round-trip times it measured.

When ping Cannot Connect

|| tip

If unable to contact the remote system, ping continues trying until you interrupt it with CONTROL-C. There may be several reasons why a system does not answer: The remote computer may be down, the network interface or some part of the network between the systems may be broken, there may be a software failure, or the remote machine may be set up, for reasons of security, not to return pings (try pinging www.microsoft.com or www.ibm.com).

traceroute: Traces a Route over the Internet

The traceroute utility, supplied with Red Hat Linux, traces the route an IP packet follows, including all the intermediary points traversed (called *network hops*), to its destination (the argument to traceroute—an Internet host). It displays a numbered list of host names, if available, and IP addresses, together with the round-trip time it took for a packet to get to each router along the way and an acknowledgment to get back. You can put this information to good use when you are trying to determine where a network bottleneck is.

The traceroute utility has no concept of the path from one host to the next; it simply sends out packets with increasing *TTL* values. TTL is an IP header field that indicates how many more hops the packet should be allowed to make before being discarded or returned. In the case of a traceroute packet, the packet is returned by the host that has the packet when the TTL value is zero. The result is a list of hosts that the packet travels through to get to its destination.

The traceroute utility can help you solve routing configuration problems and routing path failures. When you cannot reach a host, use traceroute to see what path the packet follows, how far it gets, and what the delay is.

The next example is the output of traceroute following a route from a local computer to **www.linux.org**. The first line tells you the IP address of the target, the maximum number of hops that will be traced, and the size of the packets that will be used. Each numbered line contains the name and IP address of the intermediate destination, followed by the time it takes a packet to make a round-trip to that destination and back. The traceroute utility sends three packets to each destination; thus there are three times on each line. Line 1 shows the statistics when a packet is sent to the local gateway (under 3 ms). Lines 4–6 show it bouncing around Mountain View (California) before it goes to San Jose. Between hops 13 and 14 the packet travels across the United States (San Francisco to somewhere in the East). By hop 18 the packet has found **www.linux.org**. The traceroute utility displays asterisks when it does not receive a response. Each asterisk indicates that traceroute has waited three seconds.

```
$ /usr/sbin/traceroute www.linux.org
traceroute to www.linux.org (198.182.196.56), 30 hops max, 38 byte packets
 1  gw.localco.com. (204.94.139.65)  2.904 ms  2.425 ms  2.783 ms
 2  covad-gw2.meer.net (209.157.140.1)  19.727 ms  23.287 ms  24.783 ms
 3  gw-mv1.meer.net (140.174.164.1)  18.795 ms  24.973 ms  19.207 ms
 4  d1-4-2.a02.mtvwca01.us.ra.verio.net (206.184.210.241)  59.091 ms d1-10-0-0-200.a03.
       mtvwca01.us.ra.verio.net (206.86.28.5)  54.948 ms  39.485 ms
 5  fa-11-0-0.a01.mtvwca01.us.ra.verio.net (206.184.188.1)  40.182 ms  44.405 ms 49.362 ms
 6  p1-1-0-0.a09.mtvwca01.us.ra.verio.net (205.149.170.66)  78.688 ms  66.266 ms 28.003 ms
 7  p1-12-0-0.a01.snjsca01.us.ra.verio.net (209.157.181.166) 32.424 ms 94.337 ms 54.946 ms
 8  f4-1-0.sjc0.verio.net (129.250.31.81)  38.952 ms  63.111 ms  49.083 ms
 9  sjc0.nuq0.verio.net (129.250.3.98)  45.031 ms  43.496 ms  44.925 ms
10  mae-west1.US.CRL.NET (198.32.136.10)  48.525 ms  66.296 ms  38.996 ms
11  t3-ames.3.sfo.us.crl.net (165.113.0.249)  138.808 ms  78.579 ms  68.699 ms
12  E0-CRL-SFO-02-E0X0.US.CRL.NET (165.113.55.2)  43.023 ms  51.910 ms  42.967 ms
13  sfo2-vva1.ATM.us.crl.net (165.113.0.254)  135.551 ms  154.606 ms  178.632 ms
14  mae-east-02.ix.ai.net (192.41.177.202)  158.351 ms  201.811 ms  204.560 ms
15  oc12-3-0-0.mae-east.ix.ai.net (205.134.161.2)  202.851 ms  155.667 ms  219.116 ms
16  border-ai.invlogic.com (205.134.175.254)  214.622 ms *  190.423 ms
17  router.invlogic.com (198.182.196.1)  224.378 ms  235.427 ms  228.856 ms
18  www.linux.org (198.182.196.56)  207.964 ms  178.683 ms  179.483 ms
```

host and dig: Queries Internet Name Servers

The host utility looks up an IP address given a name or vice versa. This utility is easy to use and replaces nslookup in its simplest case. The following example shows how to use host to look up the domain name of a machine, given an IP address:

```
$ host 140.174.164.2
2.164.174.140.in-addr.arpa. domain name pointer ns.meer.net.
```

You can also use host to determine the IP address of a domain name:

```
$ host ns.meer.net
ns.meer.net. has address 140.174.164.2
```

The dig (domain information groper) utility queries DNS servers and individual machines for information about a domain. A powerful utility, dig has many features that you may never use. It is more involved than host and replaces nslookup in its complex cases. The following dig command uses the keyword **any** to get any available information about the **upstate.edu** domain.

```
# dig any upstate.edu
; <<>> DiG 9.1.3 <<>> any upstate.edu
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30224;; flags: qr rd ra; QUERY: 1,
ANSWER: 11, AUTHORITY: 5, ADDITIONAL: 7

;; QUESTION SECTION:
;upstate.edu.                    IN      ANY

;; ANSWER SECTION:
upstate.edu.             74567   IN      NS      dns.duc.upstate.edu.
upstate.edu.             74567   IN      NS      dns.eng.upstate.edu.
upstate.edu.             74567   IN      NS      dns.acesag.upstate.edu.
upstate.edu.             74567   IN      NS      dns.upstate.edu.
upstate.edu.             74567   IN      NS      nr01.netmgt.upstate.edu.
upstate.edu.            83413   IN      SOA     dns.upstate.edu. bailebn.noc.upstate.edu.
2002022106 3600 900 259200 86400
upstate.edu.             83438   IN      MX      10 ducserv6.duc.upstate.edu.
upstate.edu.             83438   IN      MX      10 ducserv3c.duc.upstate.edu.
upstate.edu.             83438   IN      MX      10 ducserv6b.duc.upstate.edu.
upstate.edu.             83438   IN      MX      20 ducserv3.duc.upstate.edu.
upstate.edu.             83438   IN      MX      20 ducserv3b.duc.upstate.edu.

;; AUTHORITY SECTION:
upstate.edu.             74567   IN      NS      dns.duc.upstate.edu.
upstate.edu.             74567   IN      NS      dns.eng.upstate.edu.
upstate.edu.             74567   IN      NS      dns.acesag.upstate.edu.
upstate.edu.             74567   IN      NS      dns.upstate.edu.
upstate.edu.             74567   IN      NS      nr01.netmgt.upstate.edu.

;; ADDITIONAL SECTION:
dns.duc.upstate.edu.     2867    IN      A       192.168.2.10
dns.eng.upstate.edu.     68182   IN      A       192.168.10.13
dns.acesag.upstate.edu.  132867  IN      A       192.168.46.50
dns.upstate.edu.         160958  IN      A       192.168.41.3

nr01.netmgt.upstate.edu. 74567   IN      A       192.168.253.191ducserv6.duc.upstate.edu.
83438  IN      A       192.168.2.27
ducserv3c.duc.upstate.edu. 83438 IN      A       192.168.2.148

;; Query time: 29 msec
;; SERVER: 140.174.164.2#53(140.174.164.2)
;; WHEN: Thu Feb 28 15:54:33 2002
;; MSG SIZE  rcvd: 499
```

The dig utility displays a lot of information.

- The Authority Section specifies the primary name servers, and the Additional Section specifies the IP addresses that correspond to the names in the Authority Section.
- The SERVER line (toward the end) specifies the name and IP address of the DNS server that the local system uses: This is where dig gets its information.
- The second column specifies the *TTL* (page 1499) in seconds.
- IN in the third column is the query class and indicates that this is an Internet class query.
- NS, SOA, MX, NS, or A in the fourth column specifies the type of information (DNS query type) that the row holds:
  - The NS (name server) record(s) specify name servers that **upstate.edu** uses. An NS record is meaningful only when you query a domain.
  - The MX (mail exchanger) record(s) specify a mail server for the domain you are querying. The **upstate.edu** domain has several mail servers. The lower the preference value (the number before the mail server domain in the right column), the higher the priority (**ducserv6b** is always tried before **ducserv3**).
  - There is one SOA (start or zone of authority) for a given domain. The SOA
    - Is the authoritative primary DNS for the domain.
    - Defines who the point of contact is for the domain.
    - Controls the TTL for records from the DNS.
    - Controls how often another name server will retry the domain's name server.
    - Controls when another name server will timeout when trying to contact the domain's name server.
  - The A (network Address) record specifies in the last column the IP address that corresponds to the domain name in the first column.

The dig utility has many query types. The **any** type is used in the preceding example. You can also use **mx**, **ns**, **soa**, and others. Refer to the dig man page for more details.

whois: Looks Up Information about an Internet Site

The whois utility queries a whois server for information about an Internet site. This utility returns site contact and InterNIC or other registry information that can help

you track down the person responsible for a site: Perhaps that person is sending you or your company *spam* (page 1493). Many sites on the Internet are easier to use and faster than whois. Use a browser to search on whois or go to www.netsol.com/cgi-bin/whois/whois or www.ripe.net/perl/whois to get started.

When you search by name, whois may return more than one entry. In the following example, whois returns SOBELL.NET and SOBELL.COM when queried for sobell:

```
$ whois sobell
[whois.crsnic.net]

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

SOBELL.NET
SOBELL.COM

To single out one record, look it up with "xxx", where xxx is one of the
of the records displayed above. If the records are the same, look them up
with "=xxx" to receive a full display for each record.

>>> Last update of whois database: Tue, 26 Feb 2002 05:22:08 EST <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

When you do not specify a whois server, whois defaults to **whois.crsnic.net**. Set the **NICNAMESERVER** or **WHOISSERVER** shell variables, or use the **–h** option to whois to specify a different whois server.

To obtain information on a domain name, specify the complete domain name as in the following example:

```
$ whois sobell.com
[whois.crsnic.net]

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: SOBELL.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS.MEER.NET
   Name Server: NS2.MEER.NET
   Updated Date: 05-nov-2001
```

```
>>> Last update of whois database: Tue, 26 Feb 2002 05:22:08 EST <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.

Registrant:
Sobell Associates Inc (SOBELL-DOM)
   PO Box 1089
   Menlo Park, CA 94026
   US

   Domain Name: SOBELL.COM

   Administrative Contact, Billing Contact:
      Sobell, Mark  (MS989)  sobell@MEER.NET
      Sobell Associates Inc
      PO Box 1089
      Menlo Park, CA 94026
[No phone]
   Technical Contact:
      meer.net hostmaster  (MN85-ORG)  hostmaster@MEER.NET
      meer.net
      po box 390804
      Mountain View, CA 94039
      US
      +1.888.844.6337
      Fax- +1.888.844.6337

   Record last updated on 09-Apr-2000.
   Record expires on 08-Apr-2004.
   Record created on 07-Apr-1995.
   Database last updated on 26-Feb-2002 01:57:00 EST.

   Domain servers in listed order:

   NS.MEER.NET                140.174.164.2
   NS2.MEER.NET               216.206.136.2
```

Several top-level registries serve various regions of the world. The ones you are most likely to use are

- North American Registry: **whois.arin.net**
- European Registry: **www.ripe.net**
- Asia-Pacific Registry: **www.apnic.net**
- American Military: **whois.nic.mil**
- American Government: **www.nic.gov**

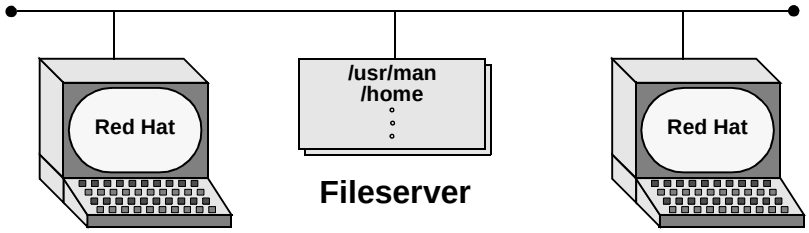

Figure 9-2 A fileserver

Distributed Computing

When many similar systems are on a network, it is often desirable to share common files and utilities among them. For example, a system administrator might choose to keep a copy of the system documentation on one computer's disk and to make those files available for all remote systems. In this case the system administrator configures the files so that users who need to access the online documentation are not aware that the files are stored on a remote system. This type of setup, which is an example of *distributed computing*, not only conserves disk space but also allows you to update one central copy of the documentation rather than tracking down and updating copies scattered throughout the network on many different systems.

Figure 9-2 illustrates a *fileserver* that stores the system manual pages and users' home directories. With this arrangement, a user's files are always available to that user—no matter which system the user is on. Each system's disk might contain a directory to hold temporary files, as well as a copy of the operating system. For more information refer to "exportfs: Stores Permissions to Mount Local Filesystems" on page 1018 and "autofs: Automatically Mounts Filesystems" on page 979.

The Client/Server Model

Although there are many ways to distribute computing tasks on hosts attached to a network, the client/server model dominates UNIX and GNU/Linux system networking. A server system offers services to its clients and is usually a central resource. In Figure 9-2 the system that acts as the documentation repository is a server, and all the systems that contact it to display information are clients. Some servers are designed to interact with specific utilities, such as Web servers and browser clients. Other servers, such as those supporting DNS, communicate with one another in addition to answering queries from a variety of clients; in other words, a server can act as a client when it queries another server.

The client/server terminology also applies to processes that may be running on one or more systems. A server process may control a central database while client processes send queries to the server and collect replies. In this case the client and server processes may be running on the same computer. The client/server model underlies most of the network services described in this chapter.

DNS: Domain Name Service

DNS is a distributed service: Name servers on thousands of machines around the world cooperate to keep the database up-to-date. The database itself, which contains the information that maps hundreds of thousands of alphanumeric hostnames into numeric IP addresses, does not exist in one place. That is, no system has a complete copy of the database. Instead each system that runs DNS knows about the hosts that are local to that site and how to contact other name servers to learn about other, nonlocal hosts.

Like the GNU/Linux filesystem, DNS is organized hierarchically. Each country has an ISO (International Standards Organization) country code designation as its domain name, (For example, `AU` represents Australia, `IL` is Israel, and `JP` is Japan; see www.iana.org/cctld/cctld.htm for a complete list.) Although the United States is represented in the same way (`US`) and uses the standard two-letter Postal Service abbreviations to identify the next level of the domain, only governments and a few organization use these codes. Schools in the `US` domain are represented by a third- (and sometimes second-) level domain: `k12`. For example, the domain name for Myschool in New York state could be www.myschool.k12.ny.us.

Following is a list of the six original, common, top-level domains. These domains are used extensively within the United States and, to a lesser degree, by users in other countries:

- COM Commercial enterprises
- EDU Educational institutions
- GOV Nonmilitary government agencies
- MIL Military government agencies
- NET Networking organizations
- ORG Other (often nonprofit) organizations

As this book was being written, the following additional top-level domains had been approved for use:

- AERO Air-transport industry
- BIZ Business
- COOP Cooperatives
- INFO Unrestricted use
- MUSEUM Museums
- NAME Name registries

As with Internet addresses, domain names used to be assigned by the Network Information Center (NIC [page 362]). Now they are assigned by several companies. A system's full name, referred to as its *fully qualified domain name* (FQDN), is unambiguous in the way that a simple hostname cannot be. The system **okeeffe.berkeley.edu** at the University of California, Berkeley (Figure 9-3) is not the same as one named **okeeffe.moma.org**, which might represent a host at the Museum of Modern Art. The domain name not only tells you something about where the system is located but also adds enough diversity to the name space to avoid confusion when different sites choose similar names for their systems.

Unlike the filesystem hierarchy, the top-level domain name in the United States appears last (reading from left to right). Also, domain names are not case sensitive. The names **okeeffe.berkeley.edu**, **okeeffe.Berkeley.edu**, and **okeeffe.Berkeley.EDU** refer to the same computer. Once a domain has been assigned, the local site is free to extend the hierarchy to meet local needs.

With DNS, mail addressed to **user@tcorp.com** can be delivered to the **tcorp.com** computer that handles the corporate mail and knows how to forward messages to user mailboxes on individual machines. As the company grows, the site administrator might decide to create organizational or geographical subdomains. The name **tcorp.ca.tcorp.com** might refer to a system that supports California offices, with **alpha.co.tcorp.com** dedicated to Colorado. Functional subdomains might be another choice, with **tcorp.sales.tcorp.com** and **alpha.dev.tcorp.com** representing the sales and development divisions, respectively.

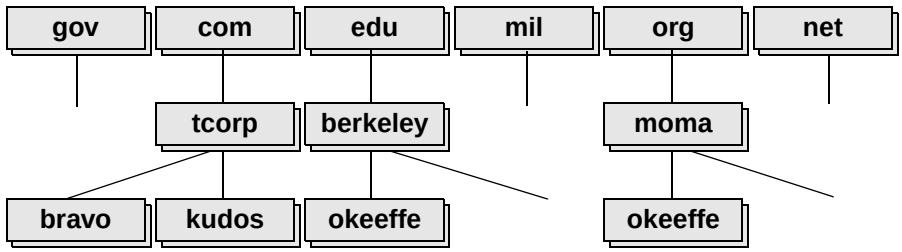

Figure 9-3 United States top-level domains

On GNU/Linux systems the most common interface to the DNS is BIND (Berkeley Internet Name Domain) software. BIND follows the client/server model. On any given local network, one or more systems may be running a name server, supporting all the local hosts as clients. When it wants to send a message to another host, a system queries the nearest name server to learn the remote host's IP address. The client, called a *resolver*, may be a process running on the same computer as the name server, or it may pass the request over the network to reach a server. To reduce network traffic and accelerate name lookups, the local name server has some knowledge of distant hosts. If the local server has to contact a remote server to pick up an address, when the answer comes back, the local server adds that to its internal table and reuses it for a while. The name server deletes the nonlocal information before it can become outdated. Refer to "TTL" on page 1499.

How the system translates symbolic hostnames into addresses is transparent to most users; only the system administrator of a networked system needs to be concerned with the details of name resolution. Systems that use DNS for name resolution are generally capable of communicating with the greatest number of hosts—more than would be practical to maintain in an **/etc/hosts** file or private NIS database.

Four common sources are used for host name resolution: NIS, NIS+, DNS, and system files (such as **/etc/hosts**). GNU/Linux does not ask you to choose among these sources; rather, the **nsswitch.conf** file (page 962) allows you to choose any of these sources, in any combination, and in any order.

NIS: Network Information Service

NIS is another example of the client/server paradigm. Sun Microsystems developed NIS to simplify the administration of certain common administrative files by maintaining them in a central database and having clients contact the database server to retrieve information. Just as the DNS addresses the problem of keeping multiple copies of the **hosts** file up-to-date, NIS keeps system-independent configuration files (such as **/etc/passwd**) current. Most networks today are *heterogeneous* (page 1470), and even though they run different varieties of UNIX or GNU/Linux, they have certain common attributes (such as the **passwd** file).

NIS was formerly named the *Yellow Pages*, and people still refer to it by this name. Sun renamed the service because another corporation holds the trademark to that name. The names of NIS utilities, however, are still reminiscent of the old name: ypcat displays an NIS database, ypmatch searches, and so on.

Consider the **/etc/group** file, which maps symbolic names to group ID numbers. If NIS is administering this configuration file on your system, you might see the following single entry instead of a list of group names and numbers:

```
$ cat /etc/group
+:*:*
...
```

When it needs to map a number to the corresponding group name, a utility encounters the plus sign (**+**) and knows to query the NIS server at that point for the answer. You can display the **group** database with the ypcat utility:

```
$ ypcat group
pubs::141:alex,jenny,scott,hls,barbara
...
```

Or you can search for a particular group name by using ypmatch:

```
$ ypmatch pubs group
pubs::141:alex,jenny,scott,hls,barbara
```

You can retrieve the same information by filtering the output of ypcat through grep, but ypmatch is more efficient because it searches the database directly, using a single process. The database name is not the full pathname of the file it replaces; the NIS database name is the same as the simple filename (**group**, not **/etc/group**). The ypmatch utility works only on the key for the table (the group name in the case of groups). When you want to match members of the group, the group number, or other fields of a map (such as the full name in the **passwd** map), you need to use ypcat with grep.

As with DNS, ordinary users need not be aware that NIS is managing system configuration files. Setting up and maintaining the NIS databases is a task for the system administrator; individual users and users on single-user GNU/Linux systems rarely need to work directly with NIS.

NFS: Network Filesystem

NFS lets you can work locally with files that are stored on a remote computer's disks. These files appear as if they are present on the local computer. The remote system is the fileserver (server); the local system is the client. The client makes requests of the server.

Unfortunately NFS is based on the trusted-host paradigm (page 373) and therefore has all the security shortcomings that plague services based on this paradigm.

NFS is configured by the person responsible for the system. When you work with a file, you may not be aware of where the file is physically stored. In many computer facilities today, user files are commonly stored on a central fileserver equipped with many large-capacity disk drives and devices that quickly and easily make backup copies of the data. A GNU/Linux system may be *diskless*, where a floppy disk (or CD-ROM) is used to start GNU/Linux and load system software

from another machine on the network. The Linux Terminal Server Project (LTSP.org) Web site says it all: "Linux makes a great platform for deploying diskless workstations that boot from a network server. The LTSP is all about running thin client computers in a GNU/Linux environment." Because a diskless workstation does not require a lot of computing power, you can give older, retired computers a second life by using them as diskless systems.

Another type of GNU/Linux system is the *dataless* system, in which the client has a disk but stores no user data (only GNU/Linux and the applications are kept on the disk). Setting up this type of system is a matter of choosing which filesystems are mounted remotely.

You can even *netboot* (page 1481) some machines. Red Hat includes the PXE (Preboot Execution Environment) server package for netbooting Intel machines. Older machines with netcard-mounted boot ROMs sometimes use tftp (trivial file transfer protocol) for netbooting. Non-Intel architectures have historically included netboot capabilities that Red Hat Linux supports. The Linux kernel contains the capability to be built to mount **root (/)**, using NFS.

Of the many ways to set up your system, the one you choose depends on what you want to do. Setting up these specialized boot configurations is not a trivial task. See the *Remote-boot mini HOWTO* for more information.

The df utility displays a list of the filesystems available on your system, along with the amount of disk space, free and used, on each. Filesystem names that are prepended with **hostname:** are available to you through NFS.

```
[bravo]$ pwd
/kudos/home/jenny
[bravo]$ df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda1               311027    189038    105926  64% /
...
/dev/sdc3              1336804        13   1267712   0% /c3
zach:/c                2096160   1896704    199456  90% /zach_c
zach:/d                2096450   1865761    230689  89% /zach_d
panda:/c               1542016    433568   1108448  28% /panda_c
panda:/d               1542208   1189026    353182  77% /panda_d
kudos:/home             198275     68408    119612  36% /kudos/home
```

In this example Jenny's home directory is stored on the remote system **kudos**. The **/home** filesystem on **kudos** is mounted on **bravo**, using NFS; as a reminder of its physical location, the system administrator has made it available using a pathname that includes the remote server's name. Filesystems on **zach** and **panda** are also available on **bravo**: These are the **C:** and **D:** drives on two MS Windows machines. Use the **–h** (human) option to df to make the output more intelligible. Refer to page 1147 in Part III for more information on df.

The physical location of your files should not matter to you; all the standard GNU/Linux utilities work with NFS-remote files the same way as they operate with

local files. At times, however, you may lose access to your remote files: Your computer may be up and running, but a network problem or a remote system crash may make these files temporarily unavailable: When you try to access a remote file, you get an error message, such as `NFS server kudos not responding`. When your system can contact the remote server again, you see a message, such as `NFS server kudos OK`.

automount: Mounts Filesystems Automatically

With distributed computing you can log in on any machine on the network, and all your files, including startup scripts, will be easily available. A distributed computing environment commonly has all machines able to mount all filesystems on all servers: Whichever machine you log in on, your home directory will be waiting for you.

Having all machines mount all servers all the time can be problematic. Suppose that machine A mounts some filesystems from machine B and machine B mounts some from machine A. What happens when you bring one of these machines down for maintenance or it crashes? In what order do you reboot them when they depend on each other to be up? In a large network you can have one machine mounting disks from tens or hundreds of others for software files and home directories.

One way around this problem is to mount filesystems only on demand. On GNU/Linux machines demand mounting is handled by the autofs system (using the **automount** daemon), which is replacing the older, less efficient **amd** (automounting daemon). Because autofs runs in kernel space (**amd** runs in user space), you need to have support for it in the kernel (`Filesystems/Kernel automounter support`). For example, when you issue the command **ls /home/alex**, autofs goes to work: It looks in the **/etc/auto.home** map, finds that **alex** is a key that says to mount **franklin:/export/homes/alex**, and mounts the remote filesystem.

Once the filesystem is mounted, ls displays the list of files you want to see. If after this mounting sequence you give the command **ls /home**, ls shows that **alex** is present within the **/home** directory. The df utility shows that **alex** is mounted from **franklin**. By default the **automount** daemon automatically unmounts this filesystem after five minutes of inactivity.

Automounting filesystems is similar in concept to MS Windows 9x network neighborhood. When you know there are NFS servers named **franklin**, **adams**, and **madison**, you can see all the filesystems that are exported by each by using ls to display (for example) **/net/franklin**, **/net/adams**, and **/net/madison**. Once these filesystems are mounted, you can browse through them if you have permission.

The GNU/Linux automount facility is flexible and powerful. Refer to "autofs: Automatically Mounts Filesystems" on page 979 and the automount man page for more information.

Optional

Internet Services

GNU/Linux Internet services are provided by daemons that run continuously or by a daemon that is started automatically by the **xinetd** daemon (page 397) when a service request comes in. The **/etc/services** file lists network services (for example **telnet**, **ftp**, **ssh**) and their associated numbers. Any service that uses TCP/IP or UDP/IP uses an entry in this file. IANA (Internet Assigned Numbers Authority) maintains a database of all permanent, registered services. The **/etc/services** file usually lists a small, commonly used subset of services. Go to www.rfc.net/rfc1700.html for more information and a complete list of registered services.

Most of the daemons (the executable files) are stored in **/usr/sbin**. By convention the names of many daemons end with the letter **d** to distinguish them from utilities.[20] The prefix **in.** or **rpc.** is often used for daemon names. Look at **/usr/sbin/*d** to see a list of many of the daemon programs on your system. Refer to "rc Scripts: Start and Stop System Services" on page 944 and service: Configures Services I on page 945 for information about starting and stopping these daemons.

For example, when you run ssh, your local system contacts the ssh daemon (**sshd**) on the remote system to establish the connection. The two systems negotiate the connection according to a fixed protocol. Each system identifies itself to the other, and then they take turns asking each other specific questions and waiting for valid replies. Each network service follows its own protocol.

In addition to the daemons that support the utilities described up to this point, many other daemons support system-level network services that you will not typically interact with. Some of these daemons are listed in Table 9-4.

table 9-4

Daemon	Used For or By	Function
apmd	Advanced power management	Reports and takes action on specified changes in system power, including shutdowns. Very useful with machines, such as laptops, that run on batteries.
atd	at	Executes a command once at a specific time and date. See crond for periodic execution of a command.
automount	Automatic mounting	Automatically mounts filesystems when they are accessed. Automatic mounting is a way of demand-mounting remote directories without having to hard-configure them into /etc/fstab.

20. One common daemon whose name does not end in d is **sendmail**.

Daemon	Used For or By	Function (Continued) || table 9-4
comsat	Notifies users of new mail	Used by biff, a utility that notifies users of incoming mail. If the user is logged on and has run biff y, comsat sends a message to the user's shell, saying that there is new mail (at an appropriate time). Security-conscious sites may want to disable this service, as it has a history of security holes. Launched by xinetd.
crond	cron	Used for periodic execution of tasks, this daemon looks in the /var/spool/cron/ directory for files that have filenames that correspond to users' login names. It also looks at the /etc/crontab file and at files in the /etc/cron.d directory. When a task comes up for execution, crond executes it as the user who owns the file that describes the task.
dhcpcd	DHCP	Client daemon. Refer to "DHCP Client" on page 1028.
dhcpd	DHCP	Assigns Internet address, subnet mask, default gateway, DNS, and other information to hosts. This protocol answers DHCP requests and, optionally, BOOTP requests. See *DHCP* on page 1465.
fingerd	finger	Handles requests for user information from the finger utility. Launched by xinetd.
ftpd	FTP	Handles FTP requests. Refer to "ftp: Transfers Files over a Network" on page 378. Launched by xinetd.
gpm	General-purpose mouse or GNU paste manager	Allows you to use a mouse to cut and paste text on console applications.
httpd	HTTP	A Web server daemon. See *HTTP* on page 1472.
inetd		Deprecated in favor of xinetd.
lpd	line printer spooler daemon	Launched by xinetd when printing requests come to the machine.
named	DNS	Supports *DNS* (page 1465), which has replaced the use of the /etc/hosts table for hostname-to-IP address mapping on most networked UNIX/Linux systems.
nfsd, statd, lockd, mountd, rquotad	NFS	These five daemons operate together to handle *NFS* (page 1482) operations. The nfsd daemon handles file and directory requests. The statd and lockd daemons implement network file and record locking. The mountd daemon takes care of converting a filesystem name request from the mount utility into an NFS handle and checks access permissions. Finally, if disk quotas are enabled, rquotad handles those.

Daemon	Used For or By	Function (Continued) || table 9-4
ntpd	NTP	Synchronizes time on network computers. Requires a /etc/ntp.conf file. For more information go to www.eecis.udel.edu/~mills/ntp/servers.htm and www.eecis.udel.edu/~ntp.
portmap	RPC	Maps incoming requests for RPC service numbers to a TCP or UDP port numbers on the local machine. Refer to "RPC Network Services" on page 398.
pppd	PPP	For a modem this protocol controls the pseudointerface represented by the IP connection between your computer and a remote computer. Refer to "PPP: Point-to-Point Protocol" on page 362.
rexecd	rexec	Allows a remote user with a valid username and password to run programs on a machine. Its use is generally deprecated because of security, but certain programs, such as PC-based X servers, may still have it as an option. Launched by xinetd.
routed	Routing tables	Manages the routing tables so that your system knows where to send messages that are destined for remote networks. If your system does not have a /etc/defaultrouter file, routed is started automatically to listen to incoming routing messages and to advertise outgoing routes to other systems on your network. A newer daemon, the Gateway daemon (gated), offers enhanced configurability and support for more routing protocols and is proportionally more complex.
sendmail	Mail programs	The sendmail daemon came from Berkeley and has been available for a long time. The de facto mail transfer program on the Internet, the sendmail daemon always listens on port 25 for incoming mail connections and then calls a local delivery agent, such as /bin/mail. Mail user agents, such as pine and Mozilla mail, typically use sendmail to deliver mail messages.
smbd, nmbd	Samba	Allow MS Windows PCs to share files and printers with UNIX/Linux computers.
sshd	ssh, scp	Enables secure logins between remote machines (page 374).
syslogd	System log	Transcribes important system events and stores them in files and/or forwards them to users or another host running the syslogd daemon. This daemon is configured with /etc/syslog.conf and used with the syslog utility.
talkd	talk	Allows you to have a conversation with another user on the same or a remote machine. The talkd daemon handles the connections between the machines. The talk utility on each machine contacts the talkd daemon on the other machine for a bidirectional conversation. Launched by xinetd.

Daemon	Used For or By	Function (Continued) || table 9-4
telnetd	TELNET	One of the original Internet remote access protocols (page 376). Launched by xinetd.
tftpd	TFTP	Used to boot a system or get information from a network. Examples include network computers, routers, and some printers. Launched by xinetd.
timed	Time server	On a LAN synchronizes time with other computers that are also running timed.
xinetd	Internet *Superserver*	Listens for service requests on network connections and starts up the appropriate daemon to respond to any particular request. Because of xinetd, your system does not need to have all the daemons running all the time in order to handle various network requests. The configuration file for xinetd is /etc/xinetd.conf, which frequently includes all the files in the /etc/xinetd.d directory with the line `includedir /etc/xinetd.d` Each of the files in xinetd.d is named after a service that it controls. Each file contains a line that starts with `disable` = and finishes with `yes` or `no`. This line determines whether the service can run.

Proxy Server

A *proxy* is a network service that is authorized to act for a system while not being part of that system. A proxy server or proxy gateway provides proxy services; it is a transparent intermediary, relaying communications back and forth between an application, such as a browser and a server, usually outside of your LAN and frequently on the Internet. When more than one process uses the proxy gateway/server, it must keep track of which processes are connecting to which hosts/servers so that it can route the return messages to the proper process. The most common proxies that a user encounters are e-mail and Web proxies.

A proxy server/gateway insulates the local computer from all other computers or from specified domains by using at least two IP addresses: one to communicate with your local computer and one to communicate with a server. The proxy server/gateway examines and changes the header information on all packets it handles so that it can encode, route, and decode them properly. The difference between a proxy gateway and a proxy server is that the proxy server usually includes *cache* (page 1459) to store frequently used Web pages so that the next request for that

page is available locally and quickly whereas a proxy gateway usually does not use cache. The terms proxy server and proxy gateway are frequently interchanged.

Proxy servers/gateways are available for such common Internet services as HTTP, HTTPS, FTP, SMTP, and SNMP. When an HTTP proxy sends queries from local machines, it presents a single organization-wide IP address (the external IP address of the proxy server/gateway) to all servers. It funnels all user requests to servers and keeps track of them. When the responses come back, it fans them out to the appropriate applications, using each machine's unique IP address, protecting local addresses from remote/specified servers. Proxy servers/gateways are generally just one part of an overall firewall strategy to prevent intruders from stealing information or damaging an internal network. Other functions, which can be combined with or be separate from the proxy server/gateway, are packet filtering, which blocks traffic based on origin and type, and user activity reporting, which helps management learn how the Internet is being used.

Refer to "Proxies" on page 1016 for practical information on setting up a proxy.

RPC Network Services

An RPC (remote procedure call) is a call to a *procedure* (page 1486) that acts transparently across a network. The procedure itself is responsible for accessing and using the network. The RPC libraries make sure that network access is transparent to the application. RPC runs on top of TCP/IP or UDP/IP.

The **/etc/rpc** file lists servers for RPCs.[21] This file has three columns: the name of the server for the RPC program, the RPC program number, and the names of programs that use the RPC program.

When an RPC server is initialized, it picks an arbitrary *port* (page 1485) that it communicates over. The server then registers this port with the RPC portmapper on the same machine, using the portmap utility. The portmap utility always listens on port 111 for both TCP and UDP.

When it wishes to execute an RPC against an RPC server, a client contacts portmap on the remote machine and asks which port the RPC server (for example **rpc.rstatd**) is listening on. The portmapper looks in its tables and returns a UDP or TCP port number. The client then contacts the server on that port.

The client sends arguments, just as a local function call or procedure would; the RPC libraries take care of transmission; the remote procedure executes with the arguments and generates a result; the RPC libraries encode the result and return it over the network to the client.

21. These are Sun-style RPCs, ONC or Open Network Computing RPC, as opposed to Microsoft RPCs, which are something different.

Usenet

One of the earliest information services available on the Internet, Usenet is an electronic bulletin board that allows users with common interests to exchange information. Usenet is an informal, loosely connected network of systems that exchange e-mail and news items (commonly referred to as *netnews*). Usenet was formed in 1979 when a few sites decided to share some software and information on topics of common interest. They agreed to contact one another and to pass the information along over dial-up telephone lines (at that time running at 1200 baud at best), using UNIX's uucp utility (UNIX-to-UNIX copy program).

The popularity of Usenet led to major changes in uucp to handle the ever-escalating volume of messages and sites. Today much of the news flows over network links using a sophisticated protocol designed especially for this purpose: NNTP (Network News Transfer Protocol). The news messages are stored in a standard format, and the many public domain programs available let you read them. An old, simple interface is named readnews. Others, such as rn, its X Window System cousin xrn, tin, nn, and xvnews have many features that help you browse through and reply to the articles that are available or create articles of your own. In addition, Netscape and Mozilla include an interface that you can use to read news (Netscape/Mozilla News) as part of its Web browser. The program you select to read netnews is largely a matter of personal taste.

Because programs to read netnews articles have been ported to non-UNIX/Linux systems, the community of netnews users has diversified. In the UNIX tradition categories of netnews groups are structured hierarchically. The top level includes such designations as **comp** (computer-related), **misc** (miscellaneous), **rec** (recreation), **sci** (science), **soc** (social issues), and **talk** (ongoing discussions). Usually at least one regional category is at the top level, such as **ba** (San Francisco Bay Area), and includes information about local events. Many new categories are continually being added to the more than 30,000 newsgroups. The names of newsgroups resemble domain names but are read from left to right (like GNU/Linux filenames): **comp.os.UNIX.misc**, **comp.lang.c**, **misc.jobs.offered**, **rec.skiing**, **sci.med**, **soc.singles**, **talk.politics**. The following article appeared in **linux.redhat.install**:

```
> I have just installed linux redhat 7.2 and when i try to start X i get the
> following error message:
>
> Fatal Server Error.
> no screens found
>
> XIO: Fatal IO err 104 (connection reset by peer) on X server ",0.0" after
> 0 requests (0 known processed) with 0 events remaining.
>
> How can i solve this problem?
>
> Thanks,
> Fred
```

```
Fred,

It would appear that your X configuration is incorrect or missing.  You
should run XConfigurator and set up the configuration for your video card
and monitor.  You may also have to run mouseconfig to set it up.

Carl
```

A great deal of useful information is available on Usenet, but you need patience and perseverance to find what you are looking for. You can ask a question, as the user did in the previous example, and someone from halfway around the world may answer it. Before posing such a simple question and causing it to appear on thousands of systems around the world, ask yourself whether you can get help in a less invasive way.

- Refer to the man pages and info.
- Look through the files in **/usr/share/doc**.
- Ask your system administrator or another user for help.
- All the popular newsgroups have FAQs (lists of frequently asked questions). Consult these lists and see whether your question has been answered. FAQs are periodically posted to the newsgroups; in addition, all the FAQs are archived at sites around the Internet, including ftp://ftp.uu.net, ftp://rtfm.mit.edu/pub/usenet-by-hierarchy,[22] and the Usenet newsgroup **comp.answers**.
- Because someone has probably asked the same question before you, search the netnews archives for an answer: Try looking at groups.google.com, which has a complete netnews archive.
- Use a search engine to find an answer. One good way to get help is to search on an error message.
- Review support documents at **www.redhat.com**.
- Contact a Red Hat Linux user's group.

Use the worldwide Usenet community as a last resort. If you are stuck on a GNU/Linux question and cannot find any other help, try submitting it to one of these newsgroups:

22. Also see ftp://rtfm.mit.edu/pub for other Usenet archives and miscellaneous interesting information.

- **linux.redhat.development**
- **linux.redhat.install**
- **linux.redhat.misc**

For more generic questions try these lists:

- **comp.os.linux.misc**
- **comp.os.linux.networking**
- **comp.os.linux.security**
- **comp.os.linux.setup**
- **linux.dev.newbie**
- **linux.redhat.rpm**

One way to find out about new tools and services is to read Usenet news. The **comp.os.linux** hierarchy is of particular interest to GNU/Linux users; for example, news about newly released software for GNU/Linux is posted to **comp.os.linux.announce**. People often announce the availability of free software there, along with instructions on how to get a copy for your own use using anonymous FTP (page 380). Other tools to help you find resources, both old and new, exist on the network; see Appendix B.

Tutorial: Using pine as a Newsreader

The pine news interface resembles the pine mail interface (page 88), with as much consistency among commands, screen displays, and folder organization as possible. This consistency makes it easier for those used to the pine mailer to use pine as a newsreader. If you are using Mozilla, you may prefer to use Mozilla News (page 405).

In order to use any newsreader, you must have access to Usenet news. Ask your ISP or system administrator for the address of your news server. If your site has no news server, you will not be able to read news.

Start pine and select SETUP from the MAIN MENU. Enter **c** (Config) on the initial SETUP screen to display the SETUP CONFIGURATION screen, where you can view and modify many configurable aspects of pine's behavior (Figure 9-4). Highlight the pine variable **nntp-server**, select **a** (Add Value), and enter the URL of your news server.

In most cases this is all you need to do to start using pine as a newsreader. The next time you run pine, it will contact the news server on your behalf as you give commands to read and post news.

Figure 9-4 pine's Setup/Configuration screen

If you use pine for mail, you are probably accustomed to seeing the FOLDER LIST screen containing the mail folders INBOX, sent-mail, and saved-messages, and any other mail folders you have created. If this group, or *collection,* of folders is the only one defined, pine displays the individual folders within the collection when you select FOLDER LIST. Once you set up news, an additional collection—News—is automatically defined, and the FOLDER LIST screen becomes the COLLECTION LIST screen and displays a list of the two folder collections instead: Mail and News. Highlight the line for the News and press RETURN to use the news feature (Figure 9-5). The pine program displays the FOLDER LIST for news groups.

The news groups FOLDER LIST screen displays folders, each corresponding to a single newsgroup that you have subscribed to. To start, no folders are displayed. You can move among the newsgroups, or folders, by using the ARROW keys; select any newsgroup by pressing RETURN.

Figure 9-5 pine's Folder List screen

Figure 9-6 Partial results of a query for newsgroups containing `Linux` in their titles

Subscribing to Newsgroups

The *news subscription file,* **.newsrc** in your home directory, is used by pine to keep track of your news subscriptions. It is in a standard format that can be used by most newsreaders. You may find this file in your home directory, initialized with a list of newsgroups deemed to be of general interest. If not, pine will create the file for you the first time you subscribe to a newsgroup.

A newsgroup that might be useful at this point is **comp.mail.pine**. To subscribe to this newsgroup, while still in the FOLDER LIST for news groups screen, enter **a** (Add) followed by **comp.mail.pine** at the prompt and press RETURN. After a while pine displays `Subscribed to "comp.mail.pine"` on the message line and adds the new newsgroup to the list of newsgroups displayed on the FOLDER LIST screen. Each step involved in subscribing to a news group may take a while, depending on your environment.

To display all the newsgroups that include `Linux` in their titles, enter **linux** in response to the prompt you get when you give an **a** (Add) command. (Figure 9-6 shows partial results of this query.) Press RETURN immediately after you give an **a** command to display a list of all newsgroups (the list is long and takes a while to display).

Reading News

Use the ARROW keys to highlight one of the newsgroups that is displayed on the FOLDER LIST screen, and then press RETURN to display the `Message Index` screen. If you see the message `Group now downloading, please check back later`, press < to go

back to the FOLDER LIST screen, and return to the MESSAGE INDEX screen for that newsgroup later.

Most aspects of reading mail apply to reading news: When you view the MESSAGE INDEX screen, you see a numbered list of messages (*posts*) identified with dates, sender names, and subject lines. Highlight the line that interests you, and press RETURN to view the message.

Using pine, you can select messages that interest you, mark messages for deletion, export messages to files, and so on. When viewing a news message, you will see headers that resemble the headers used in pine mail messages. The fields `Date:`, `From:`, and `Subject:` appear in the four-line header, with similar meanings. To emphasize that recipients of news messages are newsgroups, the `To:` field is replaced with the `Newsgroups:` field. This field lists one or more newsgroups that are receiving the post.

Unlike other newsreaders, pine does not automatically delete the news messages that you have read; you must explicitly mark news messages for deletion, using the **d** command, as you do for pine mail messages. Because pine remembers which messages you have deleted between pine sessions, you can pick up where you left off the next time you run pine to read news.

Posting News

The commands to post news in pine are nearly identical to those to send pine mail. The main difference is that the list of recipients comprises newsgroup names, not the addresses of individual users. As with mail messages, news messages can be sent to multiple recipients.

Use R With Care

|| caution

When you enter r (Reply) to reply to a news post, you will be asked whether you want to include the original message in your reply; answer as you wish. Next, you will be given the following choices:

- Follow-up to news groups (F)
- Reply via email to author (R)
- Both (B)

Enter F or B only with the greatest caution; your message will reach thousands of people. Unless you want your message to go to *all* the subscribers of *all* the newsgroups listed in the `Newsgroups:` field of the header, enter R at this prompt. R causes your reply to be sent as an e-mail message only to the individual who posted the original message.

Unsubscribing from a Newsgroup

When you decide that you do not wish to belong to a newsgroup, you can unsubscribe from it. You will probably want to unsubscribe from many newsgroups if your **.newsrc** file was initialized for you; the list of such newsgroups is likely to be long and diverse.

To unsubscribe from a newsgroup, select News from the COLLECTION LIST screen, press RETURN to display the news group FOLDER LIST screen, highlight the newsgroup you want to unsubscribe from, and press **d** (Delete). Unsubscribing from a newsgroup does not remove the newsgroup from the **.newsrc** file; it simply tells pine not to include that newsgroup in the FOLDER LIST display. If you decide to subscribe to the same newsgroup again, pine remembers what messages you deleted, and you can resume reading the posts where you left off.

Netnews with Mozilla

You can also use Mozilla to read netnews. Use the Account Wizard to set up Mozilla to read and post netnews. How you display the Account Wizard depends on whether you already have a mail and/or newsgroup account set up with Mozilla. In either case select Mozilla menubar: Tasks⇨Mail & Newsgroups. If Mozilla displays the Account Wizard, you are set. When Mozilla displays the message center window,[23] select message center menubar: Edit⇨Mail & Newsgroups Account Settings to see the Mail & Newsgroups Account Settings window. Click Add Account and Mozilla displays the Account Wizard window.

From the Account Wizard window click Newsgroup account and then Next. Follow the prompts in the next windows, verify the information you entered on the final window, and click Finish to close the Account Wizard window. Click OK to close the Account Settings window.

If the message center window is not already displayed, select Mozilla menubar: Tasks⇨Mail & Newsgroups to display it. The news folder is generally at the bottom of the tree list on the left, labeled with the name of your news server. To add newsgroups, right click the news folder and click Subscribe from the pop-up menu. To see individual newsgroups, expand a news directory by clicking the plus sign (+) to the left of the news directory's name. Enter or pick the groups you want to subscribe to by double-clicking the name of the newsgroup. Click OK when you are done subscribing to newsgroups. Mozilla returns you to the message center window. Double-click the newsgroup you want to visit, and Mozilla displays items from the newsgroup.

23. The message center window displays the name of the highlighted folder on its titlebar.

The upper portion of the right side of the window lists the postings (the newsgroup items), including the name of the sender and the subject. Click a posting that interests you; after a moment the item appears in the lower portion of the screen. You can read and reply to postings from this window. For more information on Mozilla News, refer to the online Mozilla documentation by selecting Help at the right end of the menubar.

WWW: World Wide Web

The World Wide Web (WWW, W3, or the Web) provides a unified, interconnected interface to the vast amount of information stored on computers around the world. The idea that created the World Wide Web came from the mind of Tim Berners-Lee of the European Particle Physics Laboratory (CERN) in response to a need to improve communications throughout the High Energy Physics community. The first generation was a notebook program named Enquire, short for "Enquire Within Upon Everything" (the name of a book from his childhood), that he created in 1980 and that provided for links to be made between named nodes. It was not until 1989 that the concept was proposed as a global hypertext project to be known as the World Wide Web. In 1990 Berners-Lee wrote a proposal for a HyperText project, which eventually produced HTML, HyperText Markup Language, the common language of the Web. The World Wide Web program became available on the Internet in the summer of 1991. By designing the tools to work with existing protocols, such as FTP and gopher, the researchers who created the Web created a system that is generally useful for many types of information and across various types of hardware and operating systems.

The WWW is another example of the client/server paradigm. You use a WWW client application, or *browser,* to retrieve/display information stored on a server that may be located anywhere on your local network or the Internet. WWW clients can interact with many types of servers; for example, you can use a WWW client to contact a remote FTP server (page 408) and display the list of files it offers for anonymous FTP (page 380). Most commonly you use a WWW client to contact a WWW server, which offers support for the special features of the World Wide Web that are described in the remainder of this chapter.

The power of the Web is in its use of *hypertext,* a way to navigate through information by following cross-references (called *links*) from one piece of information to another. To use the Web effectively, you need to be able to run interactive network applications. The first GUI for browsing the Web was a tool named Mosaic, released in February 1993. It was designed at the National Center for Supercomputer Applications at the University of Illinois and sparked a dramatic increase in

the number of users of the World Wide Web. Marc Andreessen, who participated in the Mosaic project at the University of Illinois, later cofounded Netscape Communications with the founder of Silicon Graphics, Jim Clark. They created Netscape Navigator, a Web client program that was designed to perform better and support more features than the Mosaic browser. Netscape Navigator has enjoyed immense success and has become a popular choice for users exploring the World Wide Web. Important for GNU/Linux users is fact that from the beginning, Netscape has provided versions of its tools that run on GNU/Linux. Also, Netscape created Mozilla (mozilla.org) as an open-source browser project.

Mozilla and the Netscape Navigator[24] provide GUIs that allow you to listen to sounds, watch Web events or live news reports, and display pictures as well as text, giving you access to *hypermedia.* A picture on your screen may be a link to more detailed, nonverbal information, such as a copy of the same picture at a higher resolution or a short animation. When you run Mozilla or Netscape on a system that is equipped for audio, you can to listen to audio clips that have been linked to from a document.

URL: Uniform Resource Locator

Consider the URL http://www.w3.org/pub/WWW. The first component in the URL indicates the type of resource, in this case, **http** (HTTP—HyperText Transfer Protocol). Other valid resource names, such as **https** (HTTPS—secure HTTP), and **ftp** (FTP—File Transfer Protocol), represent information available on the Web, using other protocols. Next comes a colon and double slash (**://**). Frequently the `http://` string is omitted from a URL in print, as you seldom need to enter them to get to the URL. Following this is the full name of the host that acts as the server for the information (**www.w3.org**). The rest of the URL is a relative pathname to the file that contains the information (**pub/WWW**). Enter a URL in the location bar text box of a Web browser, and the Web server returns the page, frequently an *HTML* (page 1472) file, pointed to by this URL.

By convention many sites identify their WWW servers by prefixing a host or domain name with **www**. For example, you can reach the Web server at the New Jersey Institute of Technology at www.njit.edu. When you use a browser to explore the World Wide Web, you may never need to use a URL directly. However, as more information is published in hypertext form, you cannot help but find URLs everywhere—not just online in mail messages and Usenet articles but also in newspapers, advertisements, and product labels.

24. Netscape runs only on a GUI. If you are working on a character-based terminal or emulator, use lynx or links to access the Internet.

Browsers

You might want to consider using Web browsers other than Netscape with your GNU/Linux system. If you do not use the X Window System, try a text browser, such as lynx or links. Mozilla (www.mozilla.org) is the open-source counterpart to Netscape. Mozilla was first released in March 1998 and was based on Netscape 4 code. Since that time Mozilla has been under development by employees of Netscape (now a division of AOL), Red Hat, other companies, and contributors from the community and has released its version 1.0. KDE offers Konqueror, an all-purpose file manager and Web browser (page 286). Other browsers include Galeon (galeon.sourceforge.net), Opera (www.opera.com), BrowseX (browsex.com), and SkipStone (muhri.net/skipstone). Although each Web browser is unique, they all allow you to move about the Internet, viewing HTML documents, listening to sounds, and retrieving files.

Search Engine

Search engine is a name that applies to a group of hardware and software tools that help you find World Wide Web sites that have the specific information you are looking for. A search engine relies on a database of information collected by a *Web crawler*, a program that regularly looks through the millions of pages that make up the World Wide Web. A search engine must also have a way of collating the information the Web crawler collects so that you can access it quickly, easily, and in a manner that makes it most useful to you. This part of the search engine, called an *index*, allows you to search for a word, a group of words, or a concept and returns the URLs of Web pages that pertain to what you are searching for.

Many different types of search engines are on the Internet. Each type of search engine has its own set of strengths and weaknesses. You can obtain a partial list of search engines by going to home.netscape.com/escapes/internet_search.html or by clicking the Search button on the Netscape or Mozilla menubar.

Downloading a File

You can use Mozilla, Netscape, or another browser to look at and download a file from an FTP or HTML site. Suppose you enter ftp://ibiblio.org/Linux in the text box of the location bar and press RETURN. After seeing the initial set of directories, click **pub** (many sites give their public directory this name). You can then click any of the directories (try **Linux**) to view the available files. Following this example you will find directories named with the classifications of software, documentation, distributions, and more. Each contains a wealth of directories with more directories

and files that you can download. You will also find **html** files that display a graphical interface to the directories. When you click a file that is intended to be downloaded, Mozilla or Netscape opens a window asking you where to put the file on your system. Refer to "Installing and Removing Software" on page 926 for information about unpacking and installing the software that you download.

When a File is Downloaded to Your Screen (and You See Garbage) || tip

If garbage appears on your screen, the file is being downloaded to your screen. Click Stop and then Back: You should be back where you started. This time hold the SHIFT key down while you click the file you want: This tells Mozilla/Netscape to download the file instead of trying to display it.

Chapter Summary

A GNU/Linux system attached to a network is probably communicating on an Ethernet, which may be linked to other local area networks (LANs) and wide area networks (WANs). Communication between LANs and WANs requires the use of gateways and routers. Gateways translate the local data to a format suitable for the wide area network, and routers make decisions about optimal routing of the data along the way. The most widely used network, by far, is the Internet.

Basic networking tools allow GNU/Linux users to log in and run commands on remote systems (ssh, telnet) and copy files quickly from one system to another (scp, ftp/sftp). Many tools that were originally designed to support communication on a single-host computer (for example, finger, talk, pine) have been extended to recognize network addresses, thus allowing users on different systems to interact with one another. Other features, such as the Network Filesystem (NFS), were created to extend the basic UNIX model and to simplify information sharing.

Concern is growing for the security and privacy of machines connected to networks and of data transmitted over networks. Toward this end many new tools and protocols have been created: ssh, scp, HTTPS, IPv6, firewall hardware and software, VPN, and so on. Many of these tools take advantage of newer, more impenetrable encryption techniques. In addition, some concepts, such as that of trusted hosts, and some tools, such as finger and rwho, are being discarded in the name of security.

Two major advantages of computer networks over other ways of connecting computers are that they enable systems to communicate at high speeds and require few physical interconnections (typically one per system, often on a shared cable). The Internet Protocol (IP), the universal language of the Internet, has made it possible for dissimilar computer systems around the world to communicate easily with

one another. Technological advances continue to improve the performance of computer systems and the networks that link them.

One way to gather information on the Internet is Usenet news (netnews). Many GNU/Linux users routinely read Usenet news to learn about the latest resources available for their systems. Usenet news is organized into newsgroups that cover a wide range of topics, computer-related and otherwise. To read Usenet news, you need to have access to a news server and the appropriate client software. Many modern mailers, such as pine, Mozilla, and Netscape, are capable of reading netnews.

The rapid increase of network communication speeds in recent years has encouraged the development of many new applications and services. The World Wide Web provides access to vast information stores on the Internet and is noted for its extensive use of hypertext links to promote efficient searching through related documents. The World Wide Web adheres to the client/server model so pervasive in networking; typically the WWW client is local to a site or is made available through an Internet service provider. WWW servers are responsible for providing the information requested by their many clients.

Netscape Navigator is a WWW client program that has enormous popular appeal. Netscape and Mozilla use a GUI to give you access to text, picture, and audio information: Making extensive use of these hypermedia simplifies access to and enhances the presentation of information.

Exercises

1. Describe the similarities and differences among these utilities:

   a. scp and ftp

   b. ssh and telnet

   c. rsh and ssh

2. Assuming that rwho is disabled on the systems on your LAN, describe two ways to find out who is logged in on some of the other machines attached to your network.
3. Explain the client/server model, and give three examples of services that take advantage of this model on GNU/Linux systems.
4. What is the difference between a diskless and a dataless workstation? Name some advantages and disadvantages of each.

5. A software implementation of chess was developed by GNU and is free software. How can you use the Internet to find a copy and download it to your system?
6. What is the difference between the World Wide Web and the Internet?
7. If you have access to the World Wide Web, answer the following:
   a. What browser do you use?
   b. What is the URL of the author of this book's home page? How many links does it have?
   c. Does your browser allow you to create bookmarks? If so, how do you create a bookmark? How can you delete one?
8. Explain what happens if you transfer a binary file while running ftp in ASCII mode. What happens if you transfer an ASCII file in binary mode?
9. Give one advantage and two disadvantages of using a wireless network.

Advanced Exercises

10. Suppose that the link between routers 1 and 2 is down in the Internet shown in Figure 9-1 on page 359. What happens if someone at Site C sends a message to a user on a workstation attached to the Ethernet cable at Site A? What happens if the router at Site A is down? What does this tell you about designing network configurations?
11. If you have a class B network and want to divide it into subnets, each with 126 hosts, what subnet mask should you use? How many networks will be available? What are the four addresses (broadcast and network number) for the network starting at 131.204.18?
12. Suppose that you have 300 hosts and want to have no more than about 50 hosts per subnet. What size address block should you request from your ISP? How many class C-equivalent addresses would you need? How many subnets would you have left over from your allocation?
13. On your machine find two daemons running that are not listed in this chapter, and explain what purpose they serve.
14. Review what services/daemons are automatically started on your system, and consider which you might turn off. Are there any services/daemons in the list that starts on page 394 that you would consider adding?