CHAPTER 1

Internet Fundamentals

OBJECTIVES

1. Appreciate the scope of the Internet's impact on our everyday life.
2. Learn about the different components that make the Internet work.
3. Understand how standards, services, and protocols allow disparate hardware and software to work together.
4. Have a basic understanding of Internet architecture and addressing.
5. Be able to understand how the World Wide Web, Web browsers, and Web servers work.

INTRODUCTION

The **Internet** had its origins in American research and defense work of the 1970s and 1980s. The rise of the Internet in the 1990s once the system was released into the public domain will come to be seen as one of the great technological and sociological achievements of the modern era. In time it may have as profound an effect on civilization as the telephone had on communications, and perhaps may even rival the impact Guttenberg's 1450s invention of the moveable type printing press had on literacy. The Internet is still too young to fully gauge its impact. In 2003 it is estimated that more than 500 million people out of the 7 billion people living on this planet have access to the Internet worldwide, about 7 percent. The numbers are growing at an astonishing clip as entire countries' social systems are converted from paper to Web access.

When you are connected to the Internet, you are accessing the world's

- Largest post office.
- Greatest library.
- Biggest store.

- Most powerful data transmission network.
- The preferred method for computers of one type to talk with computers of any other type.

In the very near future, you will be able to add to this list

- The world's largest classroom (or greatest university) through remote learning capabilities.
- The world's most powerful newspaper, magazine, TV, and radio carrier.
- The main portal (or view port) to your government, health care, and most service organizations.
- The transmission network of choice (indeed!) for nearly all applications written to use networked services.

It's easy to use superlatives when talking about the Internet, but it is very hard to determine where the technology is going to take us. In an era when homes, refrigerators, cars, and even perhaps your own personal physical self will be addressable over the Internet, the possibilities are truly astonishing.

For all of these capabilities, there is the never-ending requirement for trained programmers to create and improve the applications of tomorrow. In the 21st century, perhaps only biotechnology offers as bright a future as computer programming for the Web does.

Our goal in this chapter isn't to present you with the glorious (albeit brief) history of the Internet. Here you will learn about some of the fundamental principles required to make the Internet work, how the Internet and Internet applications are structured, and the role of programmers in developing new applications and maintaining existing ones. This chapter should give you some insight into how your programs must work in order to succeed, and why different programming languages or development systems are used in different areas of Internet technology.

Infrastructure

The Internet is just what its name implies—a network of networks; or to be more precise, a wide area network, or WAN. Nobody actually owns the Internet, although there are companies that own parts of it. Two government agencies, the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF), funded the creation of the original Internet network for research and military use. There were two motivations for the work. First, they wanted to establish a network that could allow computers of all types to talk with one another, as well as a means to connect from a terminal (or, as it turns out, a personal computer) to the various types of connected

computers that projects relied on. The second motivation was to create a network that could survive a nuclear attack by being highly redundant.

The person most associated with conceptualizing the Internet's infrastructure and playing a pivotal role in its funding was DARPA's Director J. C. R. Licklider, who funded the initial work in 1963–64 and directed the group at Bolt, Beranek, and Newman in Cambridge, Massachusetts, that did the initial experiments in time sharing and networking. Prior to the fund DARPA gave out to universities, there wasn't a single university in the United States that even offered a Ph.D. in computer science.

The solution to both of these needs was to create a distributed network architecture that provided redundant pathways from one computer to another. A fabric topology was adopted, which is often referred to as the "Web." When one transmission point or pathway fails, data is routed to other paths—thus providing fault tolerance in case of system failures. Your brain uses a similar approach, a so-called neural network to create redundant and multiple pathways for the electrical transmission of information.

In order to make data redundant, the system uses what are called packets, bite-size chunks of data with headers that describe what the data belongs to, who's sending it, and who is the intended receiver. These packets are sent as a stream and will be rebroadcast by the source unless a signal is received from the recipient that the package was received. The recipient assembles the data from the collection of packets using the information they contain, and by validating the packets using error-checking algorithms. By adopting a packet-based approach, the Internet is able to successfully send large data files using a transmission medium that is intermittent and variable. Packets that don't arrive are retransmitted, perhaps traveling a different path, until the entire message can be assembled by the host. The "cloud" architecture of the Internet is illustrated in Figure 1.1.

The Internet can be broken down into the following fundamental parts:

- **Routers and switches.** **Routers,** such as the ones sold by Cisco (among others) are special-purpose computers that direct traffic in the form of data packets to their intended recipient. When one route is blocked or busy, routers choose another path using routing tables (both static and dynamic tables) and intelligent algorithms, thus ensuring that data traffic on the Internet is fault tolerant. You'll also find hubs or switches, bridges, and gateways on the Internet. *Hubs* or *switches* connect nodes without any routing or data translation abilities. *Bridges* are used to link different local area networks, or LANs, together. *Gateways* are a special type of bridge that also can do data translation—for example, turn a PC file into a Macintosh file.
- **Cabling backbone.** The fiber optic networks called *backbones* that carry much of the Internet traffic under our city streets or across the oceans

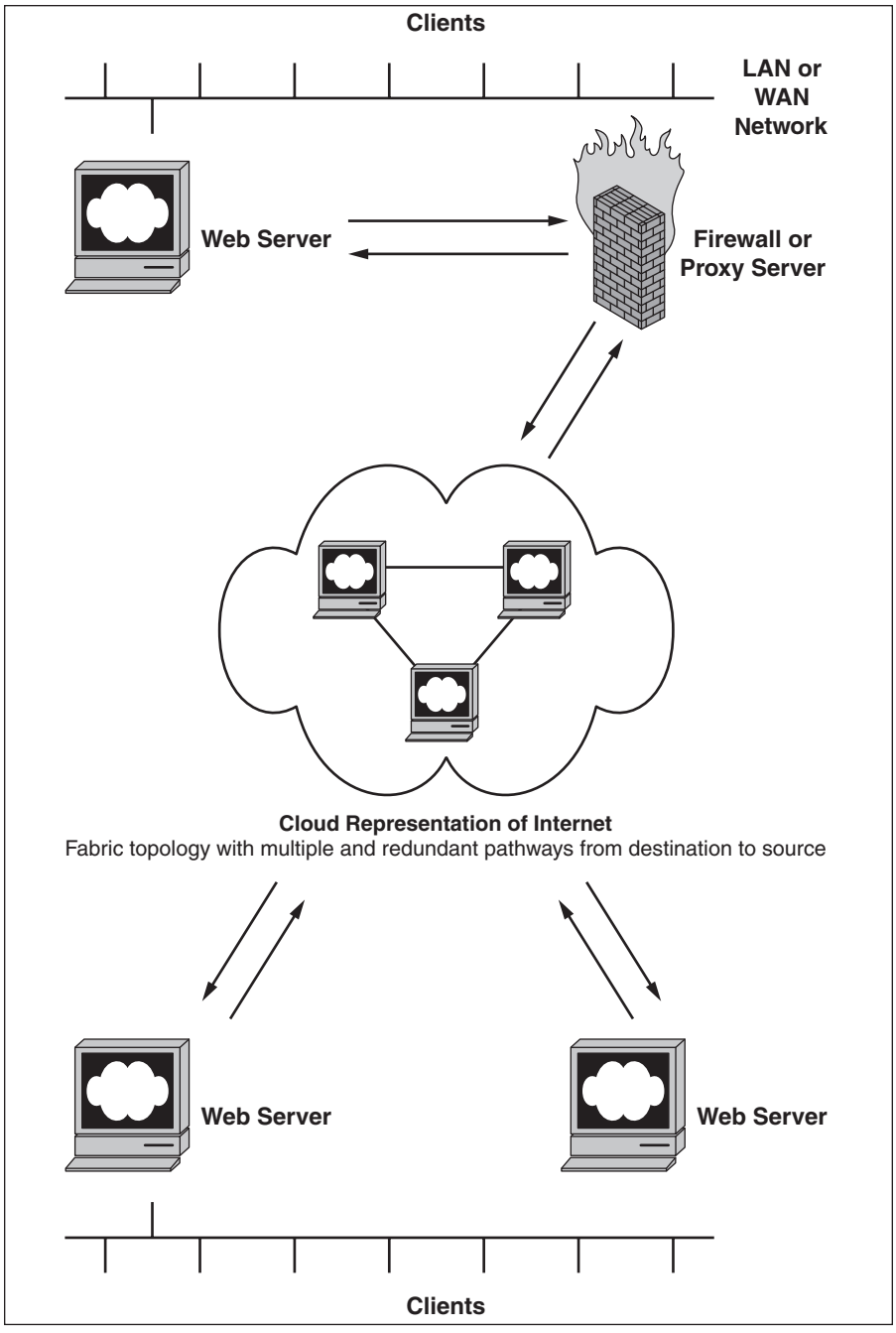

FIGURE 1.1 Cloud Presentation of Internet

The Internet uses a packet-switched fabric-type architecture that is highly redundant and fault tolerant. The multiple paths available from origin to destination make a specific path impossible to describe, so often the Internet infrastructure is portrayed as a "cloud."

and the satellites that circle our planet all contribute to the transmission of Internet traffic and are all owned by government agencies or large corporations.

A next generation version of the Internet backbone called *Internet2* (http://www.internet2.edu/) also has been funded and is in limited usage. It will link over 100 universities with a 2.4-gigabits-per-second network that is capable of, among other things, being able to transmit full-motion streaming video. Many people believe that the Internet2 will serve as the replacement for the current Internet.

- **Last mile connection.** The last mile connections are typically owned by phone and telecommunications companies, with hookups offered by *Internet service providers,* or ISPs (hosts). ISPs are companies that sell a monthly connection and related services to customers. AOL, MSN, Earthlink, and AT&T Worldnet are examples of service providers.
- **Name servers**. Several companies and government agencies own and operate the **name servers** that translate the Internet address (which for IPv4 are in the octet form ###.###.###.###) into friendly names like AOL.com or CNN.com, and vice versa. For example, CNN's Web site's address is 64.36.16.84. Each three-number set allows numbers up to 256. When you register a domain on the World Wide Web, you in a sense are renting an ownership interest in the Internet using a **registrar** that works with the InterNIC (a company that owns the registration database for .COM, .ORG, .NET, and others), which owns and administers the name servers. You are assigned an IP address, as well as a network type (A, B, C, or D), which determines how many nodes can be accommodated by your domain.

QuickCheck Questions

1. What happens when a router fails?
2. Who owns the Internet?

The Role of Standards

It's best to think of the Internet in terms of a set of standards or agreed-upon conventions. Those standards govern how nodes (points) on the Internet are addressed, what the data stream over the wires must look like, and how applications should behave in order to communicate with other applications. These standards are codified by working groups (most often international in scope) who accept proposals, examine their implications often by issuing requests for comments (RFCs) to which any interested party can respond, and draft and disseminate standards for people to follow in their work on the Internet. This sort of "peer" review process arises from the Internet's roots as a research tool for universities and the defense establishment.

The best known standards organizations are

1. The Internet Society (www.isoc.org), a nonprofit group that works on broad Internet use issues.
2. The Internet Engineering Task Force (IETF; www.ietf.org), which determines how different protocols or agreements are structured.
3. The World Wide Web Consortium (W3C; www.w3c.org), which develops the standards for Web applications and is based at MIT.

The principle behind a standards-based approach is to try and make disparate hardware, operating systems, and applications work together. Fundamental aspects of Internet behavior are nearly universally adopted. Thus, for example, all **browsers** (programs that can read HTML markup **tags** and compose a page based on the content) can read a simple text file with standard HTML tags like HEAD, BODY, and so forth. However, later submissions such as particular kinds of animation tags or certain kinds of Cascaded Style Sheets (a set of styles used in Web documents that were proposed by Microsoft) are always adopted by one browser or another. So what you see in Microsoft Internet Explorer won't necessarily be the same as what you see in Netscape Navigator.

All vendors observe what may be called the "Principle of Graceful Degradation." If a browser doesn't understand a particular tag or can't display the results of some external applet (small program or module), then the browser ignores that data or instruction and composes the page without it. At least that's how it's supposed to work, but, alas, browsers do crash sometimes from these incompatibilities.

Often there are specifications for what to do when a browser can't display something. When a browser is text based (as many of the early browsers on cell phones are now, and many of the early browsers on the Internet were), the ALT tag instructs the browser to display an alternative text instead of the graphics file that is referenced. We'll get further into this topic in the next chapter, but the important point to remember from this discussion is that you have to test your programs and code against different browsers to see that the functions you need are supported. It's an unfortunate fact, but many Web sites optimize for a particular browser or offer alternate sites. This practice is slowly going away as Microsoft Internet Explorer's market dominance is forcing other browser manufacturers to follow suit with features that that browser supports.

Tech Tip

Web Resources
A good place to start out reading more about the Web and its development is the three sites mentioned here. In addition, you will find a list of resources that describe the history of the Internet at http://www.isoc.org/internet/history/.

Tech Tip

Web Resources
Remember when programming for the Internet that you should always test your Web sites and programs for their actions in at least Internet Explorer and Netscape Navigator. It's even prudent to test how your features will appear in previous versions of the browser. Don't assume everyone is running the latest and greatest browser. If your audience is using other browsers (like HotJava, for example), make sure you test on those as well.

Services and Protocols

As already mentioned, the Internet uses a **packet-switching** architecture—and for good reason. If you are sending data over a connection that you can't count on being reliable, you need to send smaller chunks of data and have a means for determining that all of the data arrived correctly at the target from the source. When you send data traffic across the Internet, a **protocol** (or

agreed specification) called the **Transmission Control Protocol/Internet Protocol (TCP/IP)** is used. The Transmission Control Protocol (TCP) part forms the data into bite-size chunks called **packets** and the Internet Protocol (IP) part routes them to their intended address.

TCP specifies how large the packets should be (typically less than 1,500 characters), adds addressing information to the packet's header, and encapsulates the data in an envelope, along with the instructions necessary to have those packets be reassembled into the original data that was sent. Headers include sender and destination addresses, the Time-To-Live (TTL) that determines how long the packet should be kept, date and time of creation, and other factors. You see this kind of information in the headers of your email when they are displayed.

Error checking in the form of a checksum is included in each packet to determine if the packet arrived intact at its destination. It's assumed that some percentage of packets are going to be dropped or may be corrupted upon arrival, so the intended recipient will send a message back to the initiating source that a packet is missing; and when all the packets are assembled, the target will send a received signal to the source.

The software that receives and translates TCP/IP signals and commands is called a **socket,** or a TCP/IP stack. Sockets have a feature called **ports,** which are a specified numbered "channel" that a data transfer of a specific type may use to communicate. Ports are registered in a central database so that they may be widely used, leading to the so-called standard for each protocol called a "well known port." For HTTP that port is 80, and a full specification of a URL using only IP nomenclature (not friendly DNS names) often includes the port number as the last element (e.g., http://132.94.62.15:80). Port numbers can be reassigned or hidden. For example, Microsoft's Internet Security and Acceleration Server (ISA), which is a combination firewall and caching server, uses the 8080 port for its traffic, and does the appropriate translations and mappings.

Tech Tip

Well Known Ports
For a complete and current listing of well known port assignments, go to http://www.iana.org/assignments/port-numbers.

QuickCheck Questions

3. What is a packet?
4. What role does a socket play?

In many operating systems, TCP/IP is added as a network component and operates as a system service, but for high-performance network-attached filers, TCP/IP stacks are embedded in a custom silicon chip called an ASIC for faster performance. The Windows version of the socket is called Winsock, while for the Macintosh it is MacTCP.

The TCP/IP traffic flows from your computer through your network card for a LAN, cable modem, or DSL (Direct Subscriber Lines). Your network card's driver is the software that interfaces between the computer and the network. For modem (phone) connections, the two software protocols that are used to

translate the Internet's TCP/IP protocol are the Point-to-Point (PPP) and the Serial Line Internet Protocol (SLIP). Whenever possible you should use the latest versions of this software for better performance. By the way, you can connect to the Internet without using TCP/IP, PPP, and SLIP, but you won't be able to use the World Wide Web if you do so.

While for many people the World Wide Web is the Internet, there are many services that run on the Internet. One of the oldest tools on the Internet is the Telnet service. Telnet lets your computer emulate a terminal and connect to and control another remote computer. When you use Telnet (which is a text-only service), you get a real feeling for what it was like to use the Internet to connect to hosts prior to the days when the Web existed. The service often offers not only commands but text-based menus. Other services you will encounter are the **File Transfer Protocol (FTP),** which is widely used; Gopher; WAIS; Archie; UseNet news services; a variety of mail services such as the Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and IMAP; in addition to HTML. HTML is the means used to describe Web pages to browsers.

You will find a variety of forms of HTML, of which SHTML (a secure version) sends encrypted traffic over the net to secure servers. You know when you are using SHTML because you see an address shtml:// appear in your browser's address bar, and a lock icon in the status bar. Each of these services and protocols is assigned a specific port with which it can communicate to and from a Web server.

Internet Architecture

The original architecture of the Internet was called the **client/server** model—something also called a two-tiered architecture. A server is the computer that runs an application (like a Web server) and stores and processes data, while the client requests the services of the server, receives data, and displays it. In its most common form, a PC or terminal serves the role of a client, and a more powerful computer plays the role of a server.

Web Servers

Application servers that serve up resources based on HTTP requests and transfers are called hosts or Web servers. Most of the time Web servers serve up Web pages that are displayed in a browser. The most widely deployed Web server is APACHE, an open-source product that is programmed by the general community and managed by a committee. Versions of APACHE are available for all platforms. APACHE is actually a play on words, derived from "a patch-y server." The Web server that probably is responsible for the most traffic on the Web is Microsoft's Internet Information Server or IIS. There are many different Web servers on the market today including Sun's iPlanet Web server (derived from Netscape's Enterprise server), Zeus, and others that are

purely Web servers. However, any application can add an HTTP stack and serve up Web pages, so you will find Web servers embedded in applications like FileMaker Pro and Oracle.

Web servers do more than just handle HTTP requests; they also send requests to run scripts such as Perl or Common Gateway Interface (CGI) scripts to the applications that can handle them. These program calls run applets that perform a variety of tasks such as database record retrieval, forms processing and posting, and animation. For example, many of the add-in components that are native in Microsoft's FrontPage application require that the request programs (FrontPage services) be loaded on the server. Web servers also provide address handling, security, and a variety of system configurations such as bandwidth throttling (controlling the amount of bandwidth available to a client or application), path selection, and so forth.

Keep in mind that the notion of client and server is turned on its head when UNIX X-servers are discussed. In X-architecture, the PC or terminal is the server, while the application running on the X-server is the client.

This simple Internet architectural model has over time been significantly modified in a number of ways. Since the processing demands of many users' client systems could overwhelm even the most powerful server, programmers have the option of structuring programs so that they use either server-side or client-side processing—or more often a combination of the two. When you add components to your browser—ActiveX components, Java applets (that use a client's Java Virtual Machine), Flash components, and so forth—you are enabling client-side processing. The addition of server components such as CGI scripts, Perl, and other programs enables server-side processing.

Multi-Tiered Architectures

While a two-tiered architecture serves well for many Internet applications, particularly ones with small transactions, it is poorly suited to multi-step processes where an interruption in the connection could put the entire data transaction in jeopardy. Consider what happens in a travel reservation system such as Travelocity.com or Expedia.com when you request a flight. You are in the process of paying for the flight when your connection dies. In a two-tier system, the transaction would be voided (since it wasn't completed), and the heavy-duty Oracle, DB2, or SQL Server database that underpins the system would be forced to `ROLLBACK` the transaction to the system's previous state in order to release the seats for sale to someone else. `ROLLBACK` is a SQL command that throws away any unfinished transaction(s) and returns your database to a previous state. To make this system more resilient and efficient (and not lose the sale), application developers have adopted a three-tier architecture with the following layers:

- Presentation layer. The Presentation layer takes the role of the client, requesting services and data, displaying results, and accepting input from the user.

Tech Tip

Directory Servers

Directory servers are application servers that keep a central database of objects, their properties, and their locations. They are valuable in streamlining programs because directories store information in a centrally located place on the network—and they are extensible (you can add your own information). Most directory servers are organized around the LDAP standard, from Microsoft's Active Directory (the key feature added to Windows 2000 networks) to Novell's ZENworks to Sun's iPlanet. They will play an increasingly important role in Internet applications, as they will in networks in general.

- Business Logic layer. In this layer are transaction or messaging servers (often called "middleware") that store the transaction and its state and maintain and apply rules that establish the validity of the transaction.
- Data layer. The Data layer contains the application servers that `READ` and `WRITE` the data to storage.

The three-tiered architecture is illustrated in Figure 1.2.

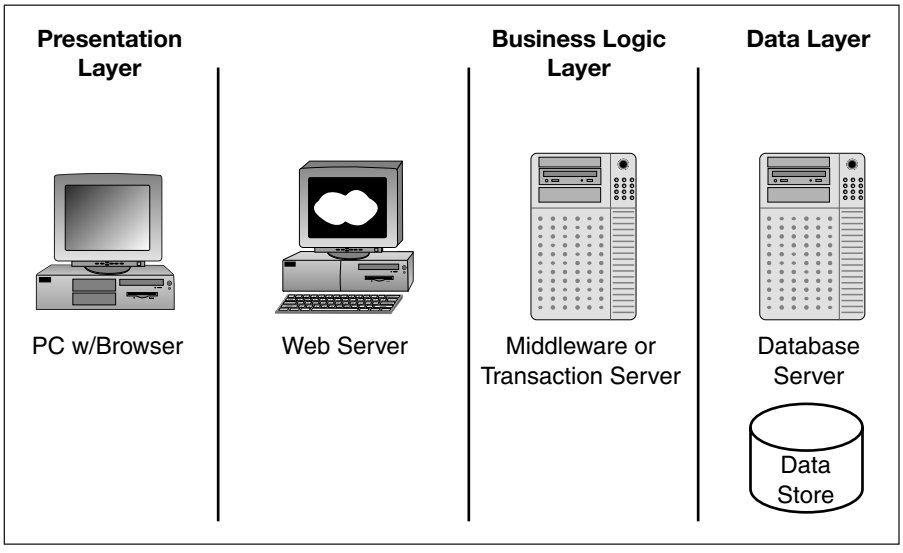

FIGURE 1.2 Three-Tiered Architecture

A three-tiered architecture allows transactions to be better processed when a connection is intermittent or transient.

In really large systems, the three-tiered model can be extended into an *n*-tiered system with multiple middle layers. Often programmers find themselves writing software for one of these layers or another. Examples of applications that run in each of these layers are a browser on a client for the Presentation layer, IBM or Microsoft Transaction Server for the Business Logic layer, and an enterprise database like Oracle as the backend Data layer.

Returning to our example of a flight purchase, when you initiate the reservation, the transaction is stored on a messaging or transaction server. Should you lose your connection, the transaction is secure. When the connection is regained, you are logged back into the transaction server, where you pick up your partially completed transaction and proceed. Essentially the *n*-tiered model eliminates the need for the client to directly access the server with a reliable or persistent connection.

QuickCheck Questions

5. How can you tell that you are using a secure HTML transfer protocol?
6. Why would a three-tiered architecture be useful to a messaging application?

Addressing

The Internet relies on a highly specified system for naming nodes, collections of nodes (domains), resources, and the means of accessing those resources. All of these addressing requirements are tightly specified by the standards bodies mentioned earlier in this chapter, and are published as part of a highly public vetting process. The IP (Internet) protocol is responsible for how nodes are addressed and a variety of security and data integrity services; so as we transition from IPv4 to IPv6, new capabilities are added. The TCP (Transmission Control Protocol) specifies how packets are formed and assembled, and how addresses are included into transmission. To begin with though, let's make the connection on how addresses are translated into friendly names, and how addresses are doled out. The net result of all of this is that resources on the Web are uniquely defined, identified, and accessible.

Domain Naming Services

The Internet Protocol version 4 specifies that Internet addresses are in the octet form ###.###.###.###, where each ### or octet can run from 0 to 255 (2^8). Thus www.cnn.com is 64.236.16.116, www.yahoo.com is 64.58.76.179, www.ebay.com is 216.33.156.119, and so forth. What an amazing chore it would be if we actually had to remember these numbers, as the numbers are essentially meaningless. So in order to use friendly names, the Internet uses a system called the **Domain Name System,** or **DNS.** With DNS, powerful *name servers* (called root servers) run a database that maps addresses to friendly names. Each domain extension such as .COM (commercial), .EDU (education), .GOV (government), .ORG (organization), .CA (Canada), .UK (United Kingdom), .BIZ, .NAME, and even .TV (for the Pacific island nation of Tuvalu) has its own name servers.

If users had to query the root name server every time they needed to translate a friendly name, the Internet would only be able to handle a small fraction of the traffic it does now. To improve DNS performance, DNS servers are placed throughout the Internet, at ISPs, for example, and on local area networks and recent requests and their responses are cached (that is, stored for later use). If your local DNS server doesn't have the answer, then it passes the request on up the line. Caching, as it turns out, is a common technique for enhancing Internet performance. You'll find cached Internet files on your computer (for Windows the cache is located by default at `C:\Documents and Settings\<Profilename>\Local Settings\Temporary Internet Files`), on your

local Internet access point (proxy server, firewall, or gateway), and even at your ISP. Large sites such as CNN.com use caching services like Akamai to replicate their data throughout the world.

While each address or node on the Internet is unique, the DNS friendly name system is organized into a hierarchical structure. When you register a domain with a registrar such as Register.com, say WebProgrammer.com, that domain is assigned an IP address. If you create additional addresses within your domain they would be named WebProgrammer.com/service, WebProgrammer.com/support, WebProgrammer.com/accounting, and so on. These second-level domain names can refer to real servers or computers, with or without a unique IP address, or they can be virtual subdomains. As you descend the hierarchy, you create addresses such as WebProgramming.com/accounting/receivables. In some instances you will see addresses of the type accounting.WebProgrammer.com that are mapped to the same subdomain.

To be really precise, the IP address that is assigned to a computer or system isn't actually associated with the computer itself. The address is assigned to the network interface, which in most instances is a network interface card (NIC) or a networking chip on your computer's motherboard. The essential distinction becomes important in computers that are multi-homed such as proxy servers, firewalls, gateways, and routers. A firewall may have two (dual-homed) or more network interfaces. One interface communicates with your internal LAN, while the other interface communicates with the outside world (usually the Internet). Both interfaces have very different addresses, and both may be running different protocols, in essence providing protocol isolation that makes it hard for hackers to penetrate inside your network.

Uniform Resource Locators

Not only is each numeric address on the Internet unique, but every document, file, or resource referenced on the Internet has a unique address referred to as the **uniform resource locator** or **URL.** A URL starts with the protocol that is used to connect to that Internet service, followed by the domain name, server (optional), and the path to the resource, finally followed by the name of the resource. Resources can be files, CGI (Common Gateway Interface) programs, Java applets (also programs), graphic files, and other things. A well-formed URL and its components are shown in Figure 1.3.

Consider the following URLs:

- http://www.nytimes.com/pages/world/index.html. This URL uses the **HyperText Transfer Protocol (HTTP)** to view information in the domain NYTIMES.COM on the COM name server. In the WORLD subfolder within the PAGES folder, the resource referenced is the INDEX.HTML file. This is an example of a static Web page, and the file INDEX.HTML (or HTM for Windows) is the default page for that folder. Static Web pages are pages that always contain the same data, while dynamic Web

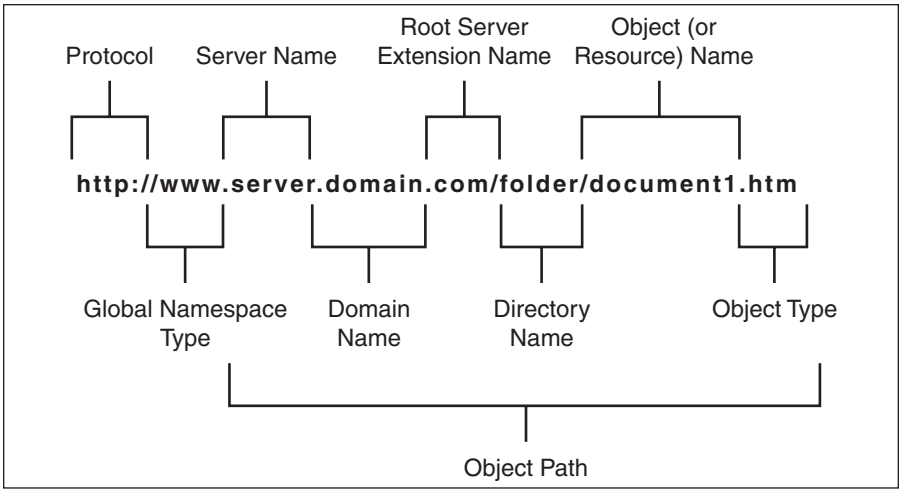

FIGURE 1.3 The Parts of a Well-Formed URL

pages are pages that change based on the user, client, or data submitted. If you enter http://www.nytimes.com, you are taken automatically to the INDEX.HTML page in the top-level folder. Keep in mind that folders could also be servers and may not necessarily refer to a hierarchy of directories in a real file system. Thus, the location of .../pages/world/... might be on one server, while /pages/national... might be on another server.

- http://www.here-now.org/archives.asp. This URL uses the HyperText Transfer Protocol to view information in the domain HERE-NOW.ORG of the ORG name server to reach a page called ARCHIVES. The suffix .ASP refers to an Active Server Page, which is a page composed on the fly using Microsoft's dynamic page specification. ASP and other dynamic page generation programming technologies are described in much more detail later in this book. Dynamic pages are used to query databases for current data or customized data and display that data on a browser or Web page.
- ftp://ftp.microsoft.com/. This URL uses the File Transfer Protocol to reference Microsoft's FTP site. FTP is a faster data transfer protocol used for file transfer. It references a hierarchical display of folders and files. Many FTP sites are viewable in part or as a whole as a GUEST with no password required. However, most FTP sites require logins in order to access specific areas.
- barries@business.earthlink.net. You probably recognize this as an email address. The first part, BARRIES, is the username, and after the "at" symbol comes the server name (BUSINESS), followed by the domain EARTHLINK.NET. More commonly, the server is omitted and you see an address that looks like barries@earthlink.net.

QuickCheck Questions

7. What is a name server?
8. What is CGI, and what does it do?

Network Types

IP addresses are assigned on the basis of the size of the enterprise and its purpose. These network sizes are described as Class A, Class B, or Class C. There are Class D and Class E networks, but they aren't assigned externally to organizations on the Internet.

The IPv4 numbering scheme is a 32-bit address, and the different classes break down as follows:

- Class A: nnn.###.###.###
- Class B: nnn.nnn.###.###
- Class C: nnn.nnn.nnn.###,

where nnn are fixed numbers and ### are variable numbers in the range of 0 to 255. Since the first three numbers are always assigned, you can tell which class the network is by the value of the first numbers. Class A networks are found in the range 0 to 127; Class B within 128 and 191; and Class C are found between 192 and 223. More importantly, the capacity of a Class A network is 16 million addresses, that of a Class B network is over 65,000 addresses, and Class C has 256 addresses (although only 255 are assignable).

Really large networks such as AOL, MSN, AT&T Worldnet, CompuServe; large companies such as IBM and Microsoft; and hosting services and telecommunication giants such as Earthlink and AT&T are assigned Class A addresses.

DHCP

If you examine IPv4 addresses, you will notice that there are one billion possible *static IP addresses* that can be assigned. The highest number possible is 999,999,999,999, while the lowest number is 000,000,000,000. One billion addresses seemed like an unimaginably large number to the researchers who first developed the Internet, but in an age when even toasters may be on the Internet along with all of China, getting a static IP address for your computer from an ISP is getting increasingly difficult and expensive. And, of course, there are many addresses that are reserved for special uses and aren't available for assignment.

To get around the problem of limited fixed addresses a system of internal routing was created using what started out as a `HOSTS` file. The routing server has a static IP address (typically), and the administrator assigns temporary

or dynamic IP addresses from a reserved pool of available addresses by entering them as records (lines) in a `HOSTS` text file along with their friendly name to clients. You can still find the `HOSTS` file in use in UNIX systems, and it's even squirreled away inside Windows' system folder (`C:\WINDOWS\SYSTEM32\DRIVERS\ETC`). The `HOSTS` file stores the friendly name of a computer and its internal IP address taken from a range of addresses that are "private" addresses and can never be assigned on the Internet by convention. You add, delete, and modify records in the `HOSTS` file by editing its text as new computers are brought online and others are retired. As you can imagine, once a network grows beyond a certain size, managing the `HOSTS` file becomes a terrific bother.

A system known as the **Dynamic Host Configuration Protocol,** or **DHCP,** automates what the `HOSTS` file did manually. DHCP is a network broadcast service that will automatically assign dynamic IP addresses from a pool and manage how long those addresses can be used (the length of the leases). When your computer connects to a network that uses DHCP, it issues a DHCPDISCOVER broadcast to find a DHCP server. DHCP server(s) will then issue a DHCPOFFER packet with an available IP address and the IP address of the server. Your computer will select an appropriate DHCP server and send back a DHCPREQUEST packet. Finally, a DHCPPACK packet confirms the address and establishes the lease. If you connect to your ISP by modem and have "Automatically Obtain an IP Address" selected (a dynamic IP address), you'll probably see a new IP address every time you log onto your service. For network-assigned addresses, your lease might run a few days, a few weeks, or a few months depending upon what your network administrator sets.

The World Wide Web

Most of us are familiar with the **World Wide Web (WWW)** and have worked with it in our daily work. The Web is a relatively new part of the Internet that offers a means of identifying and accessing data anywhere. A **hypertext** link can point to a binary file, a document, a graphics file, a location in a document, and almost any object that a computer program or file system can describe in any location. The World Wide Web's hypertext address space "virtualizes" location so that you don't need to be aware of where the information actually is stored, only how to reference its location.

The founders of the World Wide Web hoped that, as the Web was generally adopted, it would come to play a role in the way people lived and worked. Although most visualized that it would provide access to data, few would have guessed that the Web could store the state of transactions both online and locally (for example, as cookies), and that that state information could be used to tailor the interaction of the Web for a user in very powerful ways.

Tech Tip

Binary Files
A binary file is a machine-readable file that can be a program, data, or other resources that have been transformed into something that your processor can more easily process. If you open a binary file, it is filled with all kinds of letters and symbols. A program or .EXE file is one example of a binary file.

QuickCheck Questions

9. What is DHCP and why is it used?
10. To what kinds of things can a hypertext link point?
11. Why do Web servers require a static IP address?

A Brief History of the Web

To many people the World Wide Web, or just simply "the Web," is synonymous with the Internet. While the Internet has its roots in the 1970s when the work on packet-switching networks was being done, the Web as an invention is really only a dozen years old. That doesn't mean that the idea of a World Wide Web wasn't conceptualized earlier on. In the 1960s Ted Nelson first discussed the concept of links, hypertext, and hyperspace, and described how links could be made to resources leading to a new kind of educational or referential resource where the reader need not perform a sequential read in order to use the material effectively. Nelson further made the analogy to how the human mind works and learns, and to a new concept—electronic books. Nelson's contribution is more fully described at http://www.iath.virginia.edu/elab/hfl0155.html.

Nelson's work was highly regarded and launched several efforts to create a hypertext system with a global namespace. One such effort (of many) was started at the University of Minnesota and named Gopher.

Most people credit the British researcher Tim Berners-Lee with the creation of the World Wide Web. Berners-Lee, who is now the 3Com Founders Professor at the Laboratory for Computer Science (LCS) at the Massachusetts Institute of Technology (MIT), also chairs the World Wide Web Consortium, a steering committee for the development of the Web. Berners-Lee was a physics researcher in 1989 at the CERN European Particle Physical Laboratory when he proposed a global hypertext address space where information could be referenced by what was originally called a "Universal Document Identifier" (from which the URL is derived), and the information could be fetched using his and his coworkers' HyperText Transfer Protocol.

Tech Tip

Web Resources
Tim Berners-Lee's home page at www.w3.org/People/Berners-Lee/ is a good source of reference links to Web resources.

The Internet has more services than just simply the Web, but the Web has become far and away the most important part of the Internet. The vast majority of the programming jobs you will find that are Internet-related are jobs that program applications or services for the Web.

Browsers

Berners-Lee went on in 1990 to create a program called WorlDwidEweb, which was a graphical (point and click) hypertext editor for the NeXT UNIX workstation. He then wrote a Web server and released both into the public

domain in the summer of 1991. Also released at that point was the specification for UDIs, HyperText Markup Language (HTML), and HyperText Transfer Protocol (HTTP), which was put on the first server (info.cern.ch) to promote its adoption. The aforementioned server is no longer online, but CERN's address is http://public.web.cern.ch/public/, with information about the Web found at http://public.web.cern.ch/public/about/achievements/www/www.html. An account of his work also may be found in his book *Weaving the Web,* by Tim Berners-Lee with Mark Fischetti (Harper San Francisco; hardback: ISBN: 0062515861).

The World Wide Web really took off when the next generation of browsers such as Erwise, Cello, Viola, and Mosaic began to appear and capture the general public's imagination. The early browsers provided hypertext links to documents, some basic page composition (headings, lists, and text formatting, for example), but no images. Mosaic, developed at the University of Illinois's supercomputer center (www.ncsa.uiuc.edu) by Marc Andressen in 1993, added images and more page composition features and is generally regarded as having started the race to develop the dominant Web browser.

There were a number of commercial Web browsers under development in 1994 when Jim Clark (one of the original founders of Silicon Graphics workstations) joined with Andressen to form Mosaic Communications. By 1995 Mosaic had changed its name to Netscape and its Mozilla logo to the current Netscape stars logo, and released Netscape 1.1, followed by 1.2 for Windows 95. Netscape popularized the World Wide Web beyond the rarefied halls of science by adding many of the HTML extensions that made page composition more powerful and Web pages more attractive that we take for granted today. It would be difficult to underestimate the impact of Netscape; by the summer of 1995, the company owned more than 80 percent of the entire browser market. Netscape Navigator 2 in March 1996 was the second seminal product introduced by Netscape and added frames, Java, JavaScript, and the plug-ins extensible component architecture, while Navigator 3, introduced in August 1996, added background colors in tables, additional formatting options, frame borders, spacers, ARCHIVES, and APPLET elements.

In 1995 Microsoft entered the Web browser market by introducing Internet Explorer as part of its Plus package and began a two-year marketing and technology battle in what would come to be called in the trade press the "Browser Wars." By 1997 Microsoft had bundled Internet Explorer in the Windows operating system for free, and by 1998 had become the dominant browser and market leader in terms of client usage. Navigator was a commercial product requiring a paid license, which placed Netscape at a disadvantage. This battle was in large part responsible for the antitrust action filed against Microsoft by 18 states and the District of Columbia during the Clinton administration that has recently been settled by an agreement with Microsoft and the Justice Department to make it easier for other vendors to become the default application and for OEMs (original equipment manufacturers) to modify the Windows desktop.

In Figure 1.4a, we have shown the home page for the http://www.cnn.com/ site. You may use the **View | Source** option of Internet Explorer's menu to see the source code that was received and rendered by the browser. Figure 1.4b

FIGURE 1.4A Internet Explorer Rendering of CNN.com's Web Site

a. Graphical display of CNN.com's home page.

shows a partial listing of the source code. Observe and compare the graphical display of the Web page (Figure 1.4a) with the source code (Figure 1.4b). You will see that most of the source code is HTML, but many other external capabilities are also called.

Browsers have become multifunctional, incorporating mail programs, graphics rendering, file format translation, data and code interpreters (XML, for exam-

FIGURE 1.4B Internet Explorer Rendering of CNN.com's Web Site

b. Underlying source code view of CNN.com's Web site. What you see is that most of the code is HTML, but many other external capabilities are called, such as Java applets.

ple), and more. Given that browsers are also extensible, Netscape first and then Microsoft have built a whole suite of products around their browser. Today there is the distinct possibility that operating systems will play a supporting role to browsers, a possibility that Microsoft was forced to defend against in the marketplace by their development of Internet Explorer and that browser's strong attachment to the Windows operating system. On Windows, HTML is one of the formats in which you can save Microsoft Office files. The Active Desktop on Windows also lets you have ActiveX components on your desktop that will actively display stock tickers, the weather, and Web pages without your browser running.

Microsoft has the dominant position in the marketplace in Windows and a majority usage on the Macintosh, and does not compete in the UNIX and Linux marketplace. Netscape Navigator is in second place based on its majority usage on UNIX systems and a sizable percentage of Macintosh users.

QuickCheck Questions

12. Who was credited with popularizing the concept of hypertext?
13. Why are browsers so important?

Portable Languages

Another very important development in the history of the Web has involved the development of portable languages to support multiplatform programming. This is best exemplified by Sun's introduction of its Java programming language in 1991 and the JavaScript development scripting language. Java was the result of a development effort aimed at creating a language that could work with all kinds of small portable devices. It was adapted to the Web, where its ability to create programs you could "write once, run anywhere" was seen to be an important advantage in an environment meant to support multiple and disparate operating systems. With Java you install a core instruction kernel for a particular operating system called a Java Virtual Machine or JVM. Any Java applet or program that runs on that platform calls the services of the JVM and will not operate if it doesn't find those services.

The "Green Project," as it was initially called, developed as a demonstration on an interactive, hand-held device controller with an animated touch screen the called *7, or "Star Seven." The name arises from the feature on the Sun phone system that lets you answer your phone from any extension. Sun tried to apply this technology to TV set-top boxes and built a demo called First-Person—but this type of interactive TV was about 12 years early to market. So Sun set about looking for other areas where Java could be used, and the Internet was a natural choice. A detailed history of the development of Java may be found at http://java.sun.com/features/1998/05/birthday.html.

Sun's team built WebRunner, a Mosaic clone, a browser that came to be called HotJava, which is still bundled with the Solaris operating system, and was able to demonstrate some impressive multimedia technologies. Sun released Java into the public domain in 1995, and eventually the capabilities of Java caught the attention of Netscape, who included it in Navigator in 1995. Eventually the entire industry licensed the Java Virtual Machine, including IBM, Apple, and Microsoft. Java has spawned not only a scripting language, but the Java Development Kit (JDK), the JavaBeans distributed application architecture, thousands of servlets, tens of thousands of applets, Java Foundation Classes (an object model), the JavaOS for Business, and, most importantly, a generation of very capable Java-based applications that are now the standard method for programming cross-platform multi-OS applications.

Microsoft has always had a love/hate relationship with Java, primarily because the company saw it as a threat to their core operating system business. After all, why program applications for Windows when you can create applications for browsers and have these applications run everywhere? But Microsoft was in a quandary. They needed Java to fit into network and enterprise-computing environments, but they also had in-house technologies that they wanted to promote. For example, Microsoft's Visual Basic has five million programmers, and the company wanted those programmers to be able to create Internet applications. Microsoft also has a different object model called the Common Object Model, or COM. Microsoft has promoted COM as a programming model, and components written with the COM model are called ActiveX components. Microsoft first extended COM across the network as DCOM, or Distributed COM, and more recently to their initiative for the development of Web services.

The net (so to speak) result of all of this is that Microsoft created extensions of Java to support their technologies, making it difficult for many Java applets to run properly on the Windows platform. Sun, who licenses Java, accused Microsoft of violating the Java license. After litigation, Microsoft was found in violation of their Java license and forced to pay a fine. But the final result seems to be that Microsoft is moving away from Java. In the next big step, Web services, a portable programming language, is seen to play a pivotal role. Microsoft's Web services program is dubbed .NET (pronounced "Dot Net"), and for the programming framework that Microsoft has built, they have introduced a new language called C# (pronounced "See Sharp") that builds on the work of Java.

Tech Tip

Object Models

Once you understand an object model, its hierarchy, and how to address objects and use its methods to provide a variety of functions, you have a powerful tool upon which to build your programs that will both be elegant and save you a tremendous amount of programming time.

HTML: The Language of the Web

The HyperText Markup Language, or HTML, is the method used to specify page composition on the Web. At least, that is the way HTML started out. As you will see in the next chapter, and subsequent chapters, HTML has picked up tags that let you do much, much more with your pages than just simple formatting and page layout. HTML uses a subset of SGML, the Standardized General Markup Language. We won't talk much about SGML in this book, although not only is HTML based on it, but XML (eXtensible Markup Language) is as well. XML, as you will find out later in Chapter 3, is becoming important for database Web transactions and for data and document transfers. SGML is an international standard markup "language" that lets you define in a text file different parts of a document.

I put the word "language" in quotes because if you ask any hard-core programmers, they will let you know in no uncertain terms that HTML isn't really a programming language. And in a sense they are right. When you think of a well-specified programming language, you think in terms of using different logic functions, defining variables of different data types, error checking, and other features that aren't really part of HTML (or SGML or

XML for that matter). HTML contains a set of instructions on how to compose a page. You can see the underlying HTML code by selecting the View Source command in Internet Explorer or its equivalent in other browsers. The Source view is what you would see in an HTML editor or in a Web site composition tool, and represents the code or instructions necessary to compose the Web page.

If you browse an HTML document, you will find a variety of tags using the following syntax:

```
<OpeningTag> . . . stuff . . . </ClosingTag>
```

Sometimes it's hard to find the closing tag because it's a long way away from the opening tag: for example, the `<BODY> . . . </BODY>` tags. Also, although it's good programming practice to use both opening and closing tags, not all tags require their ending tag. An example would be the `<BR>` or Break tag. However, since XHTML demands well-formed tags (both opening and closing pairs in the right precedence or sequence), we always write code that conforms to this practice.

Tech Tip

Parsers

The part of a browser that "reads" HTML is called a *parser*. The parser is responsible for recognizing tags, checking for the tags' elements, and implementing the instructions. Parsers have rules that determine which tags to implement first and in what order (called *precedence*). Each succeeding version of your browser improves on the previous version's parser, both adding functionality as well as optimizing performance. Not only do some browsers have HTML parsers, but they might include other parsers such as XML parsers as well.

In practical terms, HTML's files (which take an HTM extension for PC files and usually take an HTML extension for Macintosh files) are a lot like the Adobe Page Description Language or PDL files that the Acrobat Reader reads. Both are readable text files; that is, you can open either file type using the Windows Notepad, Macintosh TextEdit, or UNIX Text Editor (or vi, the Visual Editor). Underlying PDL is the Postscript graphics programming language, which is a very complete graphics manipulation environment, while HTML relies on other programming languages to code specific functionality.

HTML became important because it was chosen as the method used to display pages in a class of software applications that came to be called browsers that run on client systems. The applications on which pages are served up to browsers on clients are now referred to as Web servers. You can have large powerful computers running as Web servers, or have a personal Web server on your own desktop such as Microsoft Personal Web Services that allows a much smaller degree of connectivity in peer-to-peer computing networks.

HTML has been continually expanded and extended over time, adding things such as Cascading Style Sheets, or CSS. In addition to different specifications of the standard, you will see technologies like Dynamic HTML or DHTML used for animation effect or changing a Web page on the fly and eXtensible Markup Language, or XML, which is used for document-based data transfers. DHTML is a collection of technologies that can use the Document Object Model, or DOM, CSS, along with client-side scripting languages to animate objects on a Web page. You'll learn about these technologies further in chapters to come.

QuickCheck Questions

14. What happens if you don't use a closing HTML tag to match the opening tag?
15. How do you view the underlying HTML in a browser?

The Web Browser: Your Window on the Web

Web browsers work by reading a text file containing markup instructions. While you can open any text file in a browser, only files with the file extension HTML or HTM that have been associated with a browser application will automatically be opened. Your browser uses the instructions contained in the tags to compose a page and display that page inside a browser window. Not only does a browser read the HTML file to display the text it contains, it also

- Formats the text.
- Adds heading, lists, and tables and links to resources or objects located elsewhere.
- Displays images (which are downloaded files), most commonly in the form of JPEG or GIF image files that reference a URL in another location.
- Runs scripts or small programs written in JavaScript, VBScript, Pearl, CGI, and a variety of other scripting languages (which are described in subsequent chapters in this book).
- Calls small programs that have been added to the browser's plug-in architecture (often called helper applets). An example of a helper applet would be when your browser uses the Adobe Acrobat Reader plug-in to open a PDF file.
- Calls external programs used to provide additional capabilities. External program calls can display an independent object in a browser frame such as a 3-D interactive visualization of a car with its component parts; or let you hear or listen to a video and audio stream in Real Audio, QuickTime, Windows Media Player, or a host of other programs.

All of these actions are instructions based on the HTML tags and the information that is embedded inside them.

An HTML file doesn't contain the images, scripts, or programs defined in the appropriate tags. When a resource is required, your browser makes a request for the object at its URL (uniform resource locator), most often using the HyperText Transfer Protocol (HTTP). To optimize network performance, first your browser's local cache is checked for the object, then your local server, and then your ISP's server; finally, the request goes out to the actual location of the site to do the data transfer. Caching is where you dedicate a portion of either your memory or disk space to recently downloaded information.

It is a very powerful method for boosting performance because your browser fetches the data from RAM or disk with none of the latency or lag time associated with downloading that information from the Internet.

Many browsers support other protocols besides HTTP. The second most common transfer protocol supported by browsers is the File Transfer Protocol, or FTP. FTP is a much more efficient method for uploading and downloading files than HTTP; and although browsers can perform FTP transfers, they aren't quite as fast as the FTP transfer utilities that you can find. To get a look at an FTP site in your browser, go to ftp://ftp.sun.com/ to view Sun's FTP site (shown in Figure 1.5). FTP sites often provide read-only or limited access to their sites as a guest (no password) until you log in as a specific user.

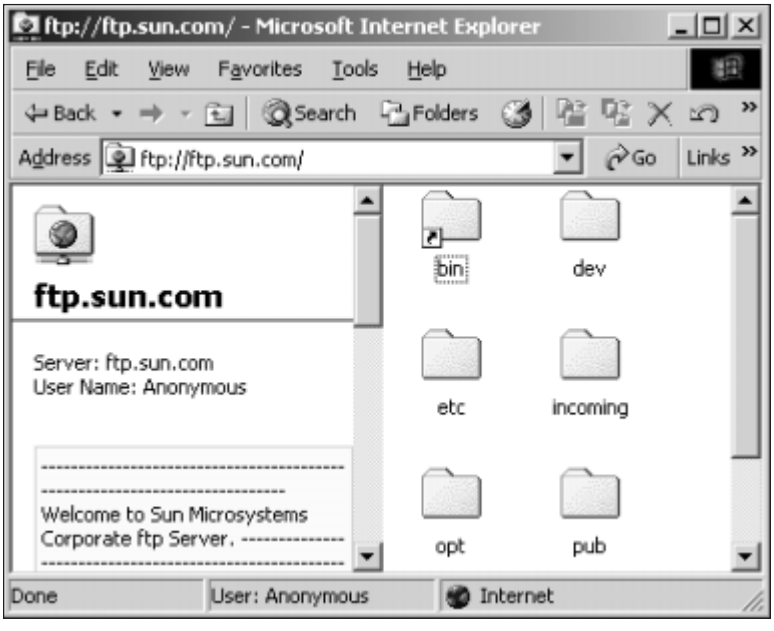

FIGURE 1.5 An FTP Site

Browsers also often display and use FTP sites. Here is Sun's site displayed in Internet Explorer.

Browsers essentially paint your screen in an order that is optimized to put the fastest, easiest-displayed information on your screen first. Other information that must be fetched is rendered later. If you have ever watched a browser display a page on a very slow computer, you can readily assess how a browser goes about its business. First, elements that the browser can render without outside resources are painted on the screen: borders, frames, and text. Then as resources such as images start to become available, they are painted one after another. So small image files like GIF representations of buttons in a toolbar

might paint in first; larger images like photographs would appear next; and if there is something that requires calculations, is large, or is accessed over a slow connection, that appears last.

To improve browser rendering speed, Web site developers spend a lot of time optimizing their pages, reducing file sizes and the number of colors (color levels), or using special tricks like interlaced GIF files. An interlaced GIF (Graphic Image File) file is a bit-mapped (paint type) file that is described in successive chunks. GIF files are built up one pass at a time as additional lines are drawn, appearing to become better resolved over time. This technique and file format, first developed by America Online, gives the viewer the opportunity to see a low-resolution image that gives them the idea of the content, and thus speeds up the decision to move onto another page.

Sometimes resources are not always available: you lose your connection, you are forbidden access, and so on. Many browsers display standard error messages, but unfortunately don't often describe them in English. From Internet Explorer 5 on, more friendly messages are displayed. Table 1.1 gives a few of the more common browser errors.

TABLE 1.1 Common Browser Errors

Error Number	Title	Meaning
400	Bad Request	The server cannot parse the URL you entered. Check that you have used all of the appropriate symbols and letters.
401	Unauthorized	You do not have privileges to view that URL.
403	Forbidden (or Connection refused by host)	This resource is not available to you; you may not have the appropriate username, have entered your password incorrectly, or some other reason.
403.9	Access Forbidden. Too Many Users Connected	Web servers allow site managers to limit the number of users that can connect to a particular page, folder, or their site.
404	Not Found (or File not found)	The Web site is live, but the page can't be located. Check your URL carefully. Sometimes you can circumvent this problem by removing the last level and moving up a folder in the URL's path (e.g., changing http://www.sosinsky-group.com/reports/storage.htm to http://www.sosinsky-group.com/reports.
409	Proxy Authentication Required	There is a problem with your connection through the proxy server or firewall. Check your client software for the correct proxy address and port number.
500	Internal Error	There is a server problem.

Continued

TABLE 1.1 Common Browser Errors (continued)

Error Number	Title	Meaning
503	Service Unavailable	The Web site can't be contacted. This can mean that the site's Web server has stopped responding, your connection is down, or a variety of other problems.
	Bad file request	Your browser may not support this file type.
	Computer connection timed out	A slow or dropped connection may generate this message. Try again later.
	Server Does Not Have a DNS Entry; Failed DNS lookup; or DNS entry not found	The DNS service cannot match your URL to a valid friendly Internet address. Check the URL, try the page again, or try later.
	File contains no data	The site exists, but your page doesn't. Sometimes this occurs when a site is being refreshed or is under construction. Try the URL again.
	Helper application not found, or Viewer not found	The browser can't associate the file with an applet.
	Host unavailable	Server may be down; try later.
	Network connection was refused by host; or Too many connections; Try again later	There are probably too many connected users at the moment.
	No virtual host specified	Occurs when you submit a form (typically in Internet Explorer). You should refresh the page in order to determine if the form was submitted.
	Unable to locate host	The server may be down or you may have lost your connection. Try refreshing the page and checking your URL for typos.
	Invalid Page Fault in Kernel132.DLL	Found on Netscape when system resources aren't available, you are using out-of-date video drivers, or have a corrupt swap/paging file. Try closing your applications and restarting.

Summary

The Internet is a network of networks tied together by a common set of services and protocols. The two key ideas that give the Internet its capabilities to both be resilient and have global reach are that it is built with a packet-switching fabric architecture and that it can support a global namespace that uniquely identifies resources for use be they on your computer or halfway around the world. Development on the Internet has used a standards-based approach so that its protocols like TCP/IP and HTTP (for example) are now de facto standards for all computer operating systems, and most applications. In time the bulk of the networked computer programs written will be written to work over the Internet.

The most influential part of the Internet is the World Wide Web. The World Wide Web uses servers running the HTTP protocol to provide resources so that a class of applications called browsers can display pages and documents on your local computer. Browsers are now powerful software suites, with extensible plug-in architectures that allow the addition of powerful small programs called applets or helpers that can perform almost any computer function imaginable. The need to communicate with different kinds of computers makes browsers a unique application to use. To further support cross-platform computing, the industry has developed a number of "portable" languages such as Java and C# that can be written once and run on many platforms using an interpreter on that platform.

We are entering an era where Web services will play an increasingly important role in providing application functionality. So Web programming may be seen to be one of the brightest and most critical disciplines in all of computer science.

Key Terms

browser
client/server
Domain Name System (DNS)
Dynamic Host Configuration Protocol (DHCP)
File Transfer Protocol (FTP)
hypertext
HyperText Transfer Protocol (HTTP)
Internet
name server
packet
packet switching
port
protocol
registrar
router
socket
tags
Transmission Control Protocol/Internet Protocol (TCP/IP)
uniform resource locator (URL)
World Wide Web (WWW)

Review Questions

1. What happens when a packet is lost and does not arrive at its intended destination on the Internet?
2. What function does the HTTP protocol serve?
3. What purpose does a socket serve?
4. How do routers work, and what service do they provide?
5. Describe how friendly names are obtained from IP addresses.
6. Why is caching so important, and where is caching used on your computer as well as on the Internet?
7. What advantages does a three-tiered architecture have for a high transaction system such as an Internet reservation system?
8. Why was Java such a programming sensation when it was first introduced?
9. What is a URL?
10. What kinds of Web services do you think would be particularly valuable to programmers?

Lab Exercises

1. Your company wishes to publish an online news magazine on the Web. Describe the process required to register a domain, create content, and upload that content so that it is available to outside people to view.
2. Find some examples of Web sites that look different in Microsoft Internet Explorer and in Netscape Navigator (or some other pair of browsers). What do you think accounts for the differences that you see?
3. Take a screenshot of the CNN Web site. Identify all of the different elements on the screen that you can.