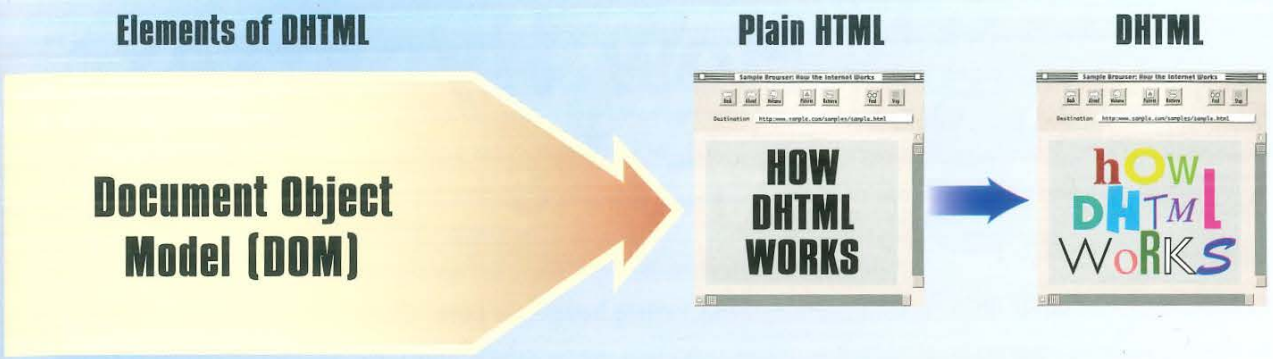
3 Although DHTML is often referred to as if it is a single technology, it is, in fact, a general term used for a group of technologies that can work together or by themselves to change a Web page after the page has been downloaded to your computer. These technologies are the Document Object Model (DOM), Cascading Style Sheets (CSS), and client-side scripting languages, such as JavaScript.



4 The *DOM* defines every object and element on a Web page and enables those objects to be manipulated or accessed. This includes fonts, graphics, tables, and visual elements, as well as elements you can't necessarily see, such as the browser's version number and the current date and time. Without DOM, all the elements on a page are static. So on the simplest level, DHTML could use the DOM to change the font of every letter, individually, on a Web page.



5 *Cascading Style Sheets* are, in essence, templates that apply formatting and style information to the elements of a Web page. They're called cascading because any single page can have more than one style sheet associated with it. Additionally, Cascading Style Sheets enable images to overlap one another. This enables animations to be created easily on a page.

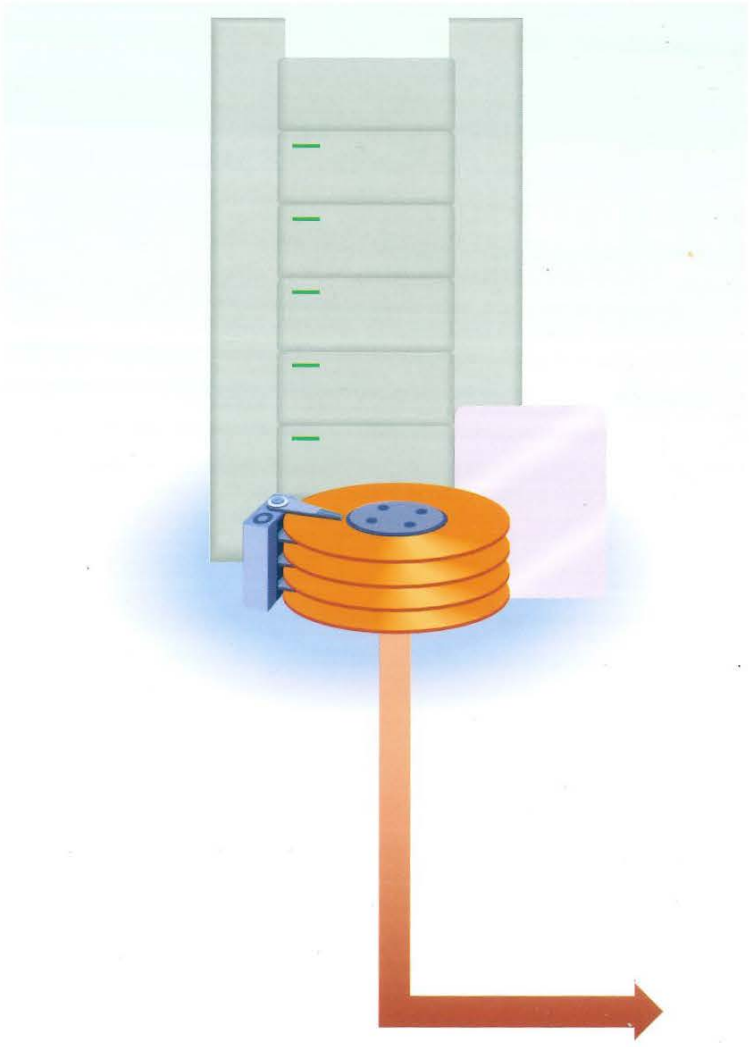


6 *Client-side scripting languages* perform much of the work of DHTML. These languages access the DOM and manipulate its elements, and they do the same to Cascading Style Sheets. They perform the actions of DHTML. So a script, for example, could turn a word a different color when a mouse moves across it, or it could create easy-to-use collapsible navigation on every page on a Web site.

VIANT-1047, Part 2 of 2

CHAPTER

28 How Hypertext Works



IN the late 1960s, a computer scientist named Ted Nelson introduced hypertext, a concept that lays the foundation for the World Wide Web and its connections between documents, or pages.

Nelson wanted to create a new way of exploring information. He wanted to provide the reader with a spontaneous means of accessing more and more in-depth information about something that sparked the reader's interest when reading text on the page. Rather than reading a document from beginning to end, digesting the material in a sequential order, the reader could highlight a word and receive more information on the meaning of that word, for example.

Nelson imagined that you could read the United States Constitution, come across the term "electoral college," and then open another document that explained how the electoral college works. From that document, you might open another document that listed the votes of the electoral college from its founding to the present. From there, you might choose to open a document about William Jefferson Clinton, then another about the First Cat, Socks, then another about the care and feeding of felines, and so forth. Ultimately, you could find a subject that wasn't even remotely connected to the Constitution, but which you would find interesting or entertaining.

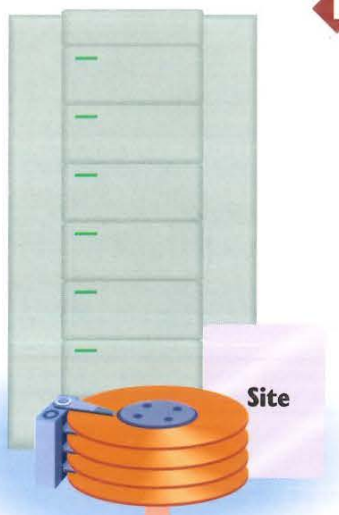
This hypertext concept was obviously on Tim Berners-Lee's mind when he began thinking about how researchers could share their work across the Internet. He envisioned a system where a document could be linked to other documents, enabling researchers to easily find more and related information simply by following a link from one document on the network to another.

Typically, hypertext consists of a hyperlink that appears onscreen as a highlighted word, icon, or graphic. By moving a mouse cursor over the item, or object, and clicking on it, you easily navigate to additional information. On the Web, that information can be located at any other place on the Web, be it on the same host server or one across the globe. A linked object can be various media, such as text (linking from one character to a whole document, for example), a graphical button (such as direction arrows that move from page to page), or still images (photos, icons, or a comic strip), for example. The documents and objects that are being linked to can be on the same site as the original document, or in an entirely different document.

Hypertext links are embedded into a Web document using HTML, Hypertext Markup Language. A text link usually appears on the screen as an underlined word or phrase and is sometimes rendered in a different color from other text, depending on how your Web browser interprets the HTML codes. When you place the mouse cursor on this underlined text and click the mouse button, you initiate a request by the browser for a new Web page or—if the text references an internal link to information in the same document—direct your browser to scroll to another, specific point within the same document.

Images or icons can also act as hyperlinks. When you move the mouse cursor over the icon or graphic and click the mouse button, you launch the request to retrieve the linked information.

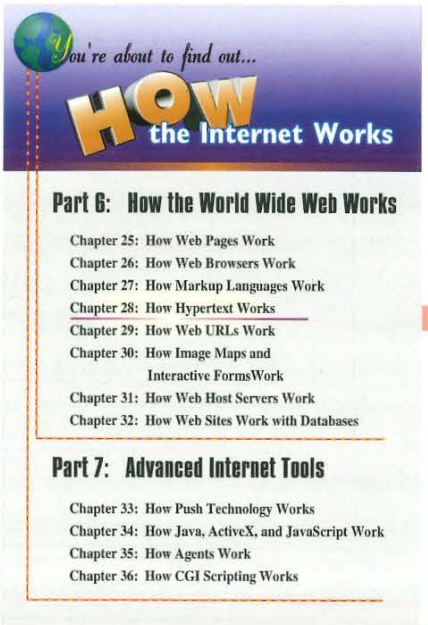
How Hyperlinks Work



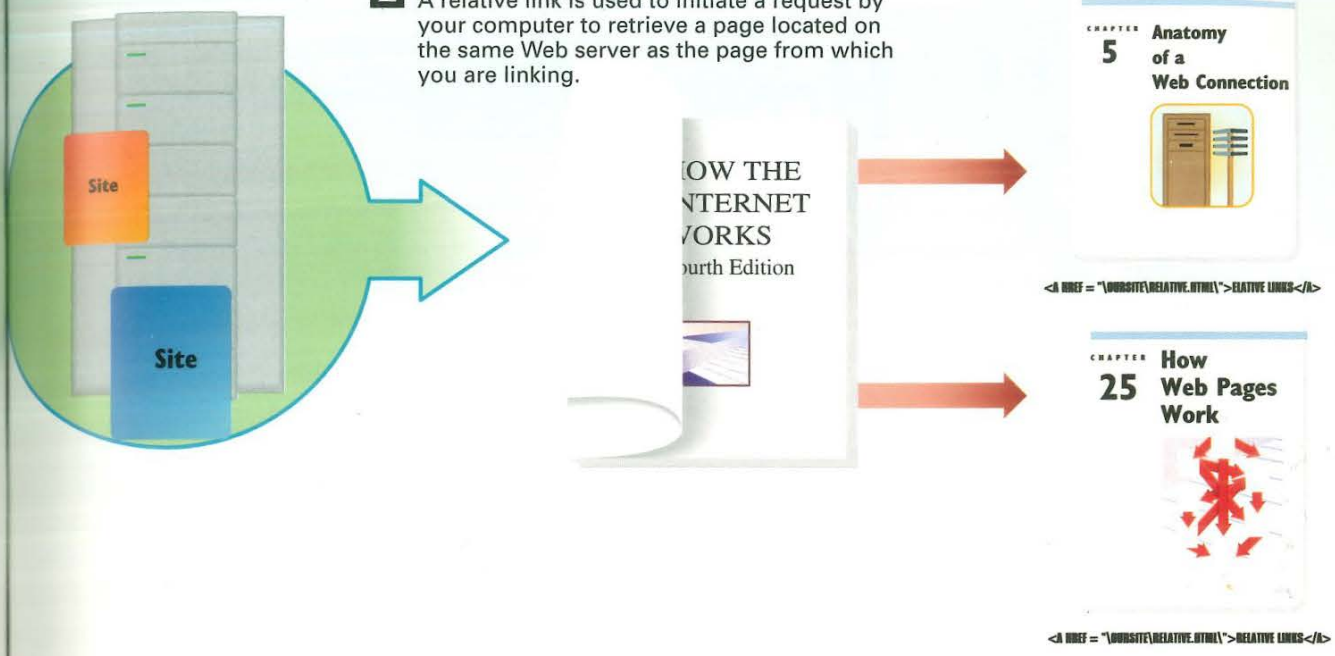
1 The "hyperlinking" begins when you first retrieve a Web page from a remote Web server. Target links within the page move you quickly from one part of the page to the next.

CHAPTER 28 How Hypertext Works



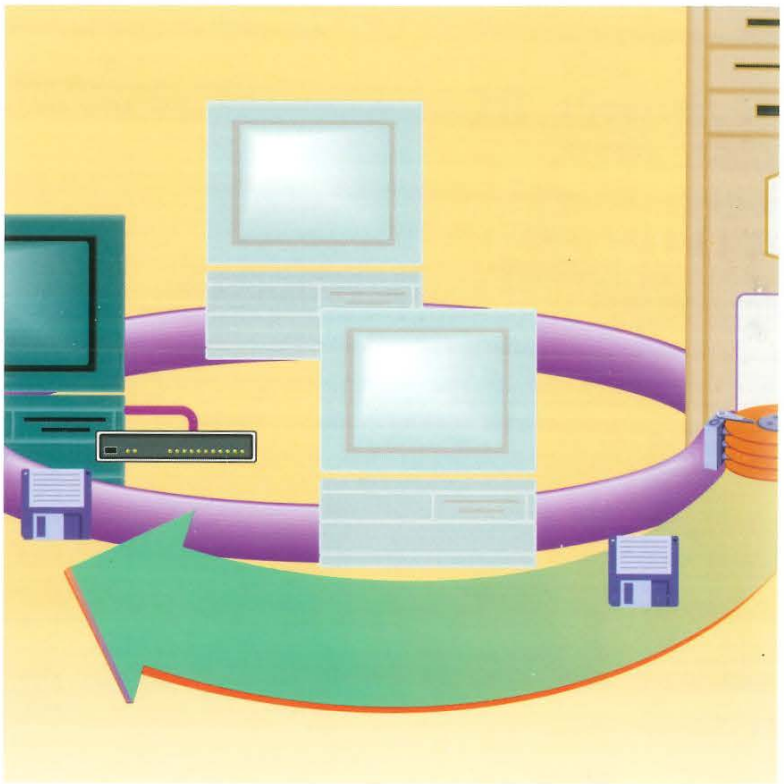


<A HREF = "#TARGET">TARGET LINKS<A/>


2
A relative link is used to initiate a request by your computer to retrieve a page located on the same Web server as the page from which you are linking.
Site
Site
OW THE
NTERNET
ORKS
urth Edition
CHAPTER 5 Anatomy of a Web Connection
<A HREF = "\OURSITE\RELATIVE.HTML\">ELATIVE LINKS</A>
CHAPTER 25 How Web Pages Work
<A HREF = "\OURSITE\RELATIVE.HTML\">RELATIVE LINKS</A>
3
A hyperlink that leads to a completely different Web server uses an absolute link.
<A HREF = "http:\\www.linksite.com\"ABSOLUTE LINKS </A>
OW THE
NTERNET
ORKS
urth Edition
<A HREF = "http:\\www.linksite.com\"ABSOLUTE LINKS </A>
HOW THE INTERNET WORKS FOURTH EDITION
PRESTON GRALLA
<A HREF = "http:\\www.linksite

CHAPTER

29 How URLs Work



THE Web pages and the hosts that make up the World Wide Web must have unique locations so that your computer can locate and retrieve the pages. The unique identifier for a host is called the IP (Internet Protocol) address and the unique identifier for a page is called the URL, or uniform resource locator. A URL functions much like a postal or email address. Just as postal and email addresses list a name and specific location, a URL, or Web address, indicates where the host computer is located, the location of the Web site on the host, and the name of the Web page and the file type of each document, among other information.

A typical URL looks like this:

```
http://www.zdpress.com/internetworks/index.html/
```

If you were to interpret the instructions in this URL from left to right, it would translate to: "Go to the host computer called zdpress (a commercial business), in a directory called internetworks, and retrieve a hypertext document with the filename index.html." The URL, or address, tells the browser what document to fetch and exactly where to find it on a specific remote host computer somewhere on the Internet.

The first part of the URL indicates what type of transfer protocol will be used to retrieve the specified document. The most common request is for a hypertext document that uses HTTP (Hypertext Transfer Protocol).

The second portion of the URL refers to the specific host computer on which the document resides, which is to be contacted by the browser software. This part of the address is also called the *domain name.* See Chapters 4, "Understanding Internet Addresses and Domains," and 6, "Internet File Types," for more information about domains.

The third part of the URL is the directory on the host computer that contains a specific Web site or multiple Web sites. This is always located after the first single slash in the URL and is essentially the subdirectory on the hard disk that houses the Web site. Subdirectories might also be indicated in this part of the address. For example, if the above URL were changed to

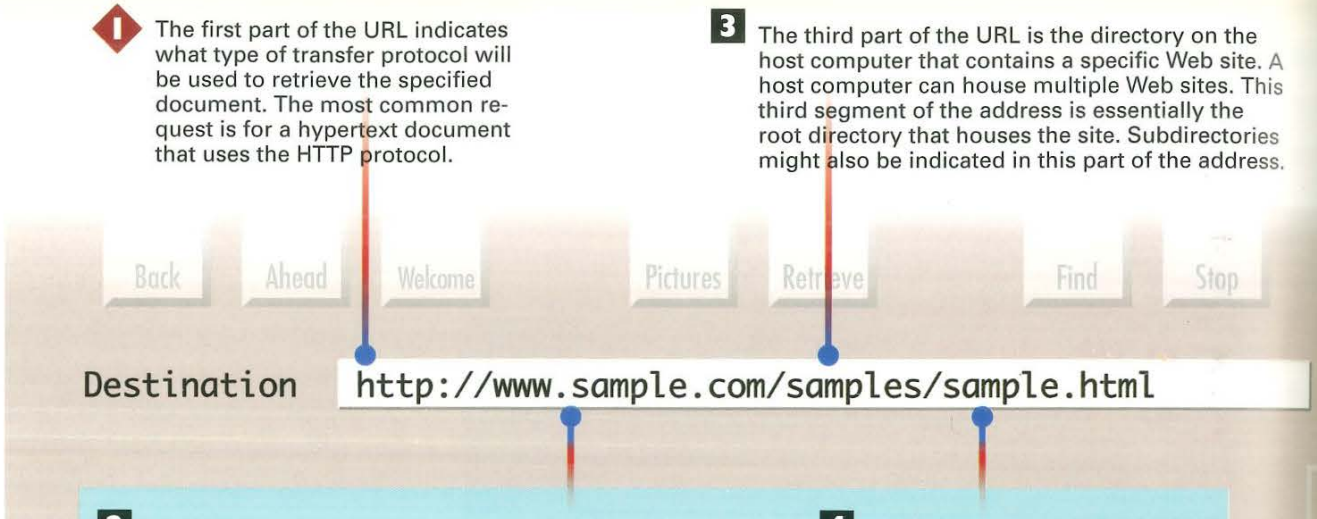
```
http://www.zdpress.com/internetworks/partone/chapters/chapte.html
```

there would be two subdirectories—part one and chapters.

In the above example, the filename is chapte.html. This is always the last portion of the URL. If you see an address without a filename, it is assumed that the filename index.html contains the requested Web page. The default document a Web server will deliver to the client when no other filename is listed is index.html.

The illustration in this chapter shows the process necessary to request and retrieve a Web document. When a request for a document occurs for the first time in a Web-browsing session, the host computer must first be located in order to find the file. After that, the specific subdirectory and document are retrieved.

The Structure of a Web URL



1 The first part of the URL indicates what type of transfer protocol will be used to retrieve the specified document. The most common request is for a hypertext document that uses the HTTP protocol.

2 The second portion of the URL is the specific host computer on which the document resides, which is to be contacted by the browser software. This part of the address is also called the domain. Domain names end in a suffix that indicates what kind of organization the domain is. For example, .com indicates a commercial business, .edu indicates a college or university, .gov indicates a government office, .mil a military facility, and .org a not-for-profit organization. The suffix can also indicate the country in which the host computer is located. For example, .ca is in Canada and .au is in Australia.

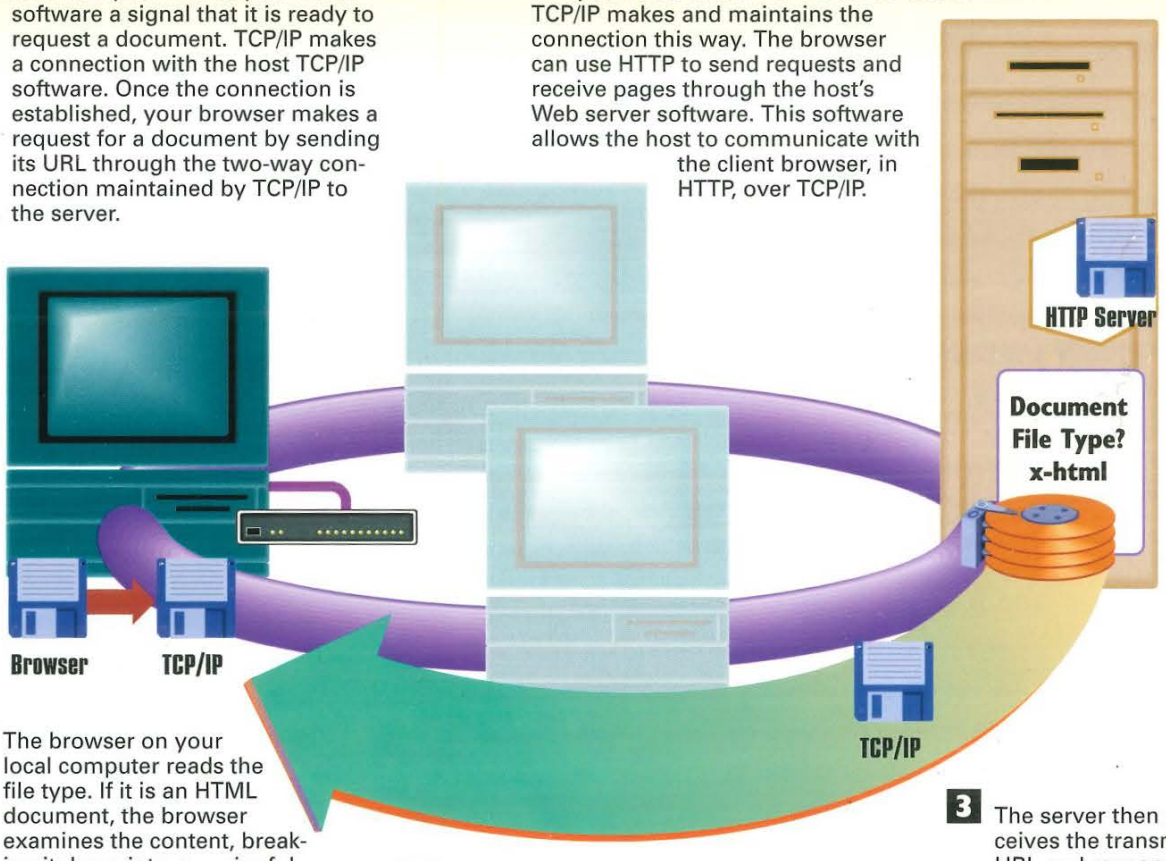
3 The third part of the URL is the directory on the host computer that contains a specific Web site. A host computer can house multiple Web sites. This third segment of the address is essentially the root directory that houses the site. Subdirectories might also be indicated in this part of the address.

4 The last segment of the URL is the filename of the specific Web page you are requesting. If no filename is indicated, the browser will assume a default page, usually called index.html.

How URLs Help Retrieve Web Documents

1 The Web browser installed on your local computer sends your TCP/IP software a signal that it is ready to request a document. TCP/IP makes a connection with the host TCP/IP software. Once the connection is established, your browser makes a request for a document by sending its URL through the two-way connection maintained by TCP/IP to the server.

2 The HTTP server is the portion of the host computer that runs HTTP server software. TCP/IP makes and maintains the connection this way. The browser can use HTTP to send requests and receive pages through the host's Web server software. This software allows the host to communicate with the client browser, in HTTP, over TCP/IP.



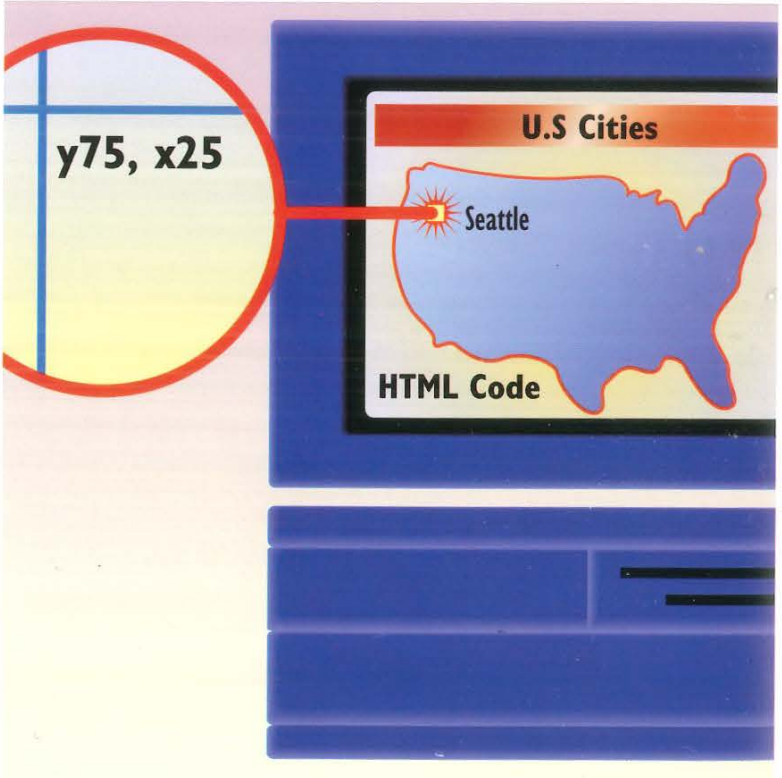
3 The server then receives the transmitted URL and responds in one of three ways. It follows the directory path given in the URL; the server finds the file on its local hard disk and opens it; the server runs a CGI script or it detects an error (such as "file not found") and generates an error document to be sent back to the client.

4 If the document is found, the host checks its file type (usually either x-html or x-text) and sends this information to the client with the requested page. When the client receives the page, it first checks the file type. If the type is one it can display, it does so; otherwise, it prompts the user to see if he or she would like to save it to disk or open it using a helper application. The x-html file type is by far the most common one used when transmitting Web pages.

5 The browser on your local computer reads the file type. If it is an HTML document, the browser examines the content, breaking it down into meaningful parts. Two general parts include text, which is displayed by the browser word for word; the other part consists of HTML markup information called *tags* that are not displayed, but display formatting information, such as normal text, bold headers, or colored hypertext. The results are displayed on your monitor.

CHAPTER

30 How Imagemaps and Interactive Forms Work



GRAPHICS called imagemaps and functions called interactive forms demonstrate two of the more common and helpful uses of HTML. *Imagemaps* are static images that have been turned into clickable images with various clickable parts. *Interactive forms* are HTML-based pages that you fill out by providing information such as your name, email address, and similar information. Both imagemaps and interactive forms are created using Common Gateway Interface, or CGI, a communications protocol by which a Web server can communicate with other applications. (For more information about how CGI works, turn to Chapter 36, "How CGI Scripting Works.")

Imagemaps may be thought of as fancy hyperlinks. However, instead of a word or entire icon or image linking to another page, an image is divided into different segments, or coordinates, that link to different HTML pages. That is, imagemaps link to another document through a predefined "hot" area within an image. As soon as you click your mouse on a hot spot, a CGI script and special imagemap coordinates file with the suffix .map go to work. A CGI application reads the map file to match coordinates of a mouse click with a corresponding URL. For instance, imagine an electronic map of the United States in which you click on Washington, DC. In the HTML code for that page the electronic map is surrounded by a tag and an attribute called ISMAP. The code looks something like this:

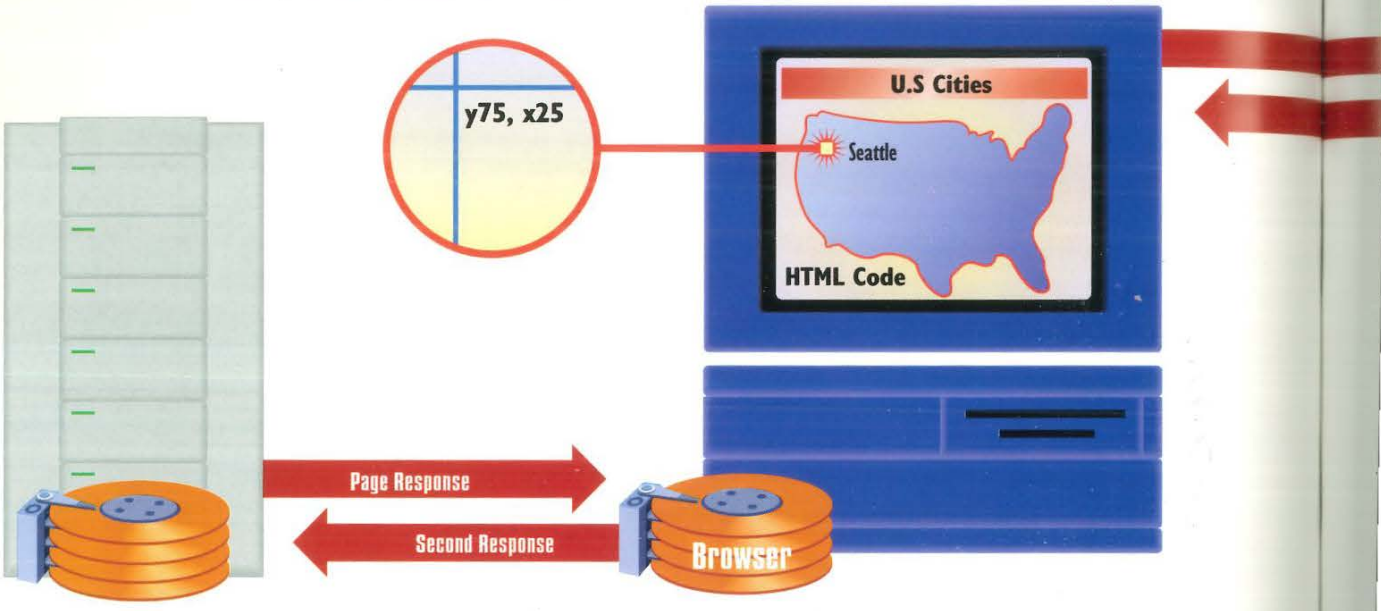
```
<A HREF="some.server/maps/clickable.map>
<IMG SRC+"clickable.map" ISMAP>
</A>
```

The x and y coordinates of your mouse click are sent to the server. The coordinates are received by the server and then redirected to a CGI application. The CGI application scans the file for matching coordinates, then forwards the corresponding URL to the server. Lastly, if the Web page resides on the same server, it will deliver that Web page to the client browser. If not, the server returns the URL to the client browser, which in turn sends a request to the correct server for the page. You then see the page about Washington, D.C. begin to load on your browser. Behind the scenes, the server had passed your mouse click coordinates to a CGI application via the CGI. Then the CGI application matched those coordinates to its URL in a .map file. Finally, the URL sent the URL back to the server, which redirected the client browser to the new Web page.

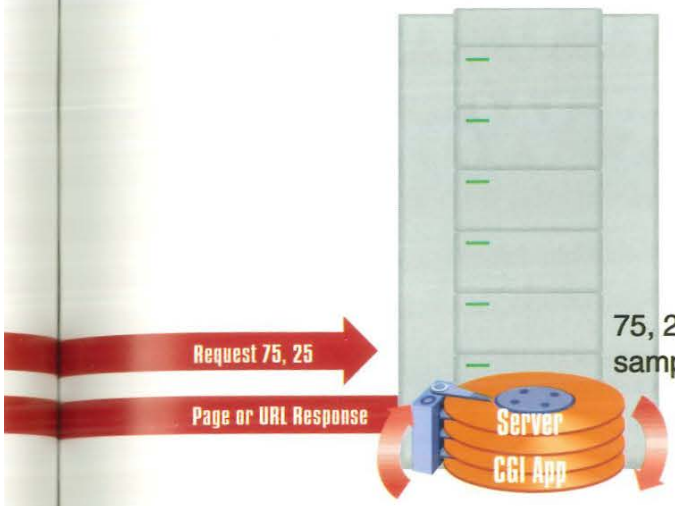
Forms work differently, although they also use CGI. In a form, when you fill in information on a Web page, that information goes to the server for processing. Next, the server redirects the information to a CGI application that is called by the form "submit." (CGI scripts are activated by the server in response to an HTTP request from the client.) Lastly, a CGI application may send form data to another computer program, such as a data-

How Imagemaps Work

1 In this map example, the user clicks on Seattle. The coordinates are 75, 25. In the HTML code, the browser recognizes the ISMAP image tag attribute. The mouse click activates the browser to send the x and y coordinates of the click to the server. The location of the "National.map" file is also sent to the server.



4 The client browser either displays the returned page or (based on the returned URL) sends a request to the correct server for the page.

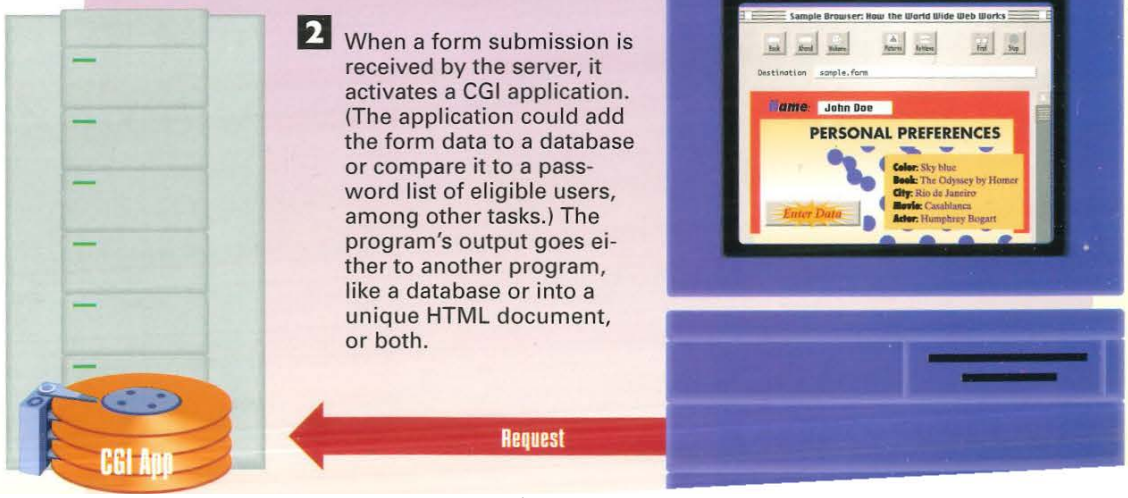


2 The server hands off the coordinate and map file data to a CGI application. The CGI application matches the coordinates to the URL that has been requested by the user by clicking on that portion of the map. This URL is handed back to the server and the server sends the page to the client.

3 The Web document is either served up (if it resides on the same server) or the client browser is forwarded the new URL.

How Interactive Forms Work

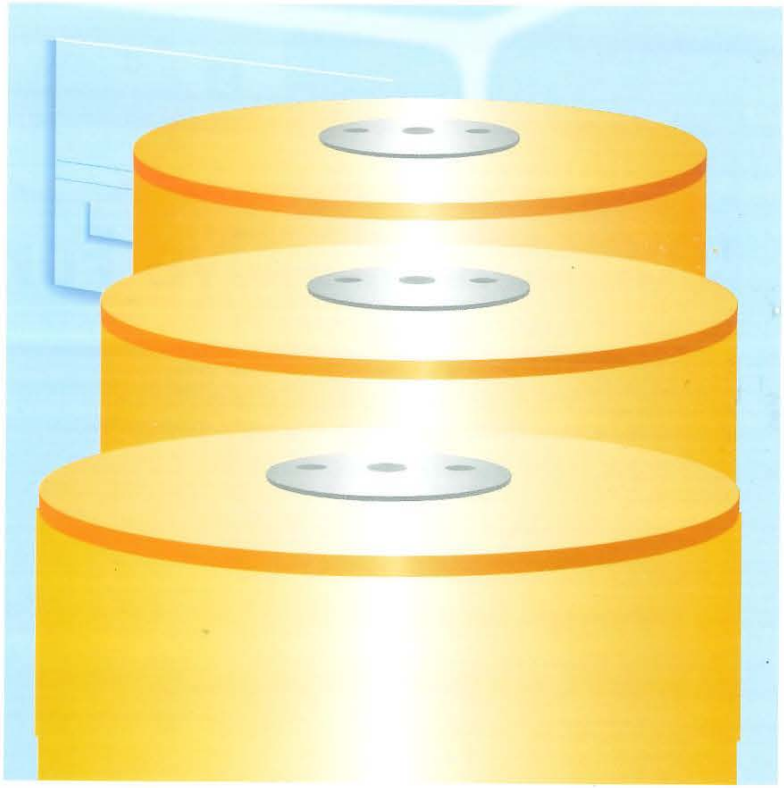
1 In working with an interactive form, the user clicks a data entry submit button. The data in the data fields is sent to the server with the request.



2 When a form submission is received by the server, it activates a CGI application. (The application could add the form data to a database or compare it to a password list of eligible users, among other tasks.) The program's output goes either to another program, like a database or into a unique HTML document, or both.

CHAPTER 31

How Web Host Servers Work



TO serve up pages, Web sites need a host—a computer—and server software that runs on the host. The host manages the communications protocols and houses the pages and the related software required to create a Web site on the Internet. The host machine often uses the UNIX, Windows NT, or Macintosh operating systems, which have the TCP/IP protocols built in.

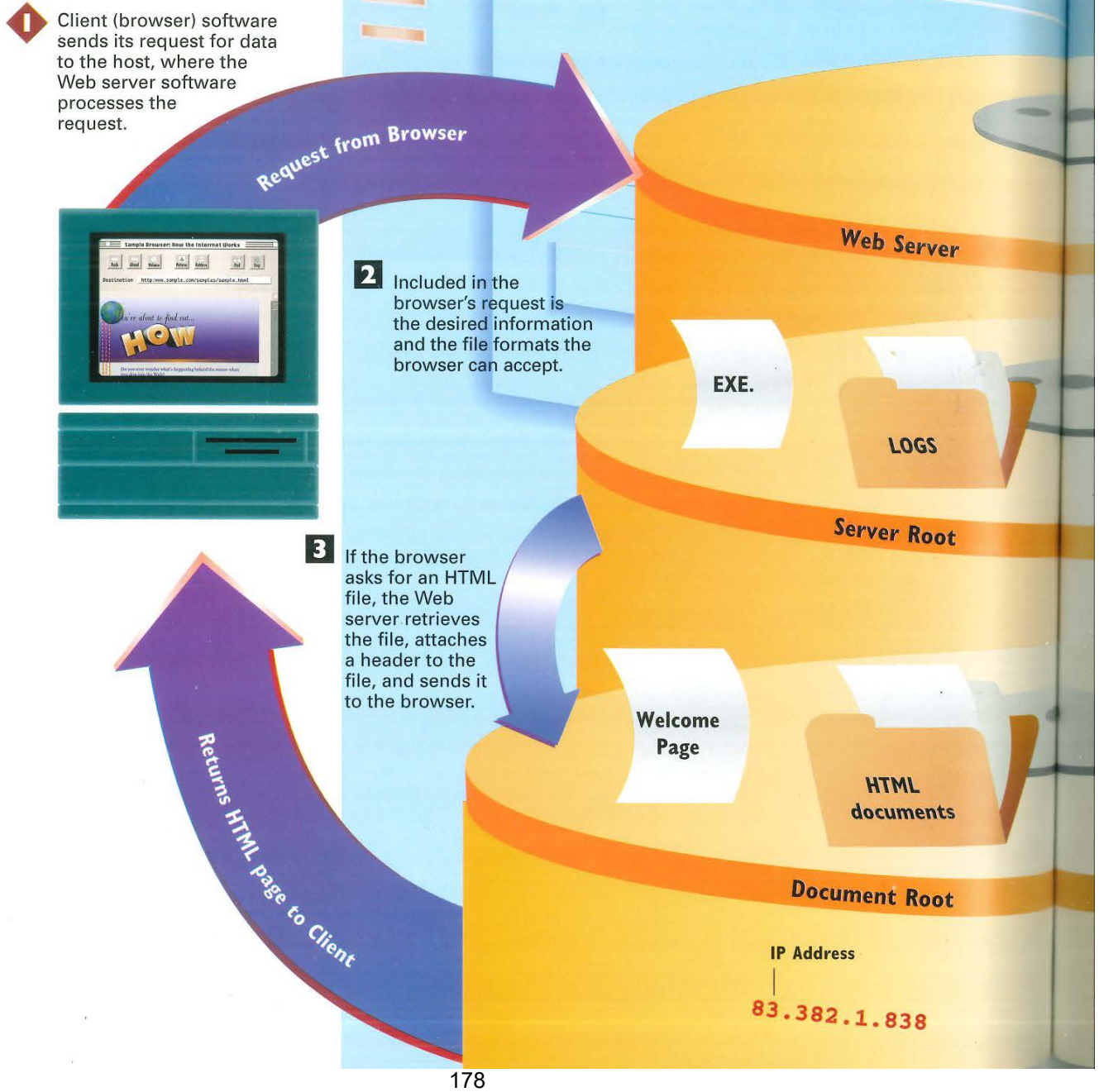
The server software resides on the host and serves up the pages and otherwise acts on the requests sent from the client browser software. The server is not responsible for TCP/IP communications—the host operating system does that—but instead the server handles the HTTP requests and communications with the host operating system.

There are different types of server software (database servers or network servers, for example) that perform different types of services for different types of clients. Specifically, a Web server is an HTTP server and its function is to send information to the client software (typically a browser) using the Hypertext Transfer Protocol.

Usually, the client browser requests that the server return an HTML document. The server receives this request and sends back a response. The top portion of the response includes transmission information and the rest of the response is the HTML file.

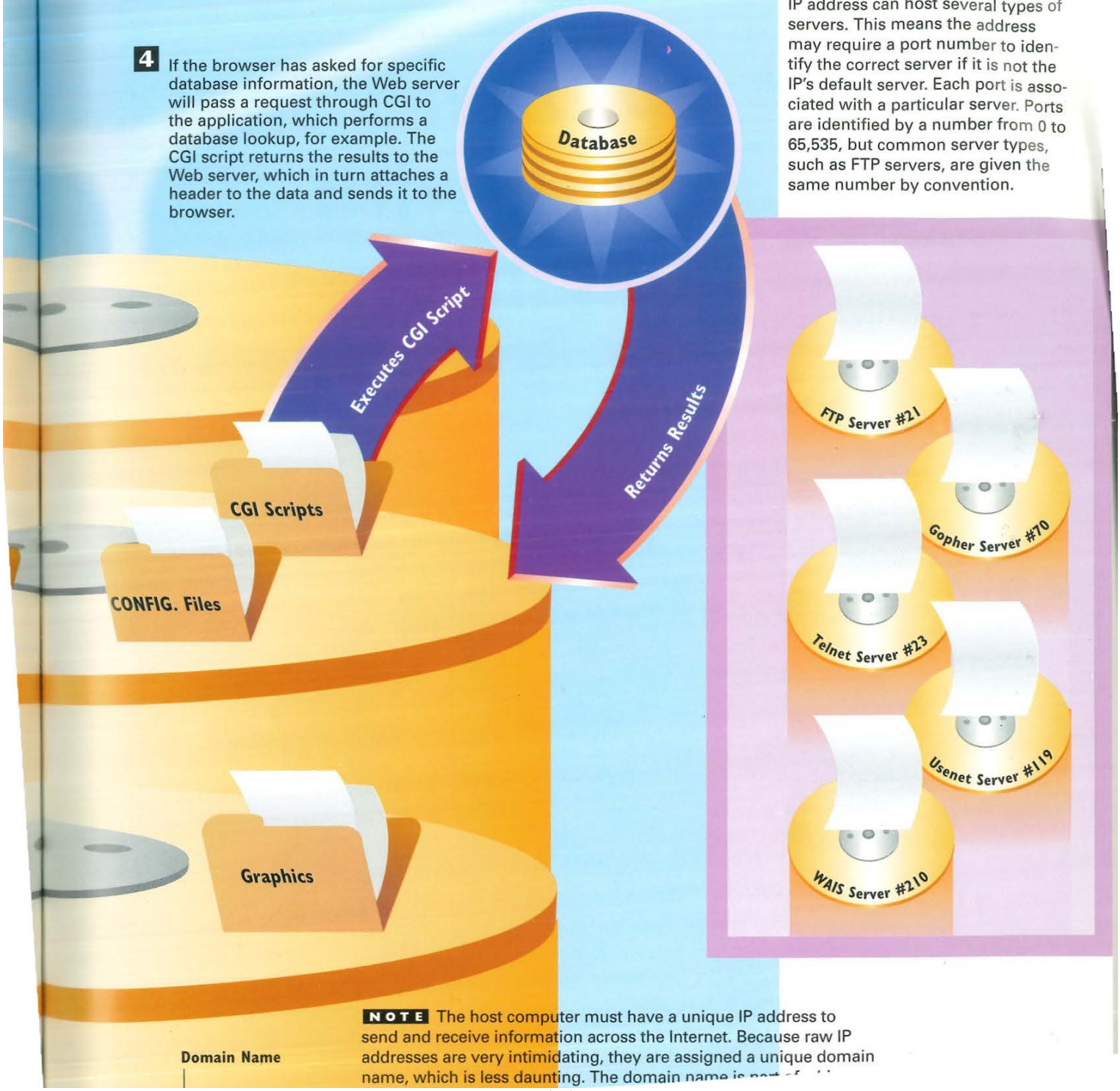
A Web server does more than send pages to the browser, however. It passes requests to run CGI (Common Gateway Interface) scripts to the CGI applications. These scripts run external mini-programs, such as a database lookup or interactive forms processing. The server sends the script to the application via CGI and communicates the results of the script back to the browser, if appropriate. Moreover, the server software includes configuration files and utilities to secure and manage the Web site in a variety of ways.

How Web Server Software Works



4 If the browser has asked for specific database information, the Web server will pass a request through CGI to the application, which performs a database lookup, for example. The CGI script returns the results to the Web server, which in turn attaches a header to the data and sends it to the browser.

NOTE A computer with a single IP address can host several types of servers. This means the address may require a port number to identify the correct server if it is not the IP's default server. Each port is associated with a particular server. Ports are identified by a number from 0 to 65,535, but common server types, such as FTP servers, are given the same number by convention.

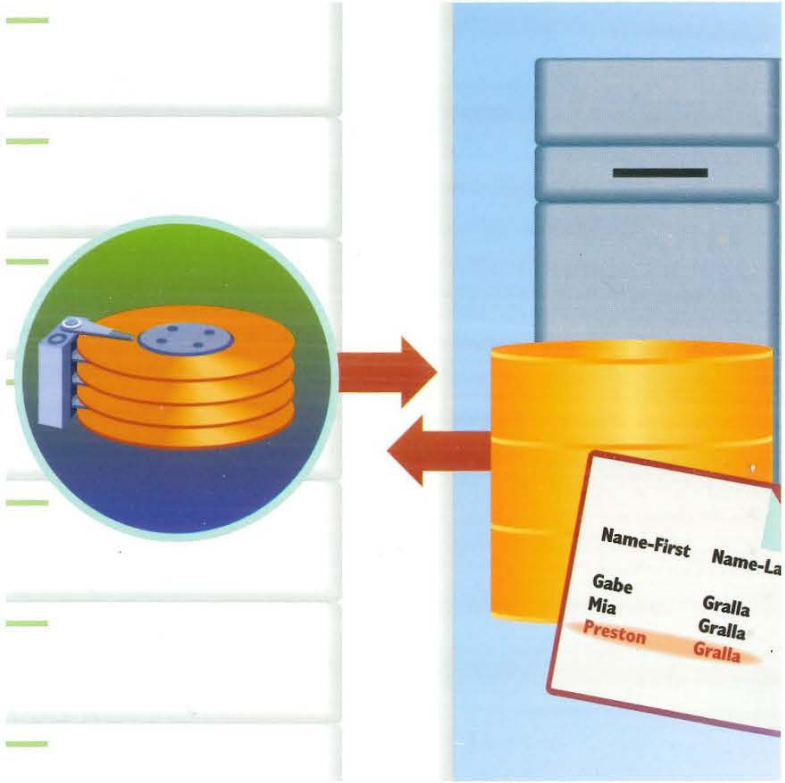


NOTE The host computer must have a unique IP address to send and receive information across the Internet. Because raw IP addresses are very intimidating, they are assigned a unique domain name, which is less daunting. The domain name is part of

Domain Name

CHAPTER

32 How Web Sites Work with Databases



ONE of the most useful applications of the Web is its capability to link a Web site with a database so Web surfers can search for information. In essence, the Web page becomes the front-end for database applications, enabling you to select search criteria and execute even complex searches of a database that resides on the host computer.

A well-known and widely used example of this type of linking between Web sites and databases is the popular Yahoo! Web site. The Yahoo! site serves as a front-end to an extensive database of Web site descriptions, which can be searched according to keywords. The Welcome page includes a search dialog box in which you enter a keyword that represents the subject matter you are looking for. Selecting "Search" from the page sends a request from the browser to the Web server to bring back a list of all Web sites that contain your keyword.

Furthermore, not only can the Web serve up data, it can also collect it. For example, many Web sites ask users to "register" their names, addresses, and other demographic information that is captured and stored in a database.

But how does this all work? You don't have to be a corporate giant—or for that matter even an able programmer—to link your Web site to a database. In fact, linking a Web site to a database can be relatively simple. The database can take just about any form and can be as simple as a FileMaker Pro database, or as complex as an Oracle SQL database. The bridge that brings together Web sites and databases is the Common Gateway Interface (CGI).

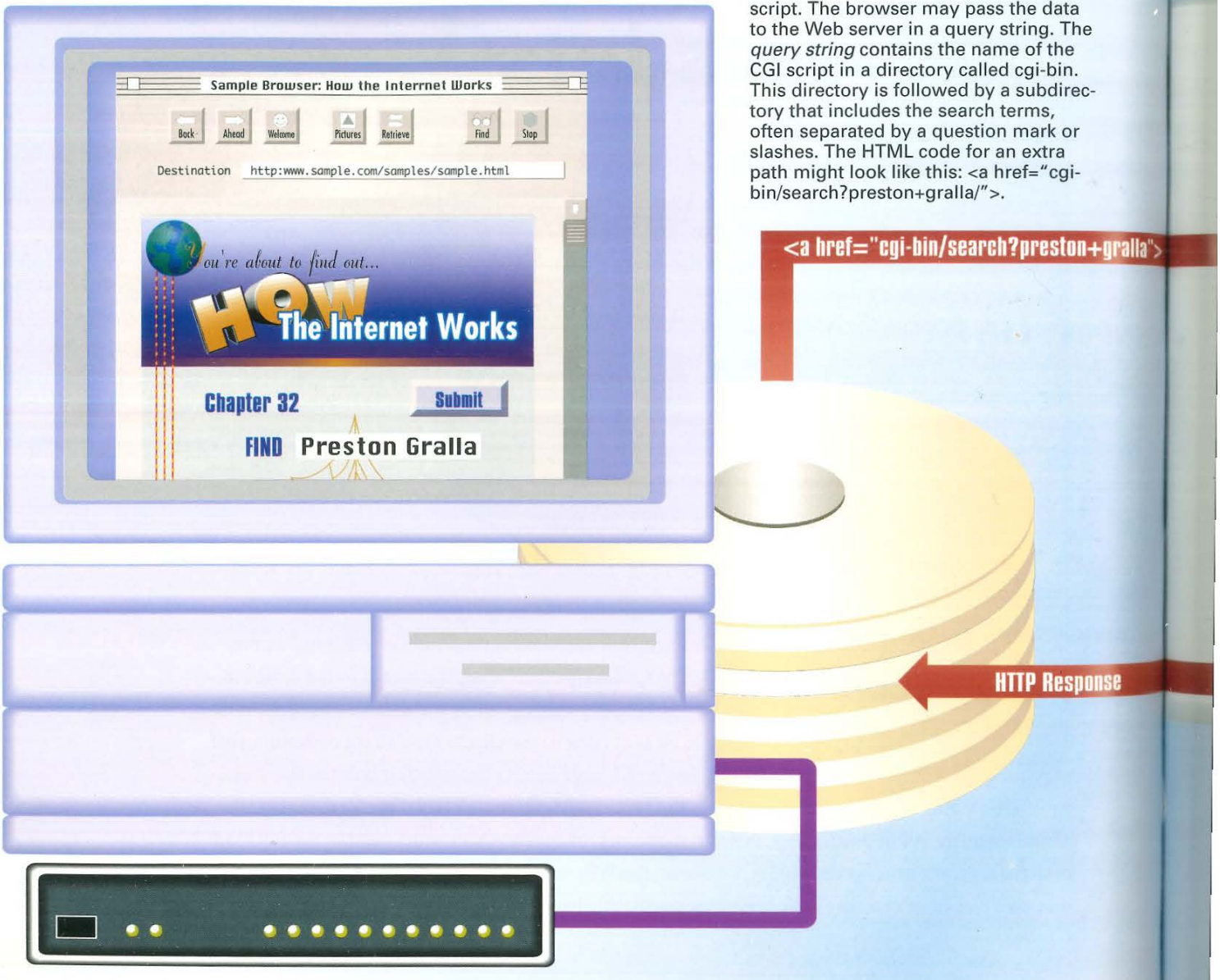
On the client side of the database, you see a Web page that includes a form in which you enter your search terms. By executing the search, you launch a CGI script that sends a search command to the Web server in the form of a link to the CGI bin on the Web server. So a search on the Yahoo! site for public relations firms looks like this:

```
http://search.yahoo.com/bin/search?p=public+relations
```

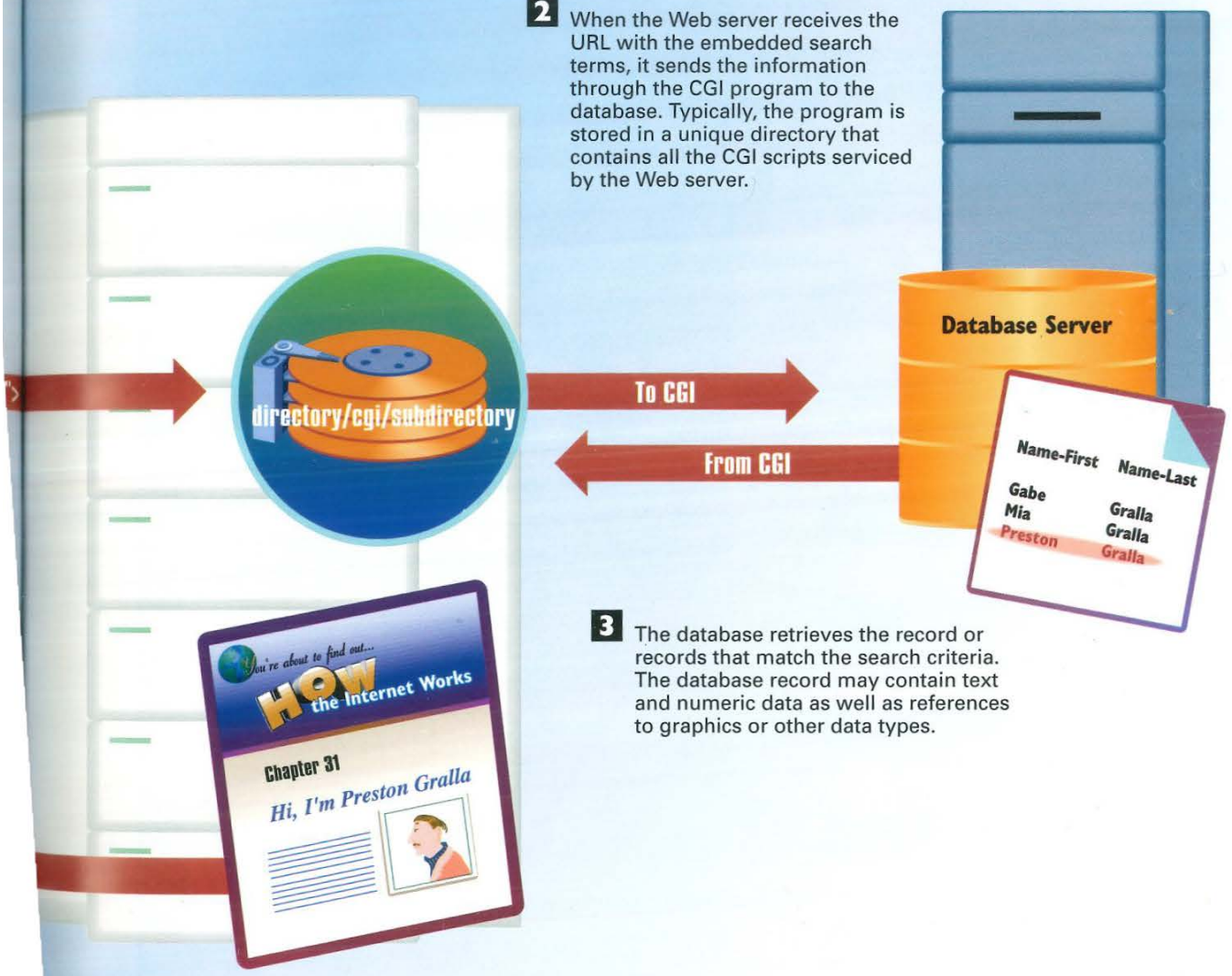
When the Web server receives this URL, it identifies the URL as a trigger for a CGI script (called "script" in this example) and passes it along with the search criteria ("public relations," in this example) to the mini-program using CGI. The CGI script then sends the search to the database, receives the results of the query along with the HTML page created by the database to contain the result, and passes it on to the Web server to be sent back to the client. That's a lot of handing off of requests and data, but typically even a search of a large database is very fast because the majority of UNIX and Windows NT databases—the types most often used—can perform these tasks simultaneously. All of this happens behind the scenes, of course—you won't need to do any kind of database work or scripting yourself. Instead, the Web sites you visit have easy-to-use interfaces that take care of interacting with databases; you'll only have to type in what you're looking for.

How the Web Works with Databases

1 The search begins on a Web page that includes a form field to accept search terms and HTML codes to execute a CGI script. The browser may pass the data to the Web server in a query string. The *query string* contains the name of the CGI script in a directory called cgi-bin. This directory is followed by a subdirectory that includes the search terms, often separated by a question mark or slashes. The HTML code for an extra path might look like this: <a href="cgi-bin/search?preston+gralla/">.

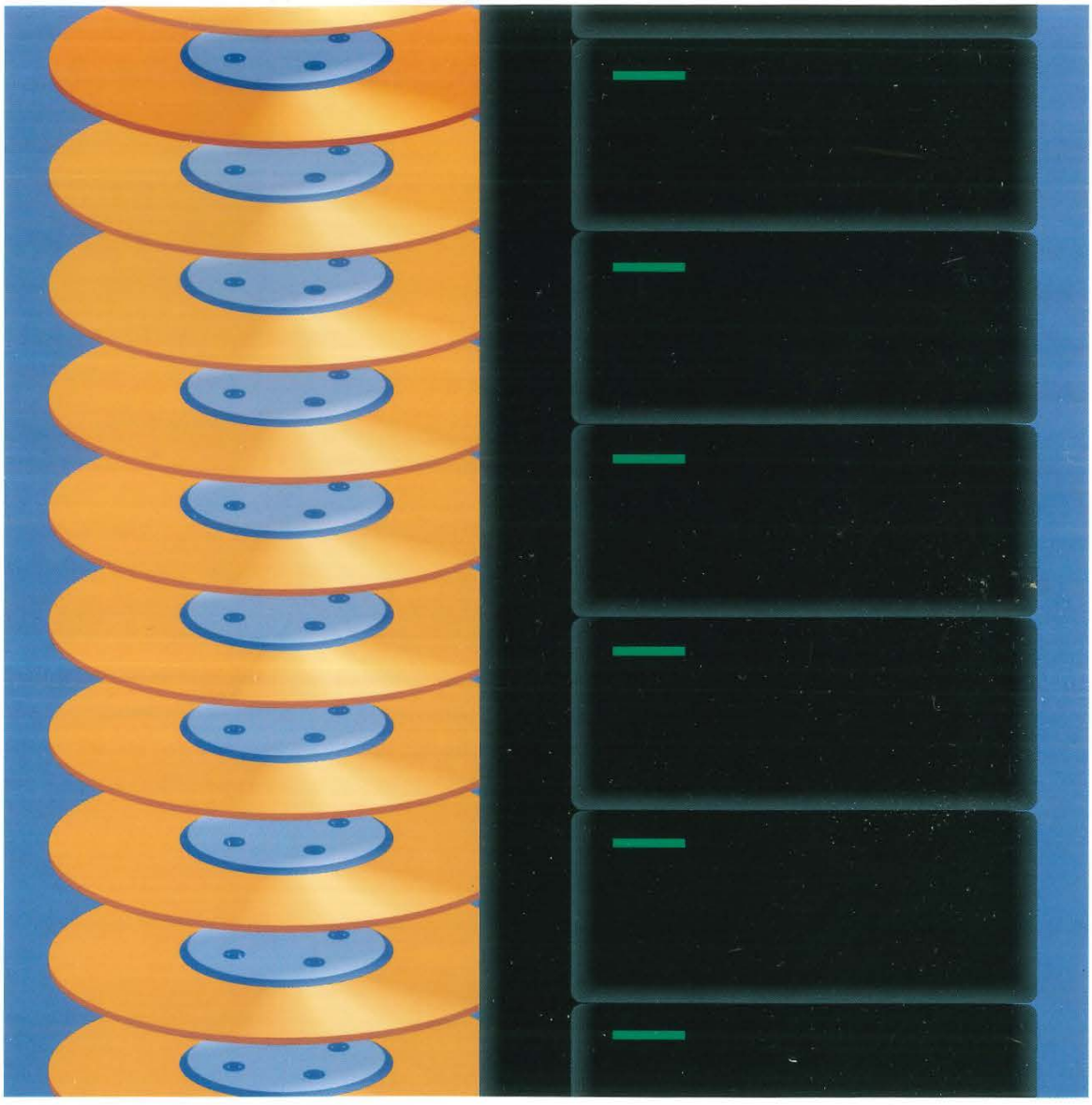


2 When the Web server receives the URL with the embedded search terms, it sends the information through the CGI program to the database. Typically, the program is stored in a unique directory that contains all the CGI scripts serviced by the Web server.



3 The database retrieves the record or records that match the search criteria. The database record may contain text and numeric data as well as references to graphics or other data types.

4 The database returns the data to the Web server via CGI in the form of a new HTML page. The server then sends the page back to the client browser as a new HTML page.



PART 7

ADVANCED INTERNET TOOLS

Chapter 33: How Push Technology Works
188

Chapter 34: How Java, ActiveX, and JavaScript Work
196

Chapter 35: How Agents Work
204

Chapter 36: How CGI Scripting Works
208

THE constant evolution of the Internet makes it more exciting than other communications media such as television or radio. Existing technologies are being improved and new ones are being developed every day that make the Internet a richer, more interactive, more entertaining, and more productive medium.

New technologies appear so fast on the Internet that people who work in Internet-related companies often have their own new way to measure time in "Web years" or "Internet years." Because technologies evolve so quickly on the Web, the amount of change that happens on the Internet in two months might equal the kinds of changes that happen over a year in the outside world. So a Web year might be two months in real-time.

Most of these changes occur in various advanced Internet technologies. Commonly, an idea for a new technology is hatched in a small startup company, or even by an individual. It spreads from there, usually by way of the Internet, and in a very short time what was once a mere dream becomes an Internet standard.



In this section, we'll look at some of these advanced Internet technologies. Many of them have changed the very nature of the Internet and have turned it into a truly interactive medium—one that people can navigate efficiently. The technologies also enable Web publishers and Internet developers to more effectively present information to people.

Chapter 33, "How Push Technology Works," looks at one of the newest technologies and one that may have enduring effects; it's known as *push*. In push technology, entire Web sites, customized information, and even applications can be sent straight to people's computers without them having to go out and ask for the information. In push technology, you *subscribe* to Web sites or channels of information, and they are then sent to you at intervals you specify. Subscribing doesn't mean you have to pay for information. It's the term that's used to say you've asked to receive information on a regular schedule.

No one standard for push technology exists—many companies have competing technologies and products. For example, Netscape has its own standard for push technology, called Netcasting, and Microsoft's standard is called Webcasting. Meanwhile, many other companies, notably PointCast, have their own methods of pushing information straight to people's computers. Chapter 33 looks at the most important underlying technologies that make push possible.

Chapter 34, "How Java, ActiveX, and JavaScript Work," examines three other kinds of technologies that are transforming the Internet—Java, JavaScript, and ActiveX. These three technologies may do more to transform the Internet than almost any other technologies currently available. These technologies add multimedia and interactivity, but more importantly, they begin to treat the Internet as if it were an extension of your computer. In essence, they enable your

computer and the Internet to interact as if they were one large computer system. This allows for things such as news tickers, interactive games you can play with others, multimedia presentations combining animations, sounds, music, graphics, and much more.

Java, a computer language developed by Sun Microsystems, enables applications to be run from the Internet. The programs run inside your Web browser. One benefit of Java applications is that they can be run on any computer, such as a PC, a Macintosh, or a UNIX workstation.

ActiveX, a competing technology from Microsoft, can also essentially turn the Internet into an extension of your computer. Like Java applets, ActiveX controls are downloaded to your computer and run there. They can do anything that a normal application can do, and can also interact with the Web, the Internet, and other computers connected to the Internet. To run them, a browser that supports ActiveX, such as Internet Explorer, is needed.

JavaScript, which despite its name is not really related to Java, is simpler than Java and ActiveX and can be written by people who don't have substantial programming experience. JavaScript is commonly used to create interactive forms, site navigation, and similar features.

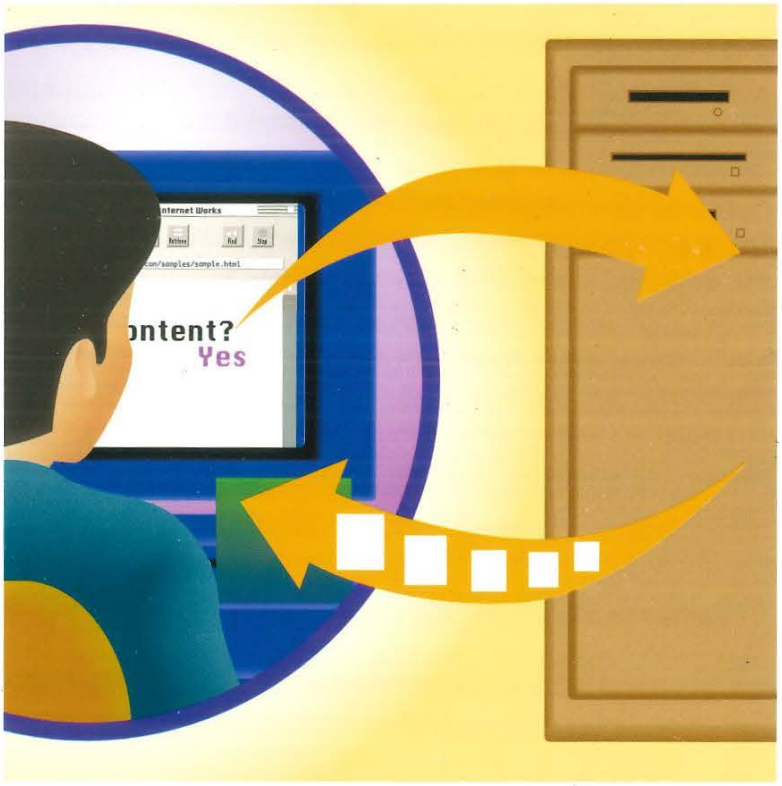


In Chapter 35, "How Agents Work," we'll look at agents on the Internet. *Agents* are programs that do your bidding across the Internet automatically, without your having to do anything. They can find the latest news and download it to your computer; they can find you the best deal on the CD you want to buy; they can perform important Web maintenance tasks; and more. They are becoming so complex that systems are being developed to allow agents to interact with one another so that they can perform jobs cooperatively.

Finally, in Chapter 36, "How CGI Scripting Works," we'll examine *CGI* (Common Gateway Interface) *scripting*. This may appear as one of the more mundane Internet technologies, but without it, very little Web interactivity would take place. CGI is a standard way in which the Web interacts with outside resources—most commonly, databases. You've probably run CGI scripts many times without knowing it. If you've filled out a form on a Web page to register to use a site and then later received an email notification with a password for you to use, you've probably run a CGI script. CGI enables programmers to write code that can access information servers (such as Web servers) on the Internet and then send the information to users.

CHAPTER 33

How Push Technology Works



INCREASINGLY, the Internet is becoming a broadcast medium. It relies less each day on people taking it upon themselves to visit certain sites. Instead, information, entire Web sites, and applications can be sent via the Internet straight to people's computers with a variety of technologies collectively known as *push.*

In push technology, you "subscribe" to Web sites, often called channels. A *channel* generally refers to an area of interest that a publisher builds, which can include HTML pages, Java applets, ActiveX components, multimedia objects, and other information packaged together to deliver customized information to people via push technology. Those sites you subscribe to and the information they contain are sent to your computer automatically at intervals that you specify. You don't pay for these subscriptions; you merely ask to receive the desired information on a regular schedule.

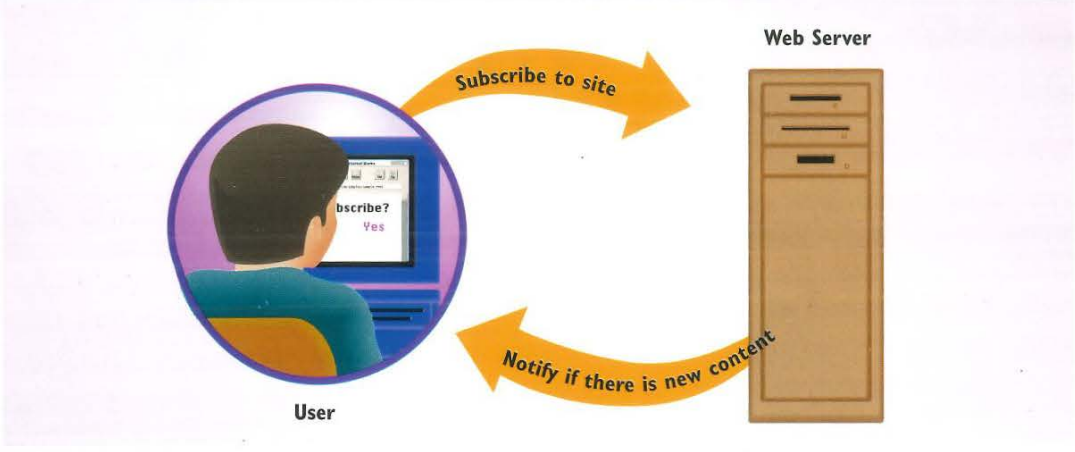
Receiving these channels requires either special client software or a push-enabled browser on your desktop. This software is generally free. Push technology is built into browsers such as Netscape Communicator and Microsoft Internet Explorer. Push client software is available from companies such as PointCast and BackWeb. Most push technologies also allow you to customize the kind of information that you receive.

This kind of customization is one benefit that push technologies offer. Another benefit is the time savings—instead of you having to go out and gather the information, it's delivered right to you with no effort on your part, and it's precisely the kind of information that interests you. There's one more benefit as well: When the information is sent to you, it resides locally on your computer. So you can read it whenever you want—you don't have to be connected to the Internet. You'll also be able to view the pages and information more quickly than if they were on the Internet instead of your local computer because you're reading it at high speed from your hard disk instead of across the slow wires of the Internet.

There is no single standard for push technology. Netscape has established its own standard and software, called Netcaster, which is part of its Communicator package. Netscape refers to its version of push technology as Netcasting. Microsoft has its own version of push technology built into Internet Explorer 4.0 and above, which it refers to as Webcasting. Aside from Netscape and Microsoft there are other push technologies as well, from companies such as PointCast, Marimba, and BackWeb.

Most push technologies have a few basics in common. For example, most allow you to customize how you'll receive your information. To receive that information, you'll have to be connected to the Internet, so many push technologies will also dial into the Internet for you to get updates. They deliver the information to your computer so you can read the information locally at your leisure. Also, most use similar terminologies, such as the word, "channel."

How Internet Explorer's Webcasting Works

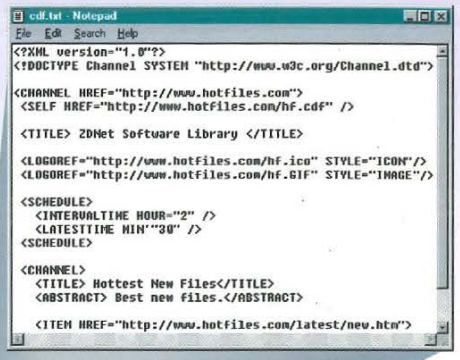


1 Internet Explorer 4.0 offers a variety of ways to use push technology and offline reading—what Microsoft calls Webcasting. In the simplest form of Webcasting, you "subscribe" to a Web site and ask to be told when the content on the site changes. Your browser will regularly visit that site and notify you if new content is available. It won't, however, automatically download any of that content.



2 In a slightly more sophisticated version of Webcasting, you subscribe to a Web site and ask that whenever there is new content, it be automatically delivered to you. In this instance, your browser regularly contacts the Web site, "crawls" the pages by looking at every page, and downloads to your computer only those pages that have changed since your last visit. You'll

3 A much more sophisticated form of Webcasting uses channels. A channel is designed from the ground up to be delivered via push technology, and incorporates multimedia, ActiveX, and other interactive kinds of content. To build a channel, developers create HTML files, ActiveX controls, and similar content, and post them on their Web pages. They then create a Channel Definition Format (CDF) file that describes the channel, how it functions, and where its content is on a Web server. This CDF file is linked to a button on a Web page.





Developer/Editor

4 When someone clicks on the button, the CDF file downloads to his or her computer.



5 The CDF file determines how the channel functions. It contains data such as how often to schedule updates, information about the structure of the channel (such as whether it has sub-channels), and links to specific Web pages and contents, such as ActiveX controls and multimedia content.

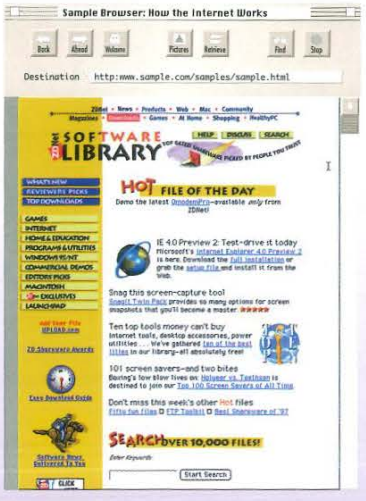
CDF file

6 The browser visits the Web site on a schedule determined in the CDF. Then it downloads the CDF file in case the file has changed, and also downloads the associated content referenced in the CDF file such as Web pages, ActiveX controls, and multimedia content.

CDF file

Get update

HTML Pages





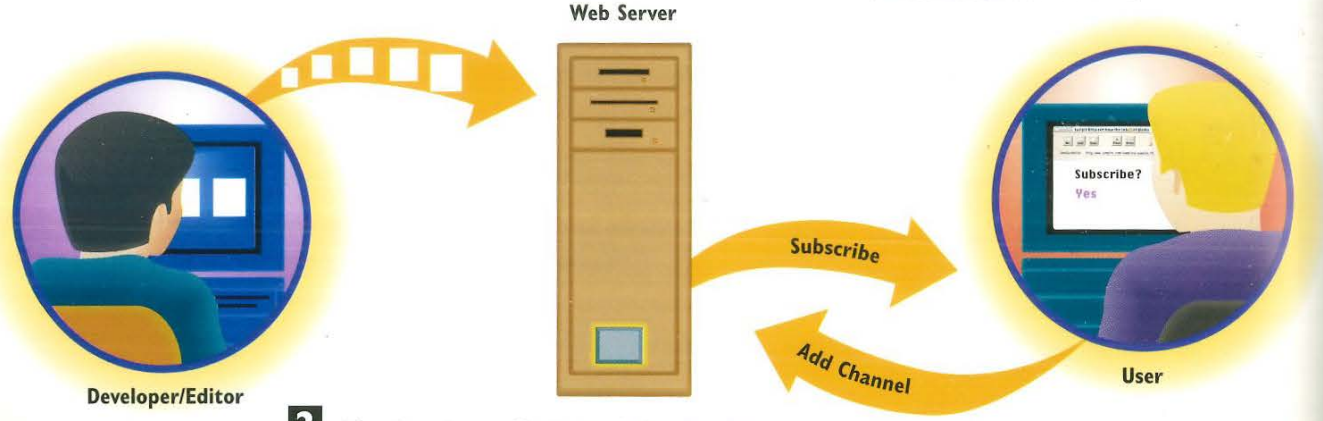
User

7 The channel pages and associated files are now on the local computer. Because the information and pages are local instead of residing on the Internet, they can be accessed while the user is not connected to the Internet. The pages and information will be accessed quickly because they are being read from a hard disk instead of across the Internet.

How Netscape's Netcasting Works

1 Netcaster is the part of Netscape Communicator that allows for information and sites to be pushed to someone's desktop, and for those sites to be browsed offline. The sites that can be pushed are called channels. Channels are built by developers using HTML and JavaScript and can also incorporate Java applets and multimedia content. Before a channel can be pushed, the developer posts all the content and channel information on a Web site, and sets default properties for that channel, such as how often Netcaster should automatically update it.

3 When the button is clicked on, an "addChannel" JavaScript call is made. A dialog box pops up, listing the properties of the channel (such as how often to update it). The user can change the properties or keep the defaults. After the user clicks on OK, the channel is added to Netcaster. The subscription to that channel is now in effect.

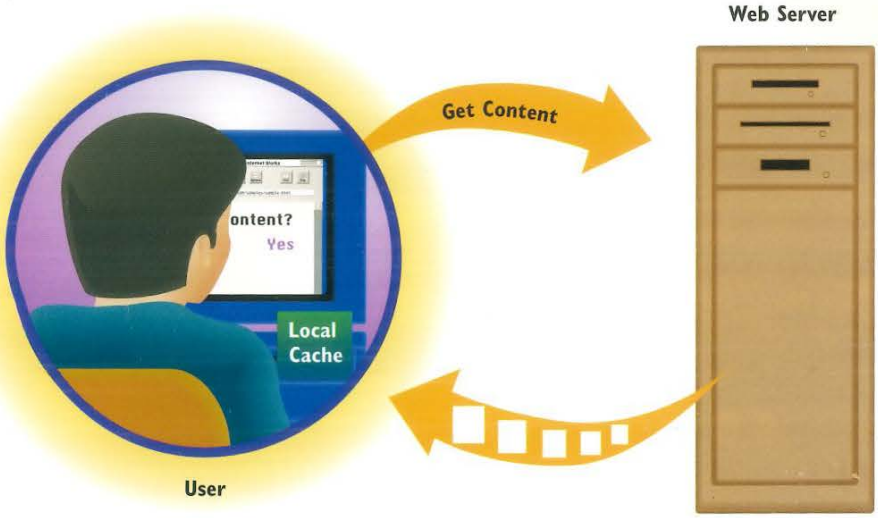


2 After the channel is created, the developer creates a button on a Web site that, when clicked, will "subscribe" people to the channel.

4 Through the Channel Properties box, the user can change many aspects of how to interact with the channel, such as how often to check the channel for new content, and how many levels of Web pages should be checked.



5 Netscaster connects to the channel at intervals the user specifies, or at intervals in the default information about the channel.



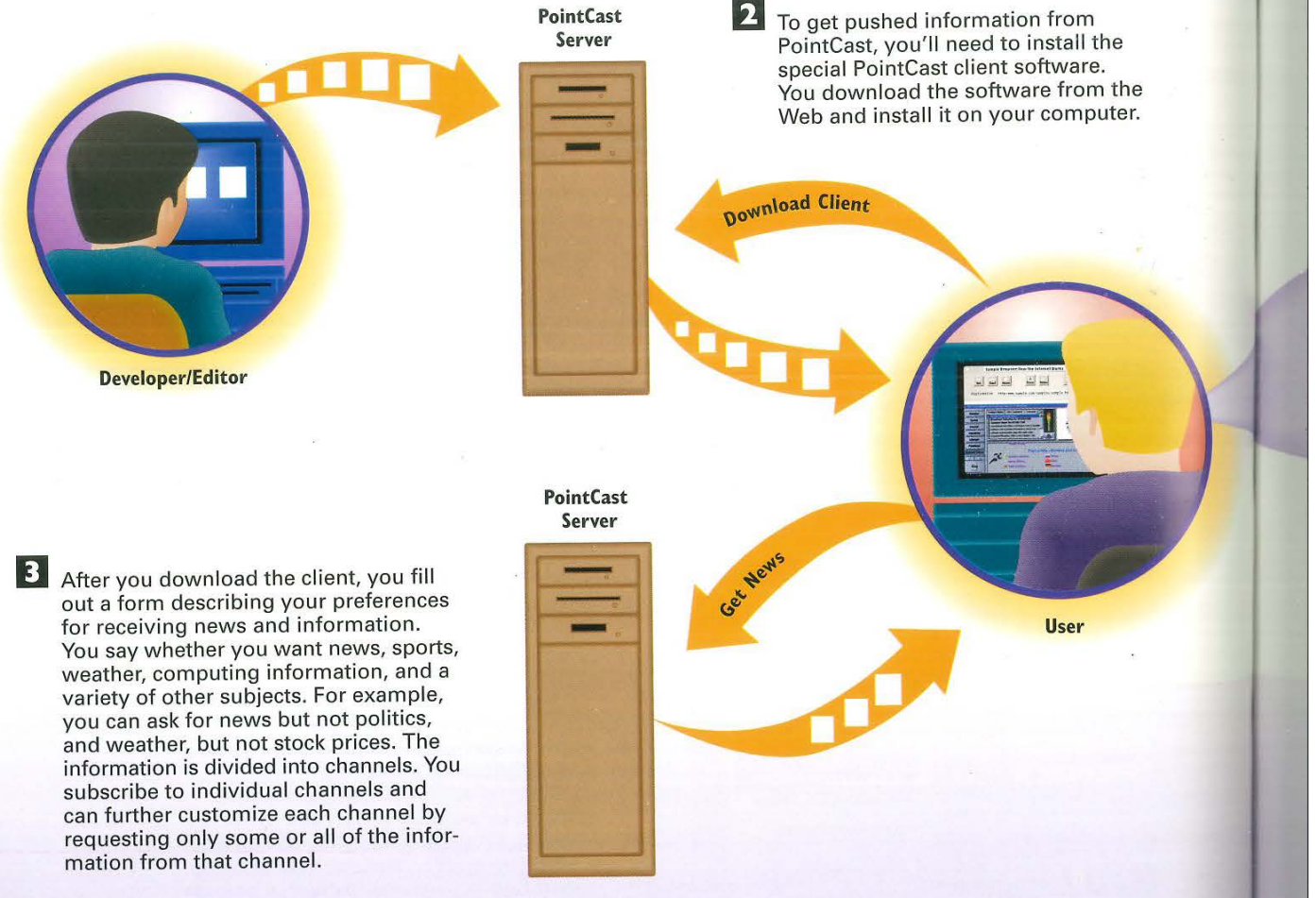
6 After Netcaster connects, it "crawls" the channel—that is, it downloads the information, data, and pages in the channel. First it downloads the HTML pages, then the associated graphics, then embedded objects such as Java applets or audio and video files. The pages, graphics, and embedded objects are all put into a cache on the local machine.



7 The content of the channel can be read from the cache. Because the information and pages are local instead of on the Internet they can be accessed when the user is not connected. They can also be accessed quickly because they're being read from a hard disk instead of across the Internet.

How PointCast Works

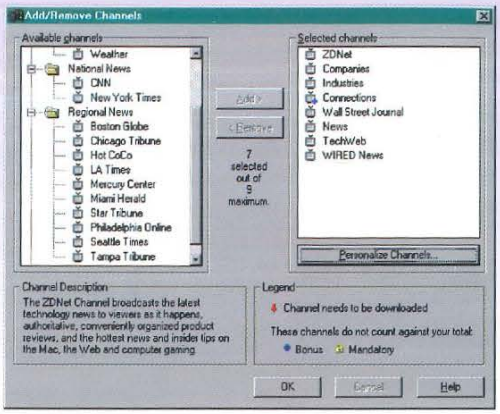
1 PointCast, the first push technology to gain widespread acceptance, delivers customized news and information to people's desktops. Publishers who want to deliver information via PointCast have to create special files and automated feeds that are placed on PointCast servers.



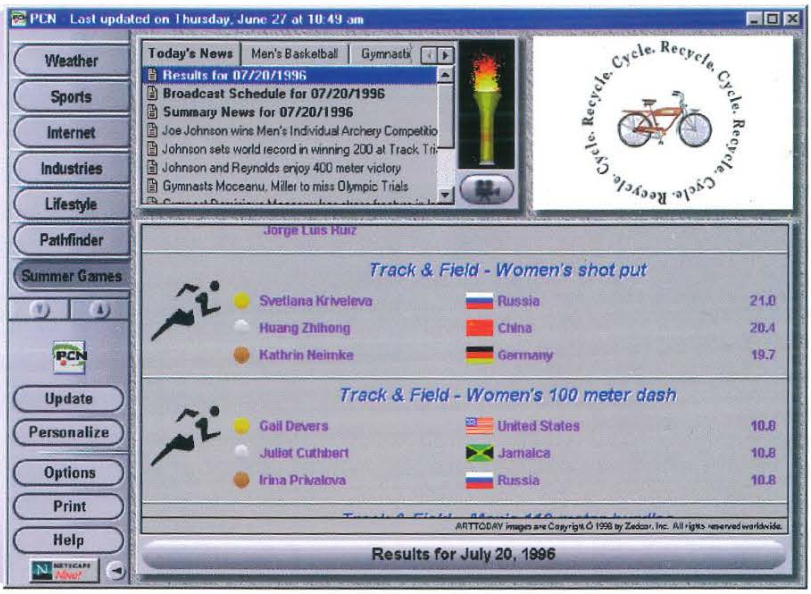
2 To get pushed information from PointCast, you'll need to install the special PointCast client software. You download the software from the Web and install it on your computer.

3 After you download the client, you fill out a form describing your preferences for receiving news and information. You say whether you want news, sports, weather, computing information, and a variety of other subjects. For example, you can ask for news but not politics, and weather, but not stock prices. The information is divided into channels. You subscribe to individual channels and can further customize each channel by requesting only some or all of the information from that channel.

4 At set times that you determine in your preferences, the PointCast client software automatically connects to the PointCast server over the Internet. The software downloads the customized news that you've requested. You can also manually tell the client software to download the news at any time you want.



5 Using the PointCast client, you can now read the news on your own computer. The software can also be set up as a screen saver so that when the computer is idle for a certain amount of time, the news will flash across your screen.

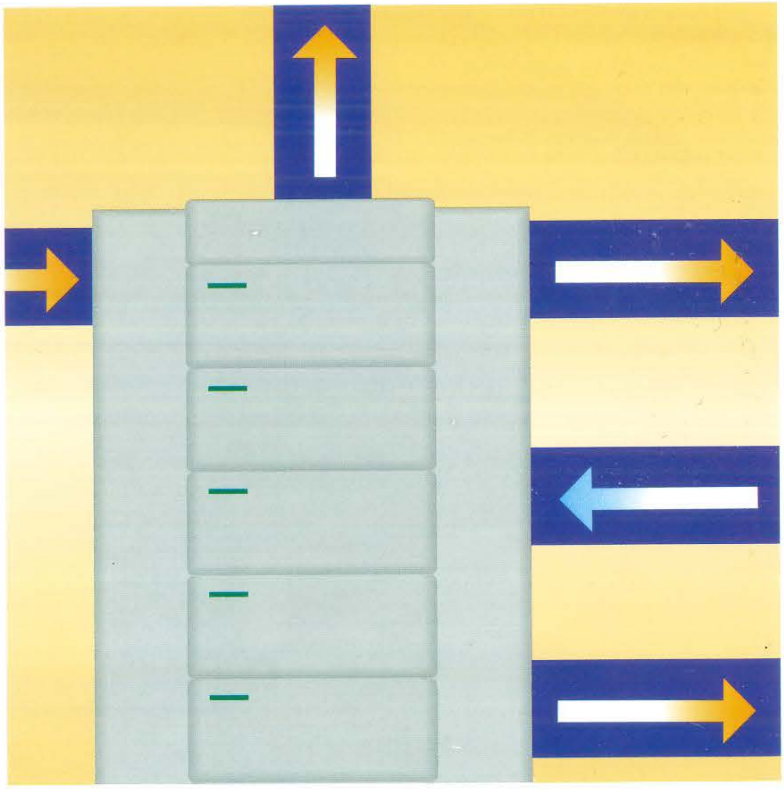




6 Often, in news stories, there are URLs to sites that have additional information. When that link is clicked on, an Internet browser is launched, and the specified site is contacted. The person can now read more in-depth news. For example, there may be a brief news story in PointCast about a company's quarterly earnings. That story could contain links to stories on the Web with more in-depth financial information, as well as other articles about that company.

CHAPTER

34 How Java, ActiveX, and JavaScript Work



THE Internet is no longer a place that you visit with your computer and merely look at documents or gather information—increasingly, it is an extension of your computer. You can now run programs that reside on the Internet rather than on your own computer, and tools have been developed that enable your computer and the Internet to interact as if they were one large computer system. This allows for all kinds of things never before possible: news tickers that flash breaking news; interactive games; multimedia presentations combining animation, sound, and music; and much more.

The three most important programming tools associated with Web technology are Java, ActiveX, and JavaScript. *Java*, a computer language developed by Sun Microsystems, enables applications to be run from the Internet, the same as word processing and spreadsheet programs are run on your computer. It is similar to the C++ computer programming language, and it is object-oriented, which means programs can be created by using many pre-existing components instead of by a programmer writing the entire program from scratch.

Java programs run inside your Web browser if you have a Java-enabled browser, such as Netscape. When Java programs are run inside a browser, they are called *applets*. You don't need to do anything to run a Java applet. When you visit a Web site that has a Java applet on it, the applet is downloaded automatically from a Web server, and then run automatically in your browser. Java applets can be run on any computer, such as a PC, a Macintosh, or a UNIX workstation.

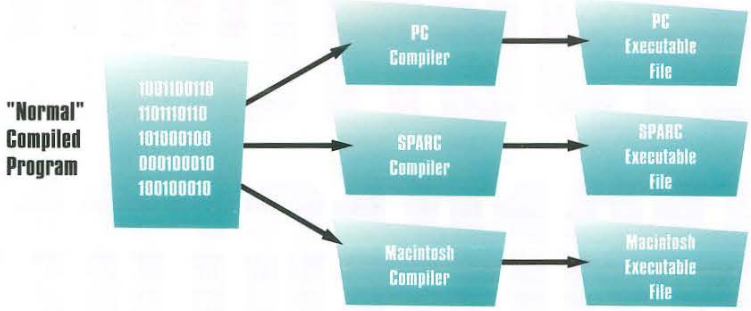
ActiveX, a technology from Microsoft, allows Internet programmers to create programs—commonly referred to as ActiveX *controls* or *components*—that can essentially turn the Internet into an extension of your computer. Like Java applets, these controls are downloaded to your computer and run there. They can do anything that a normal application can do, in addition to interacting with the Web, the Internet, and other computers connected to the Internet. To run these controls, a browser that supports ActiveX, such as Internet Explorer, is needed.

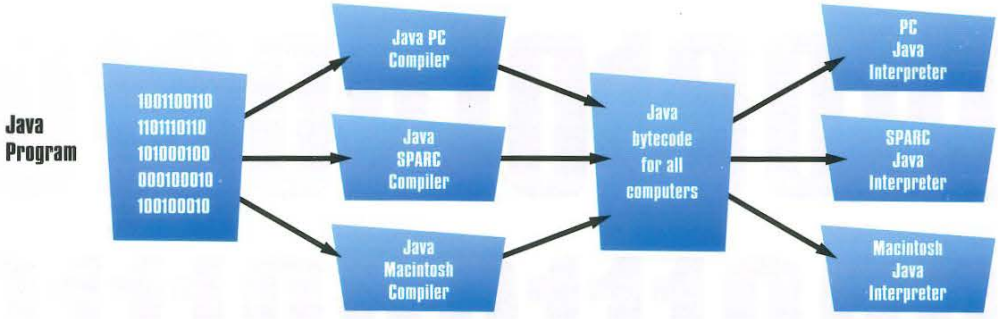
One benefit of ActiveX controls is that they are written as components, which means they can be put together, much like building blocks, to build larger and more complex applications. Another benefit is that because you have already downloaded a component, you won't ever need to download it again. This means when you visit a page with a complex ActiveX application on it, you may need to download only a small portion of it because you may already have the other components on your computer.

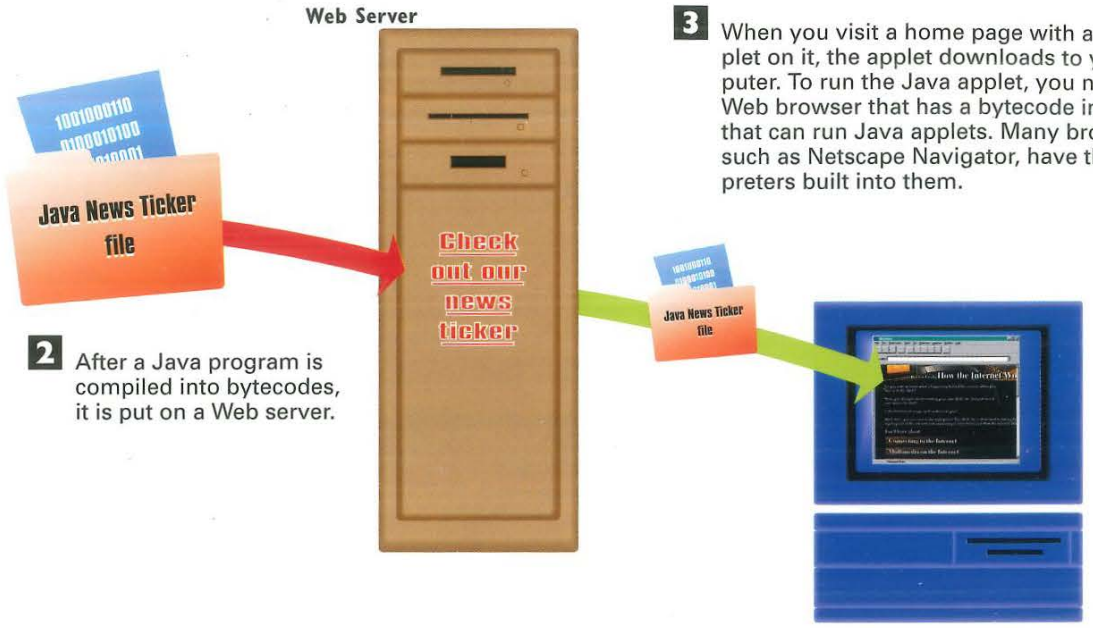
JavaScript is a scripting language that is less complex and, therefore, much easier to learn than Java and ActiveX. People without substantial programming experience can write scripts with JavaScript. It's also an interpreted language, which means that its commands are executed by the browser in the order in which the browser reads them. It's commonly used for things such as creating drop-down boxes, navigational aids, and interactive forms, although it can be used for creating more complex applications, as well.

How Java Works

Java is a *compiled language*, which means that after a Java program is written, the program must be run through a compiler to turn the program into a language that a computer can read. Java differs from other compiled languages, however. In other compiled languages, computer-specific compilers create distinct executable binary code for all the different computers on which the program can run. In Java, by contrast, a single compiled version of the program—called Java *bytecode*—is created by a compiler. Interpreters on different computers, such as a PC, Macintosh, or SPARC workstation, understand the Java bytecode and run the program. In this way, a Java program can be created once and then used on many kinds of computers. Java programs designed to run inside a Web browser on the World Wide Web are called *applets*. Java-enabled browsers contain Java bytecode interpreters.

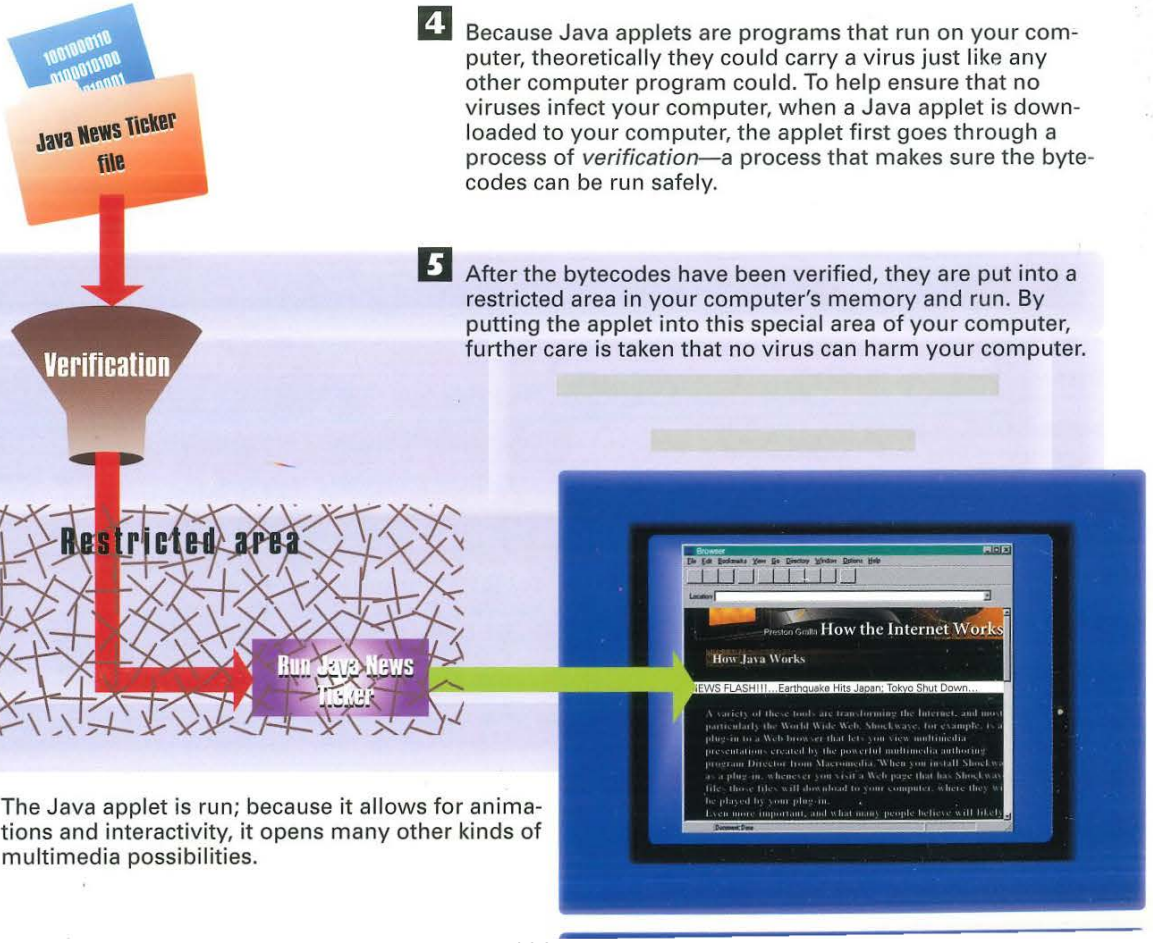






2 After a Java program is compiled into bytecodes, it is put on a Web server.

3 When you visit a home page with a Java applet on it, the applet downloads to your computer. To run the Java applet, you need a Web browser that has a bytecode interpreter that can run Java applets. Many browsers, such as Netscape Navigator, have these interpreters built into them.

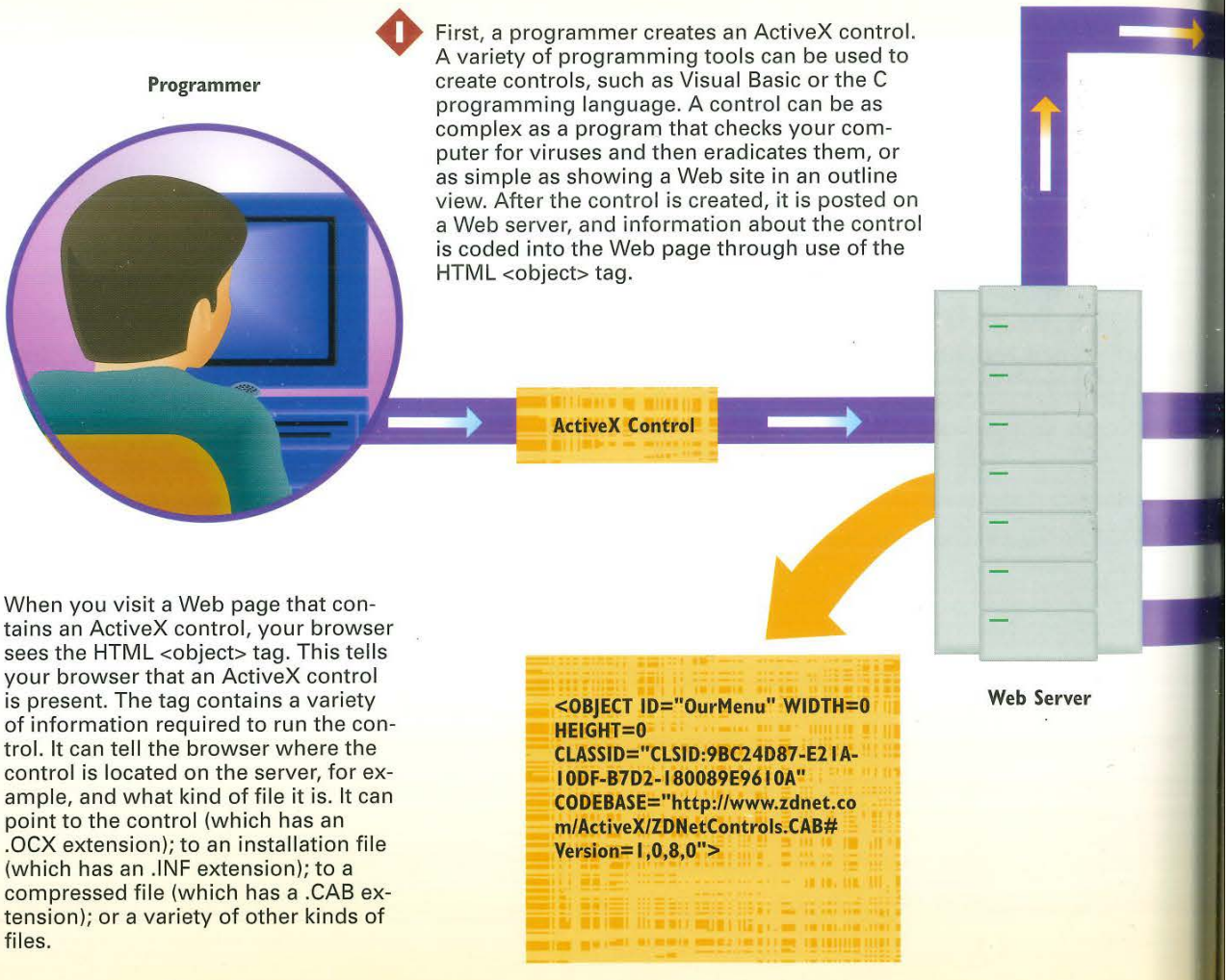


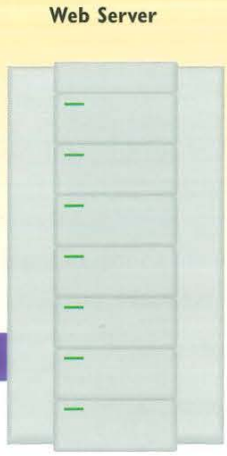
4 Because Java applets are programs that run on your computer, theoretically they could carry a virus just like any other computer program could. To help ensure that no viruses infect your computer, when a Java applet is downloaded to your computer, the applet first goes through a process of *verification*—a process that makes sure the bytecodes can be run safely.

5 After the bytecodes have been verified, they are put into a restricted area in your computer's memory and run. By putting the applet into this special area of your computer, further care is taken that no virus can harm your computer.

6 The Java applet is run; because it allows for animations and interactivity, it opens many other kinds of multimedia possibilities.

How ActiveX Works





3 Some ActiveX controls need more than a single control to work. In that event, the HTML page will contain multiple references to ActiveX controls needed to run the control on that page. The controls can be located on the same server, on a different server on the same Web site, or on another site and server on the Internet.

Get ActiveX Control

4 Using the information in the <object> tag, the browser starts to download the ActiveX controls. If you have set your browser to a certain level of security, you will get a message asking whether you want to download the control. To ensure even more security, ActiveX controls can be digitally "signed" by a digital certificate authority such as VeriSign. This signing will assure you that the control you want to run was written by the person to whom it is attributed. If a problem occurs with the control, you will be able to contact that person.



Do you want to Download Control?

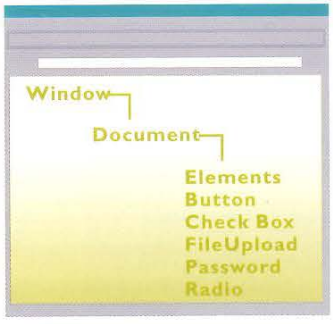
Yes

ActiveX Control

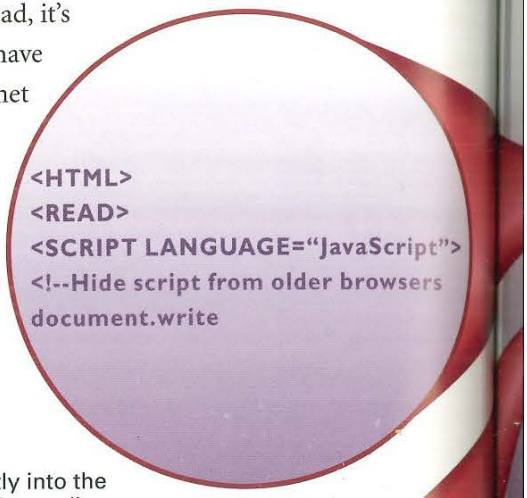
5 If you have low security set on your browser, or if you give the okay to download the control, the control and its related ActiveX controls are downloaded to your computer. Some of the controls may already be on your system, so you won't need to download them. After the control is downloaded, the file is decompressed (if it was compressed), information about it is put into the Windows Registry, and it is installed on your computer. The control will then run. An ActiveX control can do anything that any other program can do. It can interact with your computer and with any Internet resource, such as the Web, FTP, Telnet, or virtually any other Internet resource. It can also directly use the Internet's TCP/IP protocols so that it need not ride on top of another Internet resource.

How JavaScript Works

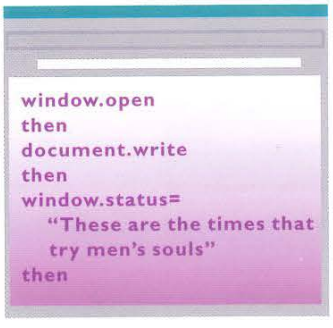
JavaScript allows Web designers to add functionality and interactivity to Web pages that otherwise couldn't be gained with HTML, the language of the Web. It can create interactive forms, for example, or make text turn colors when a mouse moves over it, or any number of other kinds of interactivity. Despite its name, JavaScript is not a form of the Java programming language. Instead, it's a simpler scripting language. JavaScript will run only on browsers that have been explicitly designed to take advantage of it, such as Microsoft Internet Explorer and Netscape Navigator.



1 JavaScript is an object-oriented language, which means that it works by manipulating objects on a Web page, such as windows, buttons, images, and documents. It groups these objects into hierarchies, which enables programmers to manipulate them more easily.

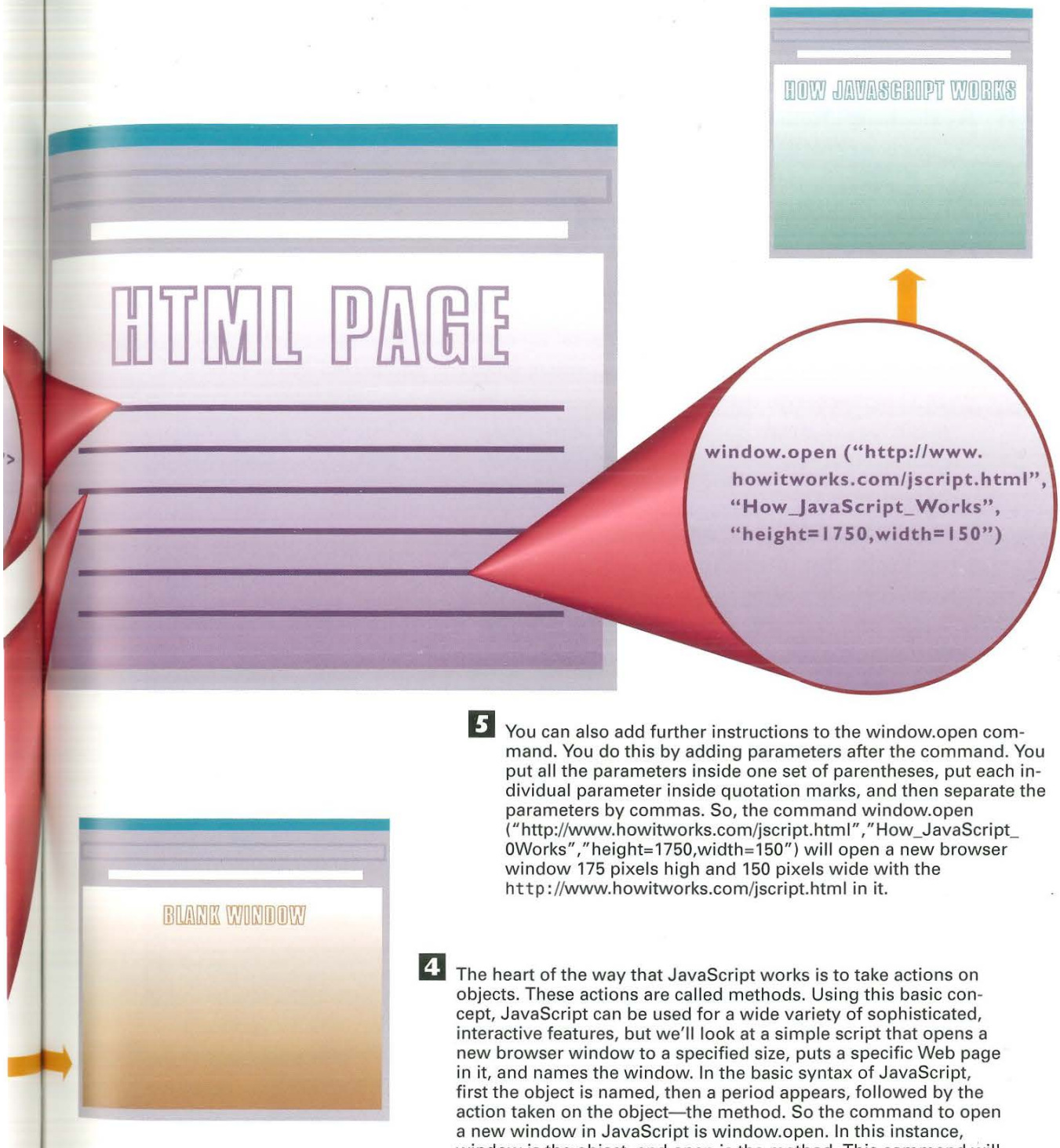


3 JavaScript commands are put directly into the HTML file that creates a Web page. Depending on the script being run, the commands can be put into several places in the file. Often, the commands are put near the top of the file. Special codes set off the commands, alerting the browser that they're JavaScript commands. If the commands are put before the HTML <body> tag at the top of the file, then the script will be able to start executing while the HTML page is still loading.



2 JavaScript is also an interpreted language, which means that its commands are executed by the browser in the order in which the browser reads them.





5 You can also add further instructions to the window.open command. You do this by adding parameters after the command. You put all the parameters inside one set of parentheses, put each individual parameter inside quotation marks, and then separate the parameters by commas. So, the command window.open ("http://www.howitworks.com/jscript.html","How_JavaScript_0Works","height=1750,width=150") will open a new browser window 175 pixels high and 150 pixels wide with the `http://www.howitworks.com/jscript.html` in it.

4 The heart of the way that JavaScript works is to take actions on objects. These actions are called methods. Using this basic concept, JavaScript can be used for a wide variety of sophisticated, interactive features, but we'll look at a simple script that opens a new browser window to a specified size, puts a specific Web page in it, and names the window. In the basic syntax of JavaScript, first the object is named, then a period appears, followed by the action taken on the object—the method. So the command to open a new window in JavaScript is window.open. In this instance, window is the object, and open is the method. This command will open a new browser window.

CHAPTER 35

How Agents Work



THE Internet has grown so quickly and its resources are so vast that we need help navigating around it. We can now use special software called *agents* to help us access the Net's resources.

While there are a lot of technical definitions for agents, put simply, agents are programs that do your bidding automatically. Many of them run over the Internet or on individual computers every day. Agents can find the latest news for you and download it to your computer; they can automatically monitor Internet traffic and report on its total usage; they can find you the best deal on the CD you want to buy; they can perform important Web maintenance tasks; and far more. They are becoming so complex that systems are being developed to allow agents to interact with one another so they can perform jobs cooperatively.

On the Internet, agents are commonly called *spiders*, *robots* (often shortened to "bots"), and *knowbots*, among other terms. Those used for searching automatically create indexes of almost every resource on the Web and then allow people to search through those indexes to find things more quickly. Common search tools such as Lycos, InfoSeek, and AltaVista use spiders in this way. We covered this specialized use of spiders in Chapter 24, "Searching the Internet."

All of these agents are software programs that are invisible to the user. You just determine the task you want done and behind the scenes the agent automatically goes off and performs that task. A variety of different languages can be used to write agent programs.

Agents may well alter the way we all use the Internet in the future. Not only will they respond to our requests, but they will also "learn" from our requests the kinds of tasks and information that interest us. They'll then go off on their own and perform those tasks and get that information, even before we make these additional requests. As we use these kinds of agents more, they'll become even smarter and more efficient.

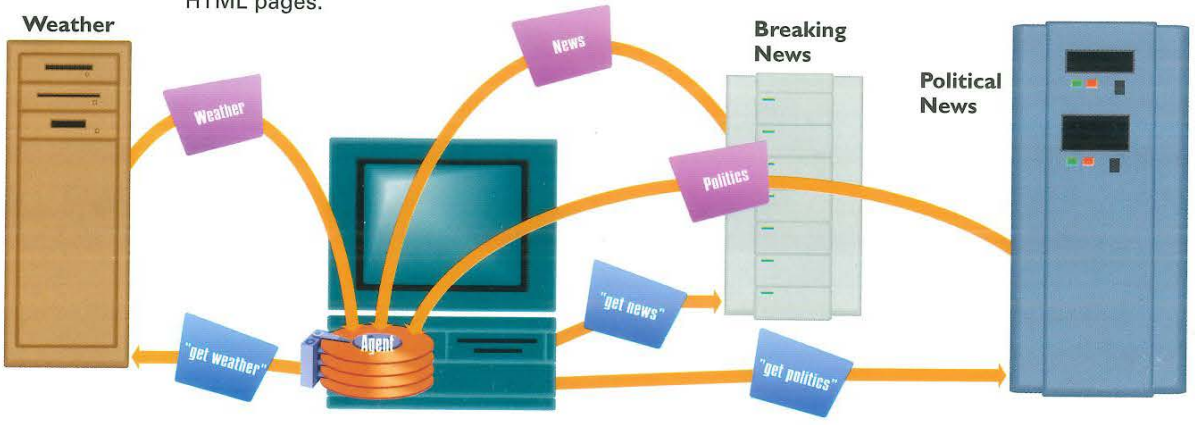
Robots and agents can cause problems for some Web sites. For example, they can overload Web servers by swamping them with too many requests in too short a time. That means users who try to get access to those Web pages will either be denied access or access will be exceedingly slow.

Another problem has to do with the way Web sites make money. Many Web sites sell ads to support themselves and they charge advertisers based on the number of pages that have been viewed. If many of those pages "viewed" are in fact never seen by people and are instead only accessed by a computer via a robot, both advertisers and the Web site suffer.

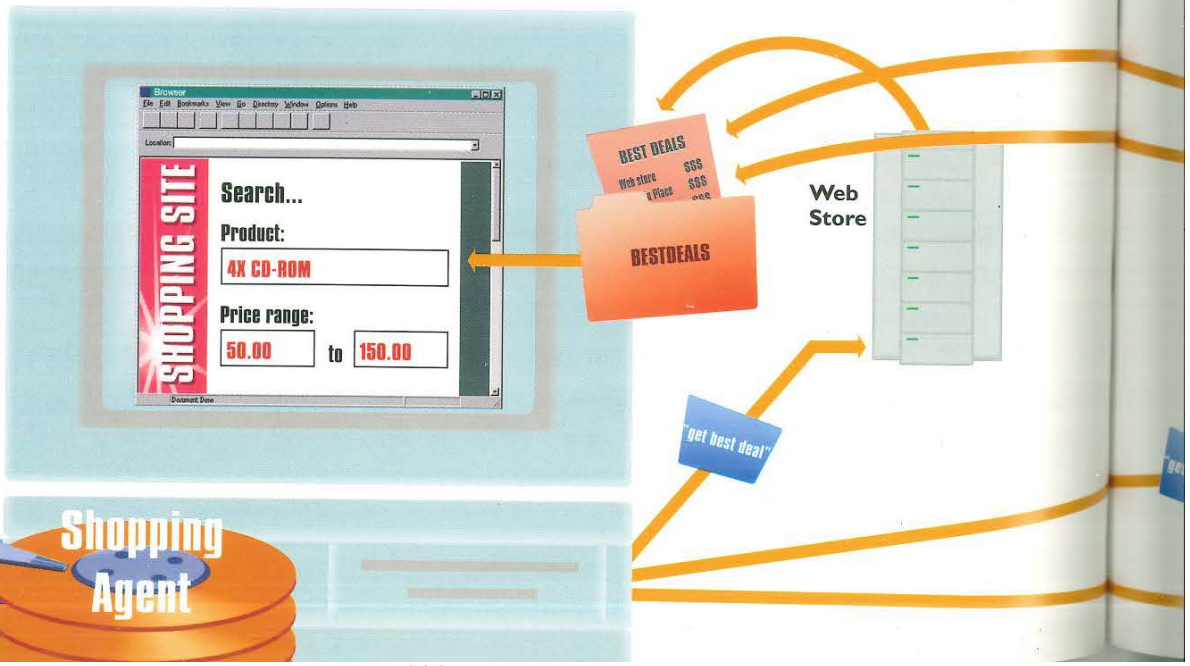
There are several ways to solve these problems and limit robot access. One way includes creating a file called Robots.txt that describes the areas that are off limits to robots, which the robots would automatically read, adhere to, and not visit. Another is to use a technology that automatically detects whether a robot or a human has visited a page and forgo charging advertisers whenever robots visit.

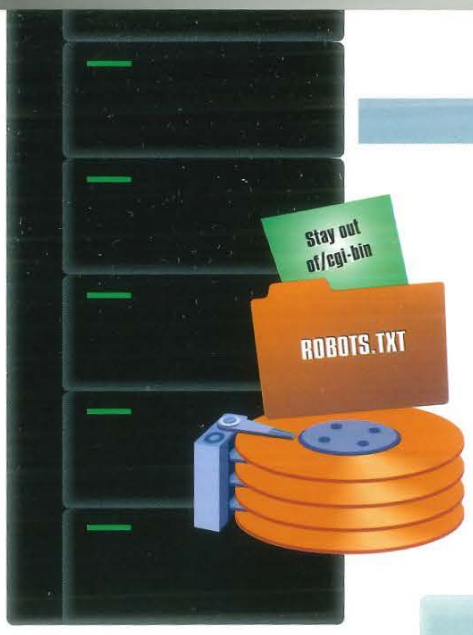
Agents on the Internet

1 A simple Internet agent is one that gathers news from a variety of sources while you're not using your computer or while you are using your computer for another task. News agents can work in several ways. In the simplest example, you fill out a form saying what kind of news you're interested in and on what schedule you want your news delivered. Based on that information, at preset intervals, the news agent dials into news sites around the Internet and downloads news stories to your computer where you can read them as HTML pages.

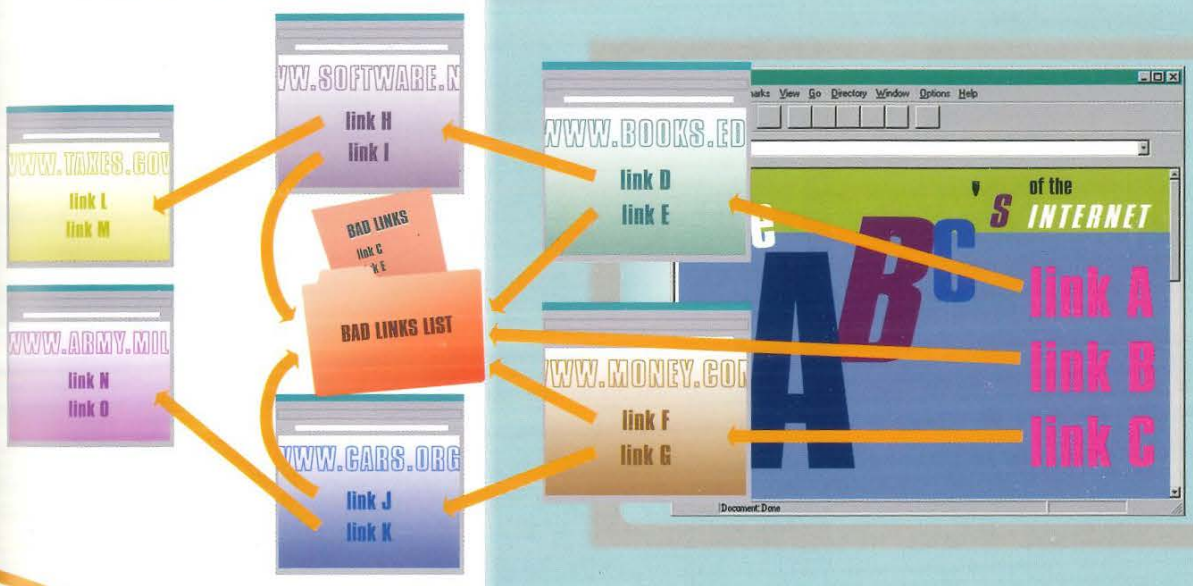


2 Shopping agents will let you search through all of the Internet for the best bargains. On the Web, you fill out a form detailing the product you want to buy. When you send the form, the shopping agent launches programs that search through a variety of shopping sites and databases on the Internet. The agent looks into the databases of those sites and finds the best prices. It then sends back to you the links to the sites so you can visit the sites with the best prices and order from there.





4 When robots and spiders do their work on a remote Internet site from where they were launched, they can put an extra load on the site's system resources—for example, by swamping the server with too many requests in too short a time. Because of this, some system administrators are interested in ways of excluding robots in certain circumstances, such as not allowing robots into certain Web directories. A variety of ways have been devised to limit robot access, including creating a file called Robots.txt that describes the areas off limits to robots, which the robots would automatically read, adhere to, and not visit.

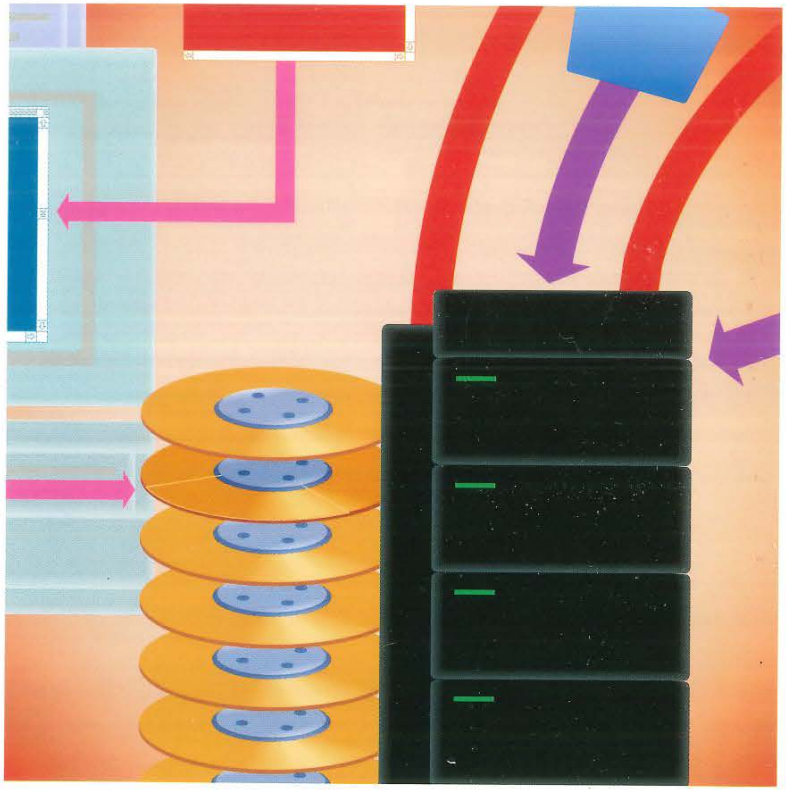




3 Web robots called *Web maintenance spiders* can perform important Web maintenance chores. On Web sites, particularly large ones, very often HTML pages can include links that become outdated. In other words, the object being linked to has been taken off the Internet. Whenever a user clicks on the link, an error message is sent. A Web maintenance spider can look at every link on every HTML page on a Web site and trace each link to see if the linked object still exists. It then generates a report of dead links. Based on that report, the system administrator can rewrite the HTML code, getting rid of the bad links.

CHAPTER 36

How CGI Scripting Works



IF you browse the Web for very long you are sure to come across the term CGI, or Common Gateway Interface. CGI refers to the communications protocol by which a Web server can communicate with other applications. For example, a CGI application, sometimes called a "script," is often used to allow Web users to access databases, or to get information from forms that people fill out. CGI can also be used to create agents that do things such as check a Web site to see whether there are any broken links on the site.

Essentially, CGI is a standard way in which the Web interacts with outside resources. Often, that outside resource is a database. You've probably run CGI scripts many times without knowing it. For example, if you've filled out a form on a Web page to register to use a site and then later received an email notification with a password for you to use, you've probably run a CGI script. In that case, the CGI script probably took the information you filled in on the form and performed several actions on it, including putting the information in a database, automatically creating a password, and then sending you mail.

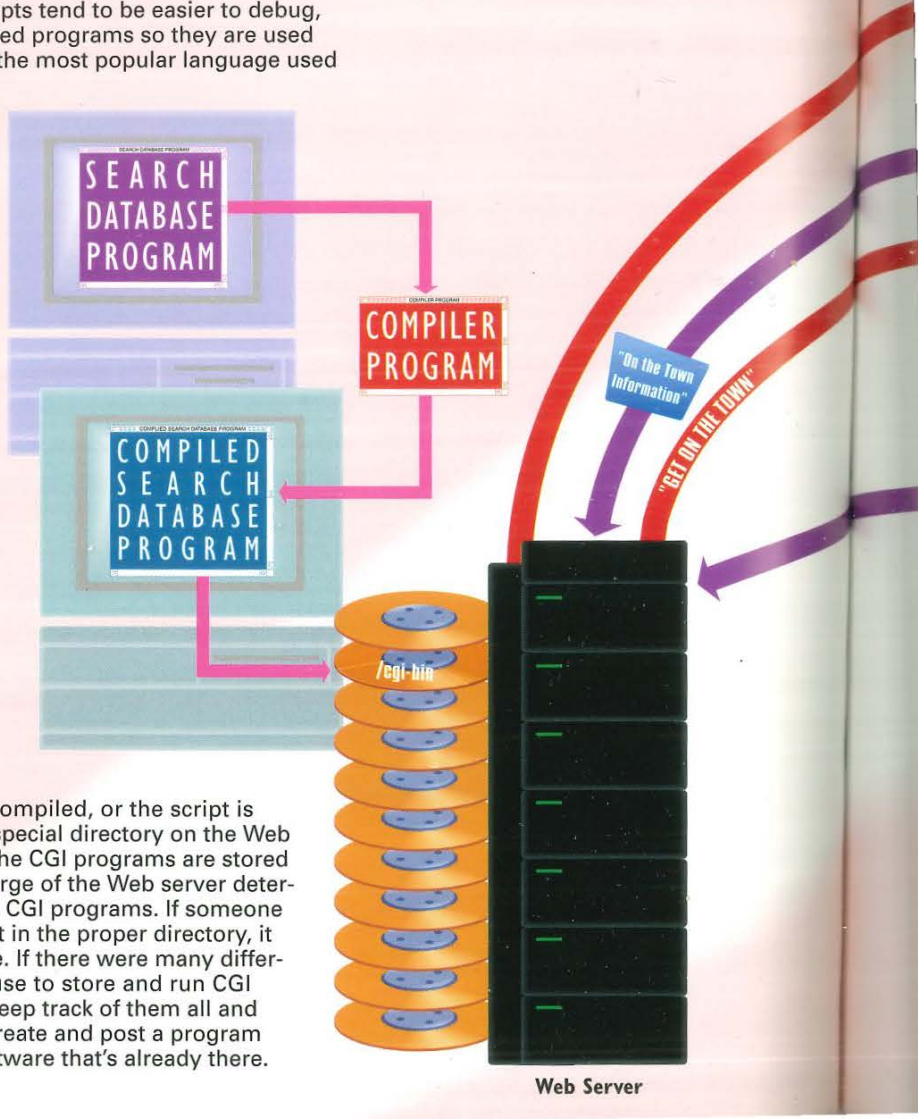
CGI and CGI applications are often confused. *CGI applications* receive data from the server and return the data via the Common Gateway Interface. CGI applications are usually written in a programming language called Perl (Practical Extraction and Reporting Language), although they may be written in C, C++, Pascal, AppleScript, or others as well. CGI itself is a standardized means of communicating between a CGI application and the HTTP server. It's the "doorway" of sorts through which the Web server sends requests and the CGI application collects and returns data.

In the example of providing information on a Web page designed to accept user input, CGI performs many tasks. First, you submit unique information—like a name or email address—to the server for processing. Next, the server redirects the information to a CGI application that is called by the form "submit." CGI scripts are activated by the server in response to an HTTP request from the client. Lastly, a CGI application may send form data to another computer program, such as a database, save it to a file, or even generate a unique HTML document in response to the user's request. This is known as an *interactive form.*

In the illustration that accompanies this chapter, we'll look at a CGI program that allows someone to search a movie database for information.

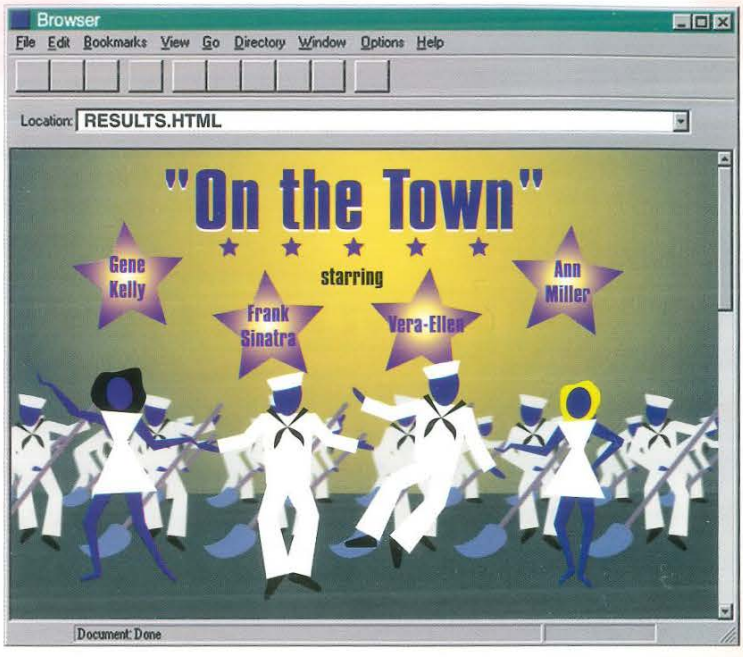
How CGI Scripting Works

1 People who dial into the Web site don't need to know programming to access CGI programs. Instead, a programmer writes a CGI program. A number of different languages can be used for CGI, such as C or C++, FORTRAN, Visual Basic, and AppleScript. An application written in a programming language such as C must be passed through a program called a compiler before it can be run. The compiler turns the application into a language that CGI can understand. Other languages, called *scripting languages*, do not need to be compiled first. CGI scripts tend to be easier to debug, modify, and maintain than compiled programs so they are used more frequently. Perl is probably the most popular language used for writing CGI scripts.

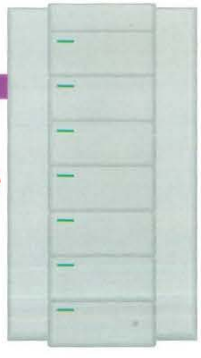


2 After the program is written and compiled, or the script is written, the program is put into a special directory on the Web server, such as /cgi-bin, where all the CGI programs are stored and maintained. The person in charge of the Web server determines which directory should hold CGI programs. If someone writes a program and doesn't put it in the proper directory, it won't run. This is a security feature. If there were many different directories that people could use to store and run CGI programs, it would be difficult to keep track of them all and someone from the outside could create and post a program that could be dangerous to the software that's already there.

6 The CGI program receives the data from the database and formats it in a way that will be understandable to the user. For example, the program might take the information and put it into HTML format so the user can read it using his or her Web browser. The CGI program sends the results in HTML format to the user, who displays it in a Web browser. The user can then access that HTML page. He or she can click on links to visit other pages, print pages, and view graphics and multimedia files.



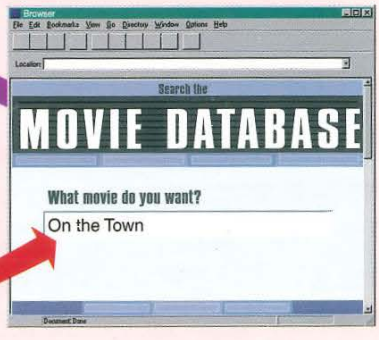
Database



5 The CGI program contacts a database and requests the information that the user is looking for. The database sends the information to the CGI program. The information can be in a variety of formats, such as text, graphics, sound and video files, and URLs.

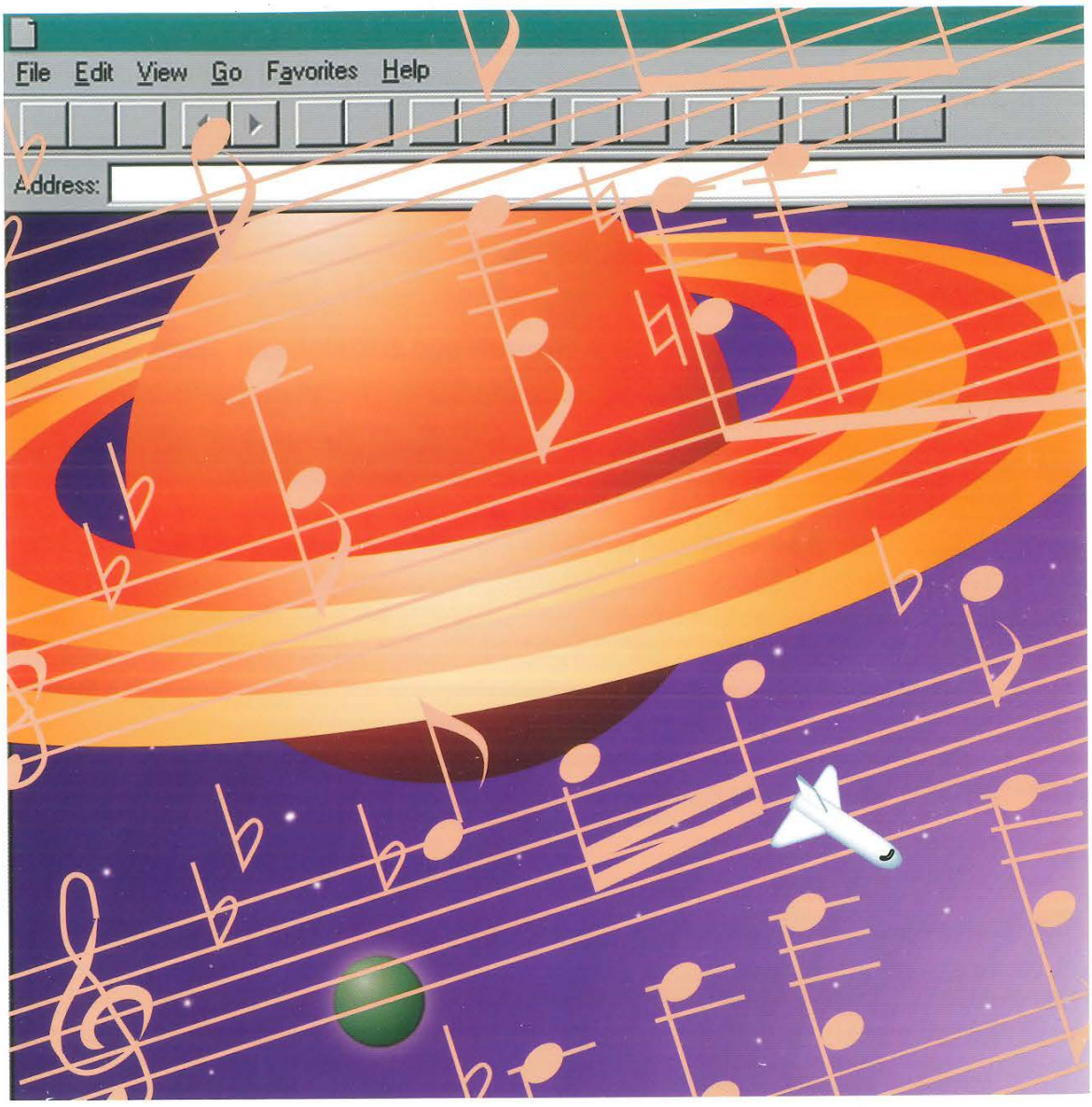
4 When you visit the Web site and click on the URL, the CGI program is launched. If the CGI program allows you to search a database, for example, it will send a form in HTML format. You then fill out the form detailing what you want to find. When you finish the form and click on Send, the data from the form is sent to the CGI program.

"get On the Town"





3 After the CGI program is posted to a special directory, a link to it is embedded in a URL on a Web page.


File Edit View Go Favorites Help
Address:

PART

8

MULTIMEDIA ON THE INTERNET

Chapter 37: Audio on the Internet
216

Chapter 38: Video on the Internet
220

Chapter 39: How Multicast IP and the MBone Work
228

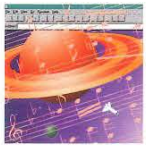
Chapter 40: Virtual Reality on the Internet
232

Chapter 41: Animation on the Web
236

POSSIBLY the most dramatic and remarkable part of the Internet is the multimedia content. You'll be able to listen to music, sound clips, and live radio stations from your computer. You can watch astronauts live while they're on the space shuttle. You can watch video clips of the news and other events. And you can even have live videoconferencing with people from all over the world.

You can do all that with the Internet's audio and video capabilities. You won't need specialized hardware and software to do it—and in many cases, you won't even need a very high-speed Internet connection. An ordinary dial-in connection to the Internet will do, although the sound and video quality will be better at higher speeds. And you'll only need free or inexpensive software, and a sound card and speakers that ship with most computers, or that are available separately.

The Internet's multimedia capabilities go beyond mere playing of audio and video clips. You can participate in virtual worlds and join in "virtual chat" sessions in which you build your own online persona, called an *avatar*, which communicates with other avatars. The Internet allows for the creation of remarkable online multimedia content, combining animation, sound, and programming via technologies such as streaming audio and video, Shockwave, and Multicast IP.



In this section of the book, we'll look at how every aspect of multimedia on the Internet works. Chapter 37, "Audio on the Internet," covers audio. We'll see how audio files are sent to your computer and played. We'll look at how streaming audio works in detail. *Streaming audio* allows you to play sounds and music on your computer while the audio file is being transferred to your computer, so that you don't have to wait for the file to download.

Chapter 38, "Video on the Internet," details how video works. We'll look at how *streaming video* works, which (like streaming audio) lets you watch a video while it is being downloaded to your computer. Today, you're able to watch news broadcasts, music videos, and even live launches of the space shuttle through streaming video technology.

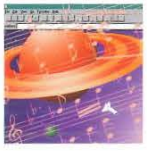
Chapter 38 also looks at videoconferencing. *Videoconferencing* enables people from different parts of the world to see each other and talk to each other—all through their computers. The voices and images are transferred over the Internet.

In addition, Chapter 38 covers a lighter topic: how NetCams work. *NetCams* are cameras on the Internet that broadcast a photograph or digital animation at regular intervals. NetCams are all over the world, from the top of Pike's Peak to the streets of Hong Kong.

Chapter 39, "How Multicast IP and the MBone Work," looks at Multicast IP and the MBone. *Multicast IP* is a technique that enables videos to be broadcast to many thousands of people simultaneously, without clogging up the Internet's backbone. And the *MBone* is a high-speed Internet backbone used for transmitting Multicast IP video across the Internet.

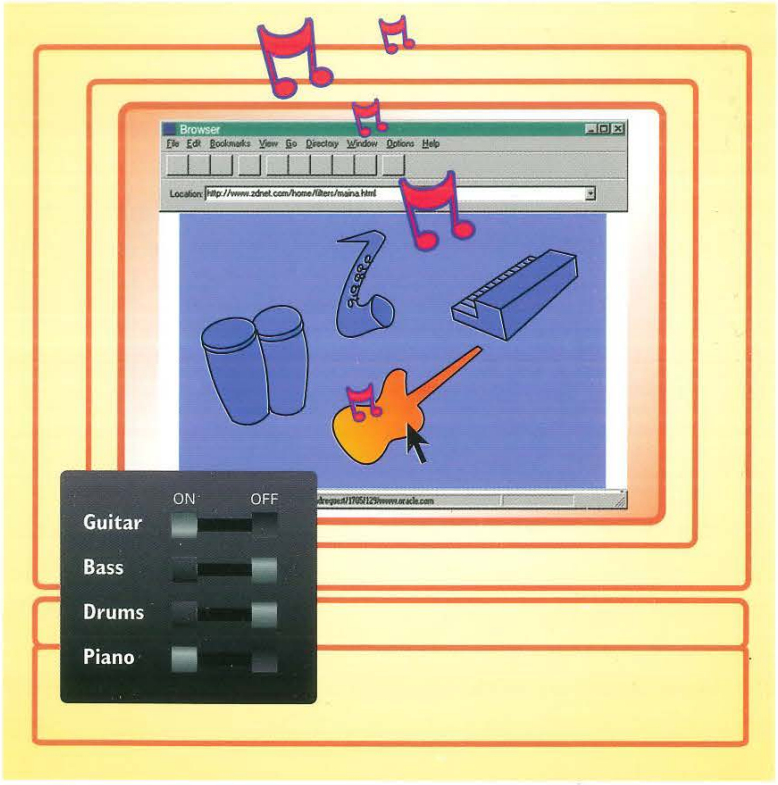
Chapter 40, "Virtual Reality on the Internet," examines virtual reality. *Virtual reality* allows for the creation of virtual worlds—3D creations on the Web through which you can walk or fly, interacting with your surroundings. As the bandwidth of the Internet increases, these virtual worlds may become increasingly popular.

Finally, Chapter 41, "Animation on the Web," looks at some of the most popular kinds of animation technologies, from the very simple to the very sophisticated. We'll see how *client pull* and *server push* technologies allow for the easy creation of simple animations. And we'll also look at Shockwave, an extremely sophisticated way in which animation, audio, and other kinds of interactive technologies can be used to create powerful multimedia presentations on the Web.



CHAPTER

37 Audio on the Internet



SOUNDS, voices, and music are now an everyday part of the Internet. Through the Internet, you can listen to radio stations, interviews, music, sound clips, and much more.

One of the most intriguing experimental uses of audio on the Internet was Internet Talk Radio. Modeled after National Public Radio, it featured special affairs programs, interviews, and a variety of special reports. Probably the best known show on Internet Talk Radio was "Geek of the Week," in which an Internet expert was interviewed every week. Internet Talk Radio no longer exists.

While Internet Talk Radio was popular, it was also hampered by a technology limitation. To listen to it on your computer, you had to download audio files to your system. Once they were on your computer, you could use special audio software and a sound card to listen to them. The problem was that the files were very large—often well over 10MB—so it could take hours to get them transferred to your computer before you could listen to them.

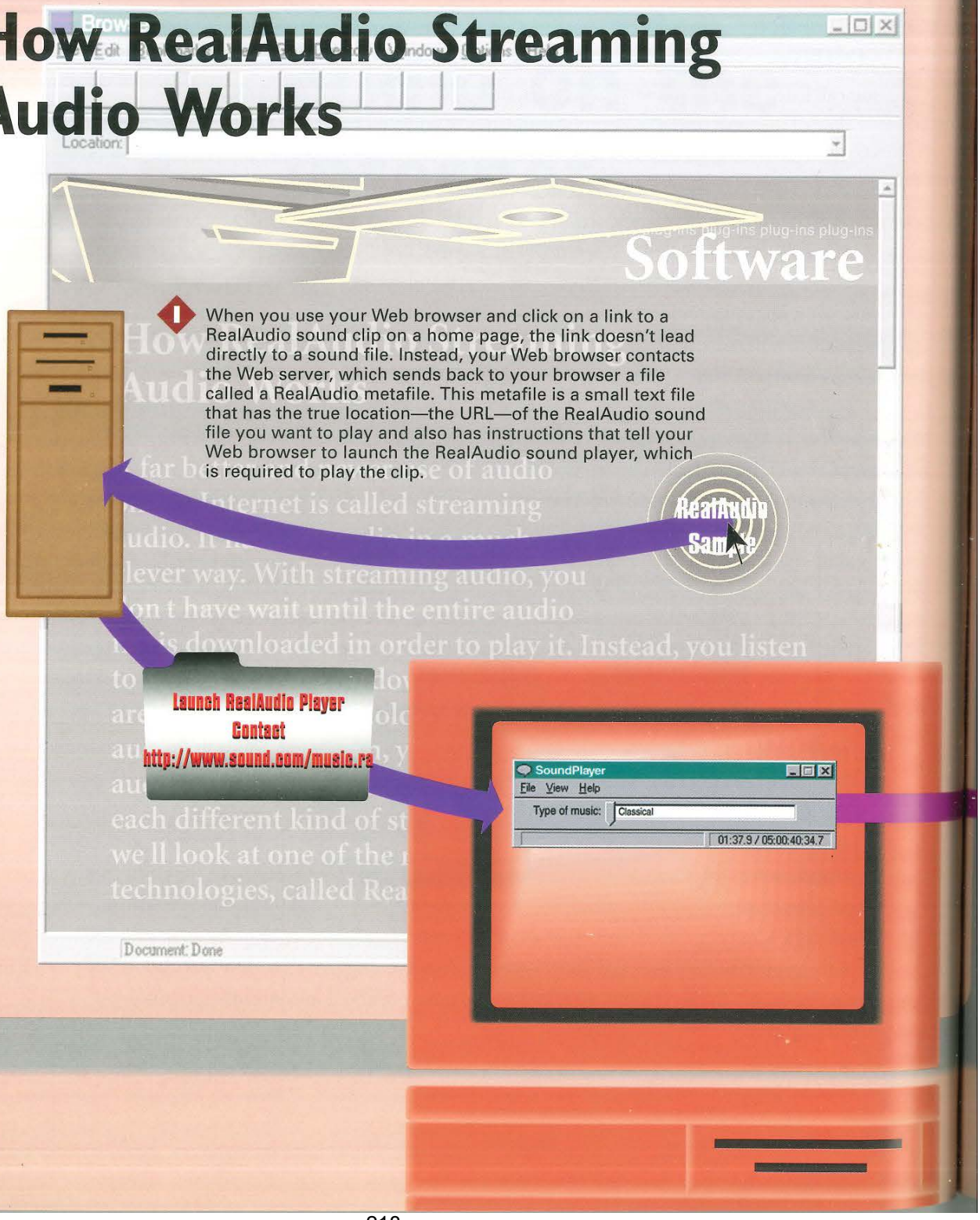
There are many other kinds of audio files you'll be able to find on the Internet. They all have one thing in common: They have been digitized so that a computer can play them. You'll find many music files and sound clips in a variety of sound formats. Each of those formats has a different extension associated with it, such as .WAV or .AU. To play those files, you'll first have to download them and then have audio player software play them on your computer. Netscape and other browsers have some of these software players built in. For additional formats, you'll have to find and download the player and then configure your browser properly to play them.

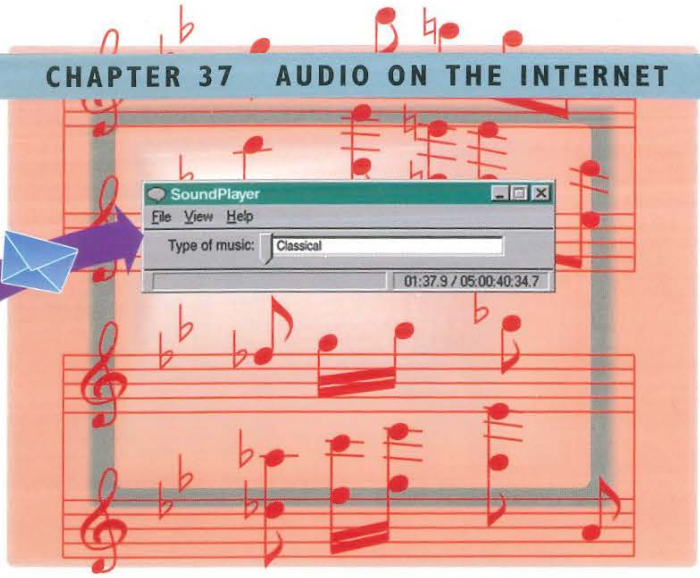
An even newer audio format, Rich Music Format (RMF) offers some intriguing possibilities. RMF allows you to interact with music in ways not possible before on the Web. For example, you can use RMF to change the music you're listening to from the Internet by eliminating some instruments or changing the music's tone and pitch.

Most sound files tend to be quite large, even after being compressed. You won't be able to listen to them until the entire file is downloaded, which can take quite awhile. It may take you 15 minutes to download a sound file that has less than a minute of sound in it.

A far better and newer use of audio on the Internet is called streaming audio. It handles audio in a much more clever way. With *streaming audio*, you don't have to wait until the entire audio file is downloaded to play it. Instead, you listen to the audio while it downloads to your computer. There are a variety of technologies that allow for streaming audio. For all of them, you'll need to have the proper audio player and you'll need a different audio player for each specific kind of streaming audio. In this chapter, we'll look at the most popular audio streaming technology, called RealAudio.

How RealAudio Streaming Audio Works

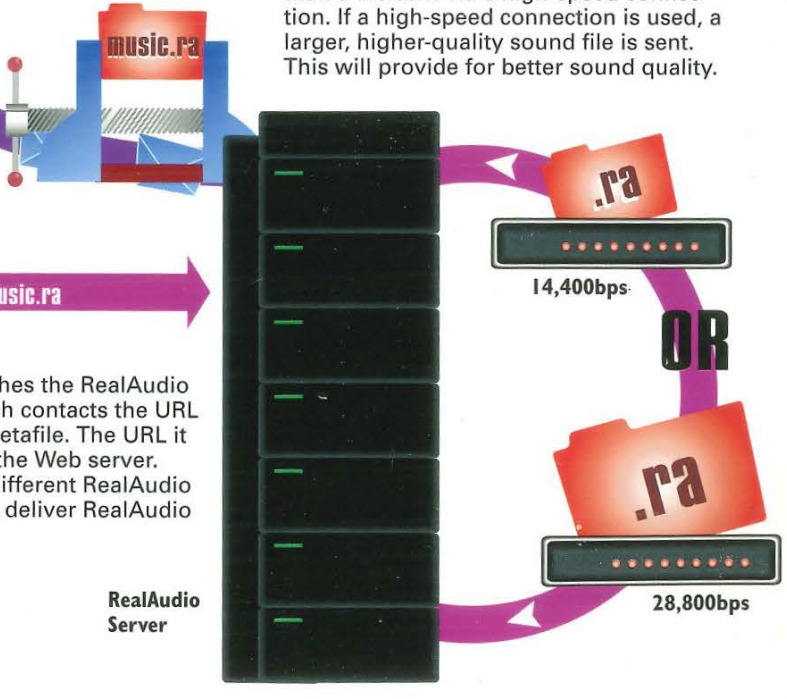




5 The packets are sent to a buffer on the receiving computer. Once the packets exceed the capacity of the buffer, they are sent to the RealAudio player, which then plays the sound file. RealAudio allows you to jump ahead or back in a sound or music clip. When you move to a different place in the clip, the RealAudio player contacts the server and tells it to start sending the file from that new place in the clip.

4 The RealAudio clip has been compressed and encoded. If the file was not compressed, the sound file would be too large and would take too long to send and be played. The clip is sent in IP packets using the UDP (User Datagram Protocol) instead of the Internet's normal TCP (Transmission Control Protocol). UDP doesn't keep resending packets if they are misplaced or other problems occur, as does TCP. If packets had to keep being re-sent, the sound player on the receiving end could be constantly interrupted with packets and could not play the clip.

3 The RealAudio server and the RealAudio sound player "talk" to one another so that the server knows at what speed the user is connected to the Internet. If the connection is a low-speed connection, a smaller RealAudio file is sent that contains less data. This will be a file of lesser quality than a file sent via a high-speed connection. If a high-speed connection is used, a larger, higher-quality sound file is sent. This will provide for better sound quality.



2 The metafile launches the RealAudio sound player, which contacts the URL contained in the metafile. The URL it contacts is not on the Web server. Instead, it is on a different RealAudio server designed to deliver RealAudio sound clips.

CHAPTER

38 Video on the Internet



THE Internet began as a way for people to share text-based information such as email, discussion groups, and file transfers. Today, however, the technology has advanced far beyond text. Currently on the Internet, you can have video conferences in which you talk live with someone and see them live on your computer screen. You can use *whiteboard* applications that let you see and talk to other people at their computers, and you can also work on a file together live on your computer screens. You can watch live video footage of astronauts from outer space. You can also watch taped videos whenever you want—not when a national broadcaster says you must watch them.

To understand how all this works, you'll need to understand three kinds of technologies. The first is called the *MBone (Multicast Backbone)*, a special Internet high-speed backbone capable of sending vast amounts of information. Many video transmissions—especially live ones—are sent across the MBone because of its high bandwidth. Turn to Chapter 39, "How Multicast IP and the MBone Work," for more information on the MBone.

The second technology is called streaming video. *Streaming video* solves a long-standing problem of sending video signals across the Internet. Video files tend to be extremely large because they have so much information packed into them. Because of that, sending video was never very practical—it could take hours to send a single video file to someone's computer. The person on the other end would have to wait until the entire file was downloaded and then play it—and it might play for only a few minutes.

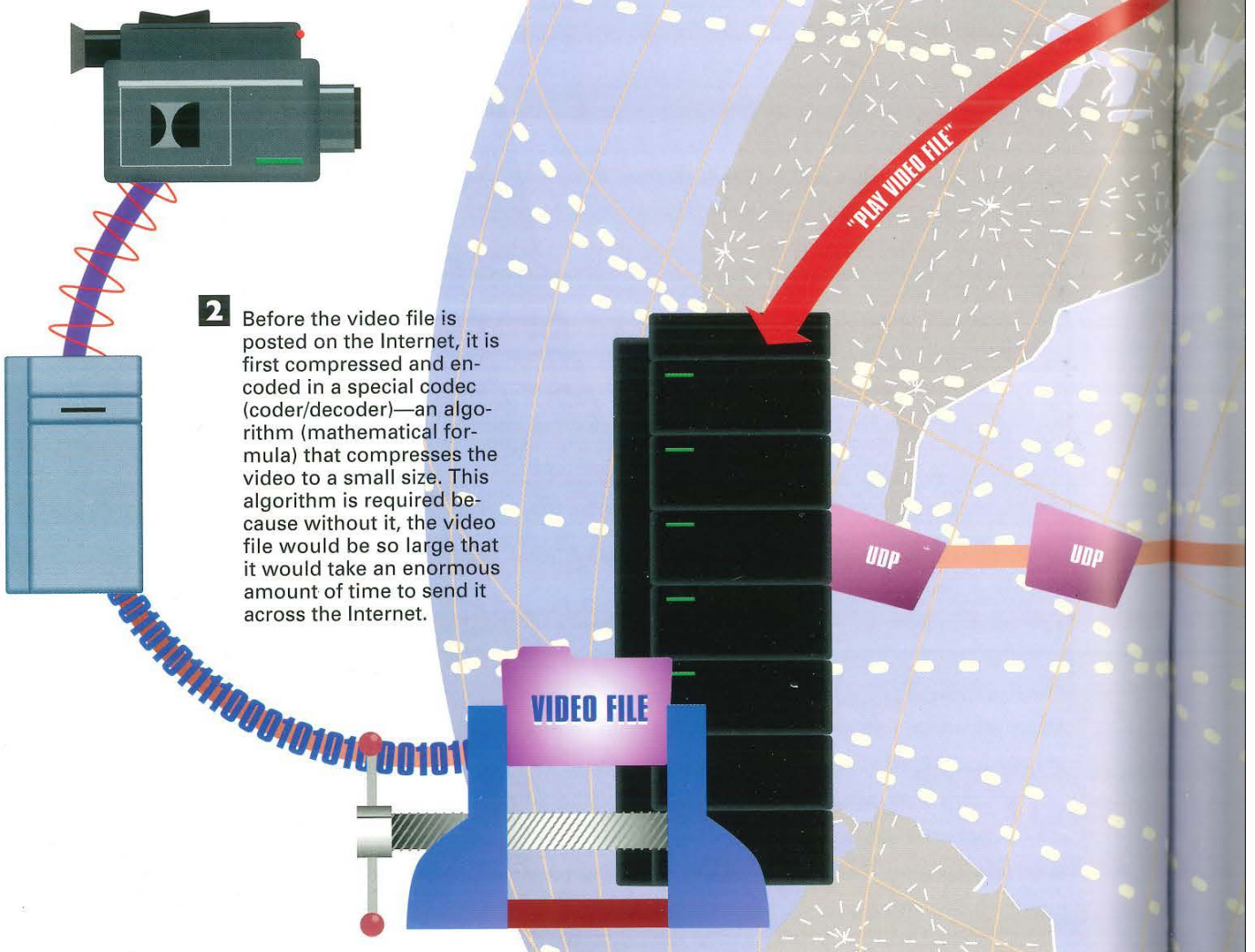
Streaming video solves the problem in two ways. First, it compresses the video file dramatically so it is much smaller as it is transmitted across the Internet. Secondly, streaming video lets the receiving computer start playing the video while the file is being transmitted. So if you receive a streaming video file, you watch the video as you receive it; no waiting for the entire file to download. Streaming video files are not usually live broadcasts. Instead, they are often files that are created ahead of time and then posted on the Internet. You can watch the video by clicking on its hypertext link. You'll need a special player to watch the video. There are a number of different ways to send streaming video across the Internet and watch it as it comes to your computer.

The third piece of technology is videoconferencing. It lets you use your computer to have live videoconferences across the Internet. Videoconferencing is done live, although the technology can also be used to broadcast taped videos as well. NASA, the National Aeronautics and Space Administration, sometimes uses the technology to broadcast live from the Space Shuttle and also to broadcast taped videos about space exploration.

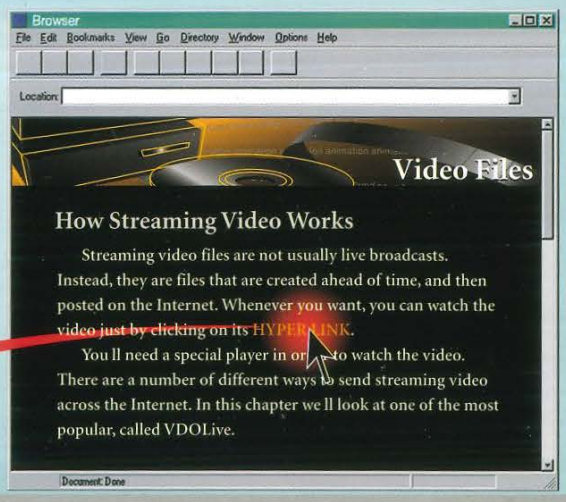
NetCams, also called digital cameras, are another intriguing use of video technology. A *NetCam* is a camera attached to the Internet that automatically broadcasts photographs of moving images at certain intervals. Photos can be downloaded or browsed with your Web browser. There are hundreds of NetCams on the Internet, sending live pictures from all over the world.

How Streaming Video Works

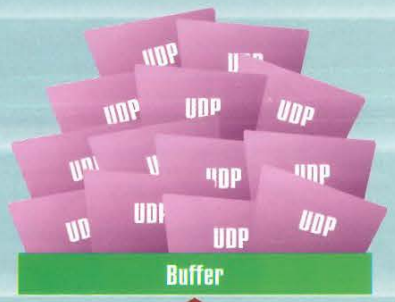
1 Streaming video refers to video that you can play live on the Internet—you don't have to wait until the download is complete to see the video. Instead, you can play the video while it is being sent to your computer. VDOLive is an example of streaming video.



2 Before the video file is posted on the Internet, it is first compressed and encoded in a special codec (coder/decoder)—an algorithm (mathematical formula) that compresses the video to a small size. This algorithm is required because without it, the video file would be so large that it would take an enormous amount of time to send it across the Internet.



3 When you send a request to see the video by clicking on an icon or a link on a World Wide Web page, you are sending a message from your computer to a server asking for the video file. The server sends the file to you in packets across the Internet, using the IP protocol. It does not use the normal Internet TCP, though. Instead it uses the UDP (User Datagram Protocol). Unlike TCP, the UDP does not constantly check to see if data has been sent, so it results in a more uninterrupted file transfer.



4 The video packets are sent to a buffer in your computer—an area of memory that ranges between 5KB and 30KB. The server can tell by how fast the buffer fills up what speed connection you have to the server. At higher speeds, it will send more video data and you will get a smoother, more lifelike video. At lower speeds, it sends less data, which causes the video quality to suffer.

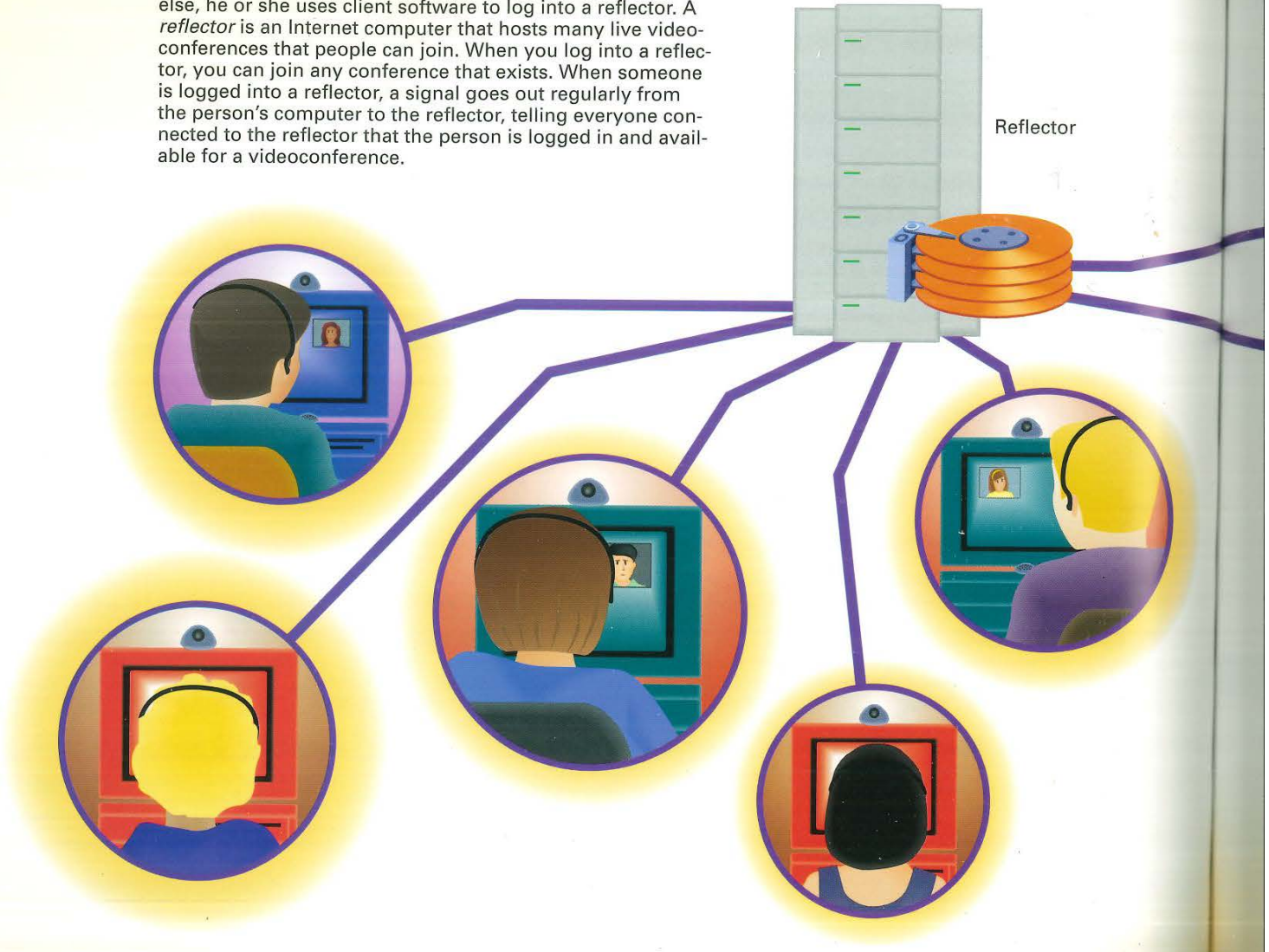


5 When the buffer fills up, which takes only a few seconds, a video player is launched on your computer. You can now watch the video on the player. As you watch the video, video packets are still being delivered to your buffer. Data from the buffer is continually sent to the player so that you can watch an entire video. When all the video data has been sent, the video will stop. The video file does not stay on your system; each section of the file is discarded after it is played.

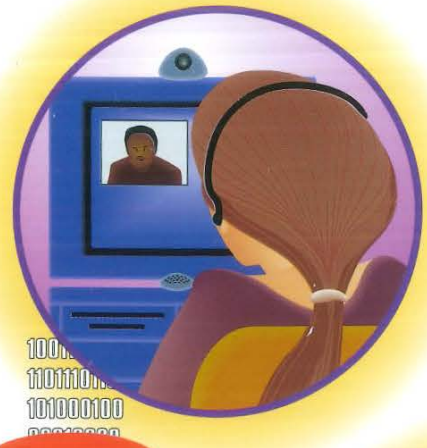
How Videoconferencing Works

1 There are a variety of ways for people to videoconference across the Internet. Most require client software that allows people to send and receive video and audio signals. Also involved are special kinds of server hardware and software (sometimes called reflectors) that host the videoconference and send the signals to everyone connected to the server.

2 When one person wants to videoconference with someone else, he or she uses client software to log into a reflector. A *reflector* is an Internet computer that hosts many live videoconferences that people can join. When you log into a reflector, you can join any conference that exists. When someone is logged into a reflector, a signal goes out regularly from the person's computer to the reflector, telling everyone connected to the reflector that the person is logged in and available for a videoconference.



3 To be a live participant in a videoconference, you need a video camera and a microphone on your computer. The camera converts your video image into digital data. The client software then compresses and encodes that data, enabling it to be sent across the Internet. If the data were not compressed and encoded, it would be too large to be sent.

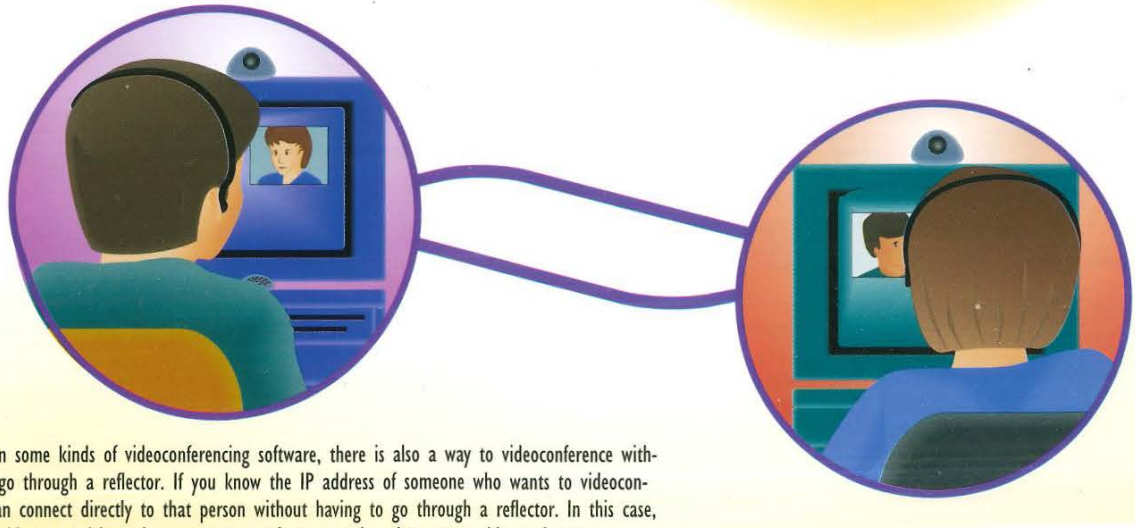


4 As another way of cutting down on the amount of data that needs to be sent across the Internet, some kinds of client software send only part of the screen and not the entire video image. They send only the part of the image that has changed. So, for example, if someone moves his or her head, it will transmit only the head moving and not the background, which hasn't changed.

5 The video data is sent across the Internet using UDP, which is more efficient than TCP when you are sending video data.

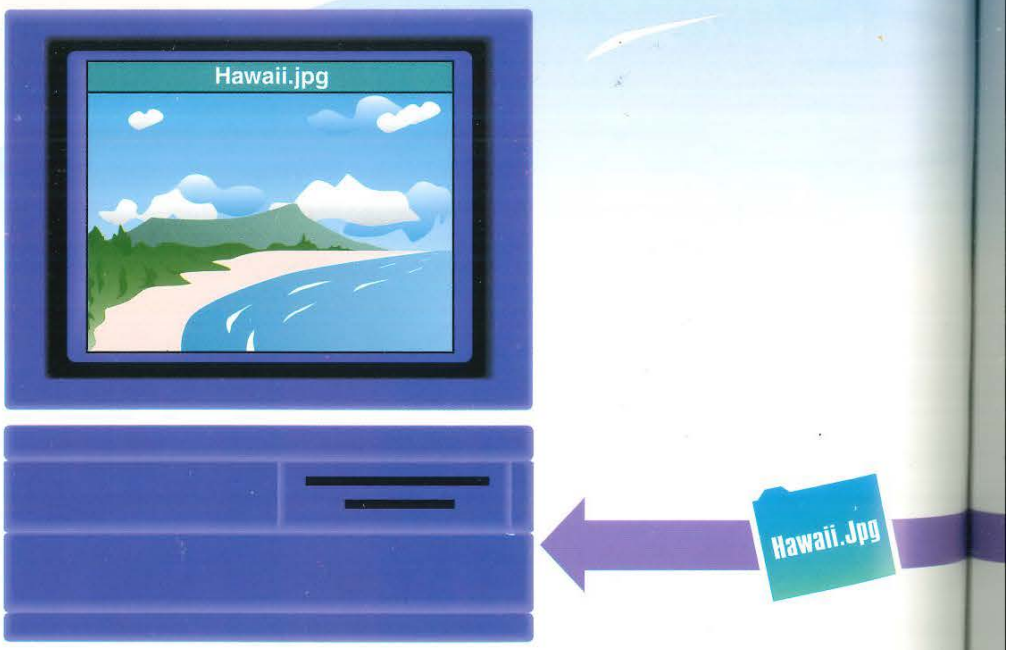


6 You and the person who wants to see your image both need the same client software. The software decodes the video image and displays it as video on the person's screen.

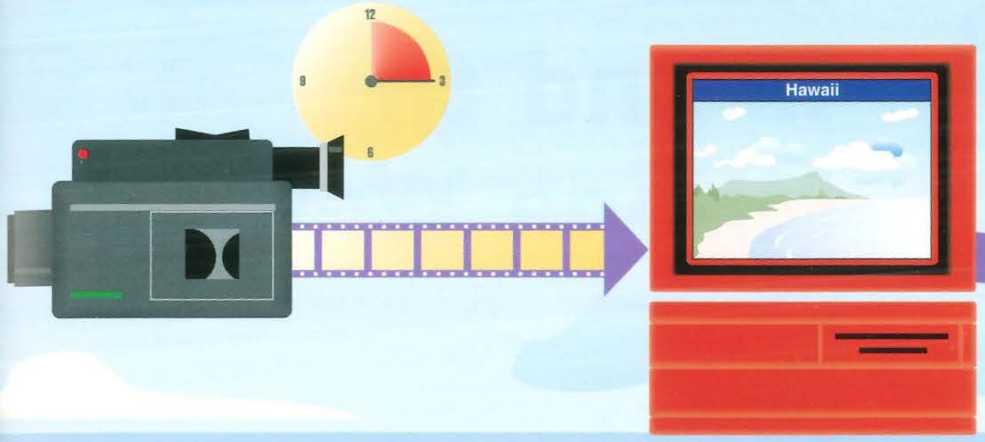


NOTE In some kinds of videoconferencing software, there is also a way to videoconference without having to go through a reflector. If you know the IP address of someone who wants to videoconference, you can connect directly to that person without having to go through a reflector. In this case, you'll only be able to participate in one-on-one conferences rather than group videoconferences.

How NetCams Work

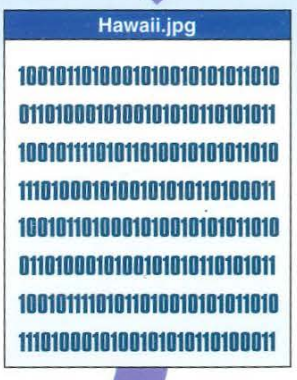


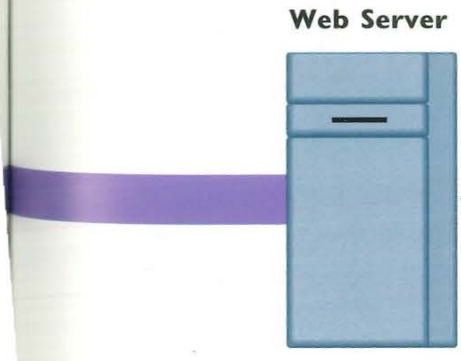
4 When someone clicks on the link, the picture is sent to his or her Web browser and displayed. Some NetCams appear to send live video images, which means the NetCam image on your computer isn't a photograph, but instead appears to be a moving image. In fact, the "moving image" is a series of photographs sent every few seconds that give the illusion of movement. When you click on the link to the image, the images will automatically be sent to your Web browser as the video camera updates them.



1 A video camera is connected to a computer on the Internet. At regular intervals, one frame of the video is sent to a computer. It can be updated once every several seconds or once a day.

2 The picture is sent to the computer attached to the Internet. A video capture board or other hardware captures the image by converting it into binary format that computers can read. Typically, the picture is converted into the JPEG format—a graphics format that compresses the image while still retaining a high level of detail.

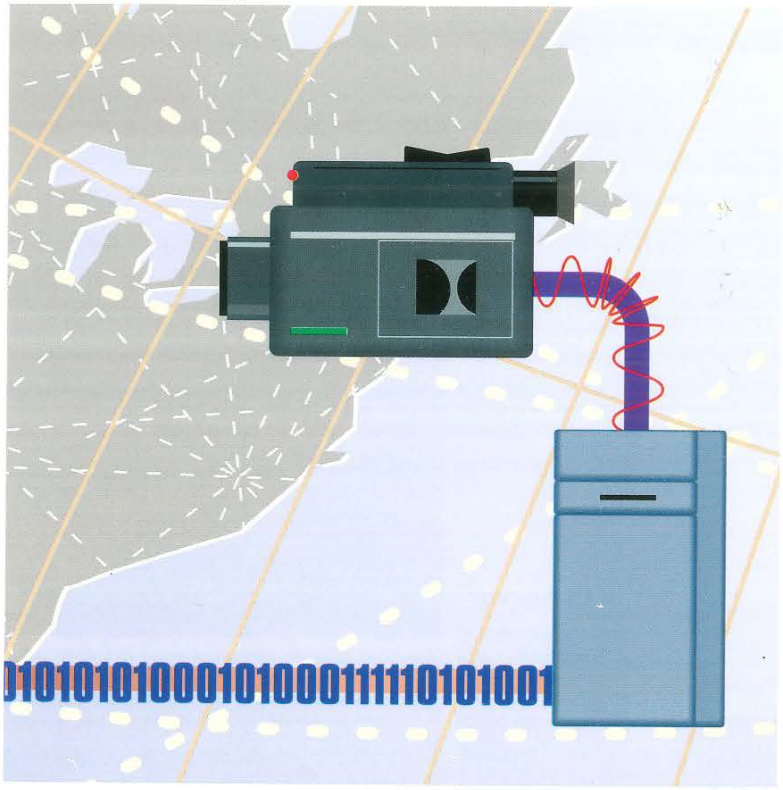




3 The JPEG image is linked to a specific URL on a Web home page. The link stays constant, even though the image itself changes regularly. That means whenever someone clicks on the link, that person will see the most recent picture that was taken by the

CHAPTER

39 How Multicast IP and the MBone Work



TO a great extent, the Internet of today is still in its infancy. Although multimedia elements can be found on it, it's still largely made up of text and static pictures. These text and static pictures individually take up very little space—an entire Web page made up of text and pictures, for example, is often only 50 kilobytes (KB) or less.

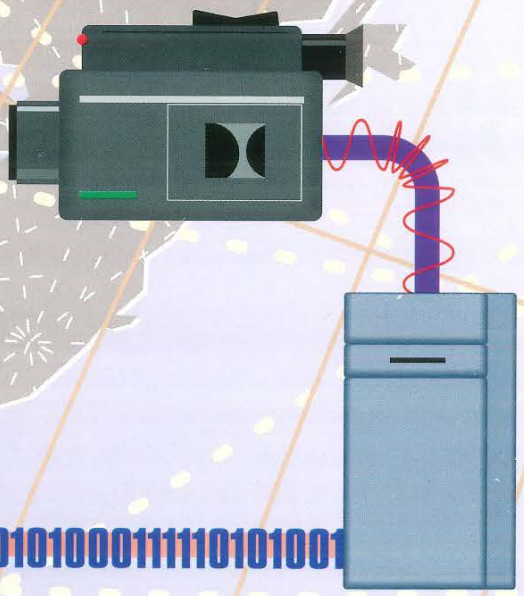
The Internet of tomorrow, however, will be made up of a wide variety of multimedia elements: sound, video, animation, 3D objects, and more. Web pages will become interactive, and video-based shows may be broadcast over the Internet.

All this will cause serious congestion on the wires and networks that make up the Internet. One more problem exists, as well: There's no practical way for broadcasts to be sent out over the Internet, because the files would clog up the Internet. Suppose, for example, that someone wants to broadcast a telecast of a concert. The size of the file that contains that broadcast may be 50 megabytes (MB). Now imagine that 10,000 people want to watch the concert. That 50MB file needs to be sent individually to each of those 10,000 people—and as you can imagine, that single broadcast could easily clog entire sections of the Internet, and the broadcast couldn't be delivered.

A potential answer is on the horizon, however, called the Multicast Backbone, or the MBone. The *MBone* is a high-capacity Internet backbone for transmitting broadcasts using the IP multicast protocol. It enables broadcasts to start out as a single transmission instead of, for example, 10,000 transmissions. Inside that transmission are the addresses of all the people who want to see the broadcast. As the file is sent across the Internet, it eventually makes copies of itself when necessary and delivers the broadcast to the networks and individuals who want to see it.

Suppose that 100 people want to see a broadcast of a 50MB file. Fifty people who want to see it are connected to the Internet via the WorldNet Internet Service Provider. 25 people are on a corporate network at `zd.com`. And another 25 use the Internet Access Company Internet Service Provider. When the broadcast goes out, it goes out as one single file, not 100 separate files. The file then splits into three parts. One part goes to WorldNet. One part goes to `zd.com`. One part goes to the Internet Access Company. After the file is on each of those separate networks, it is delivered to the people inside the networks who want to see it. The key here, though, is that instead of 100 files of 50MB traveling across the Internet—5 gigabytes (GB) of data—only three 50MB files travel, or 150MB of data. As you can see, the MBone can cut down tremendously the amount of traffic traveling across the Internet.

How Multicast IP Broadcasts Travel Along the MBone



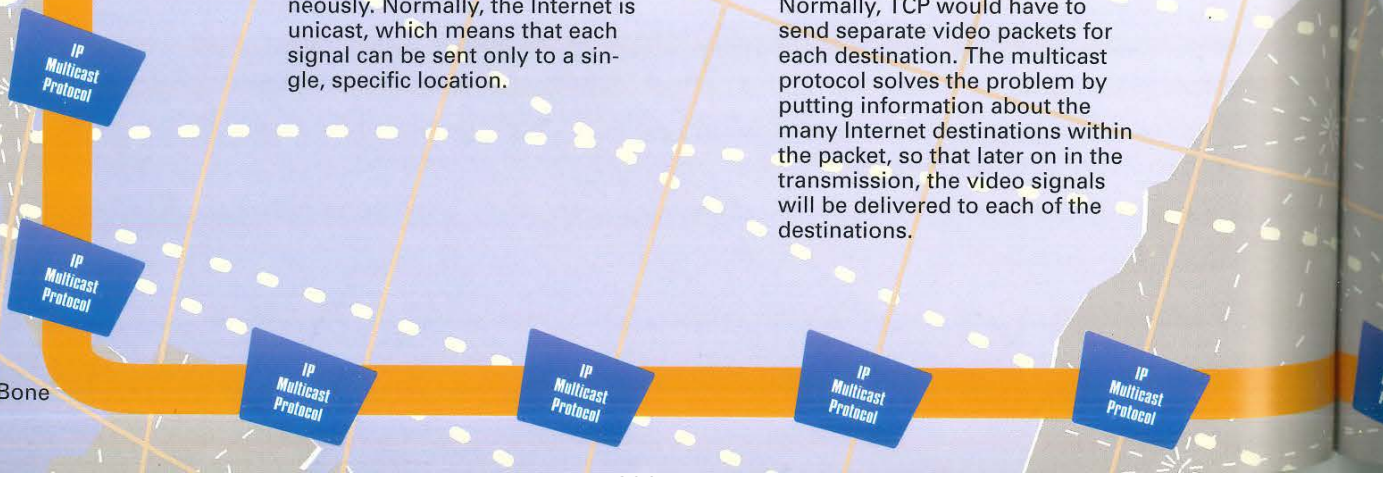
1 The MBone (Multicast Backbone) is a high-speed Internet backbone capable of sending live video and audio broadcasts. It's a network of host computers that communicate with each other using a technique called IP (Internet Protocol) Multicast. An MBone multicast begins when a video signal is digitized and compressed so that it can be sent over the Internet. Without compression, the signal would be too large and take too long to deliver.





2 The compressed, digitized signal is sent in packets using the IP multicast protocol instead of the Internet's normal TCP protocol. The multicast protocol enables the signal to be sent to a number of sites on the Internet simultaneously. Normally, the Internet is unicast, which means that each signal can be sent only to a single, specific location.

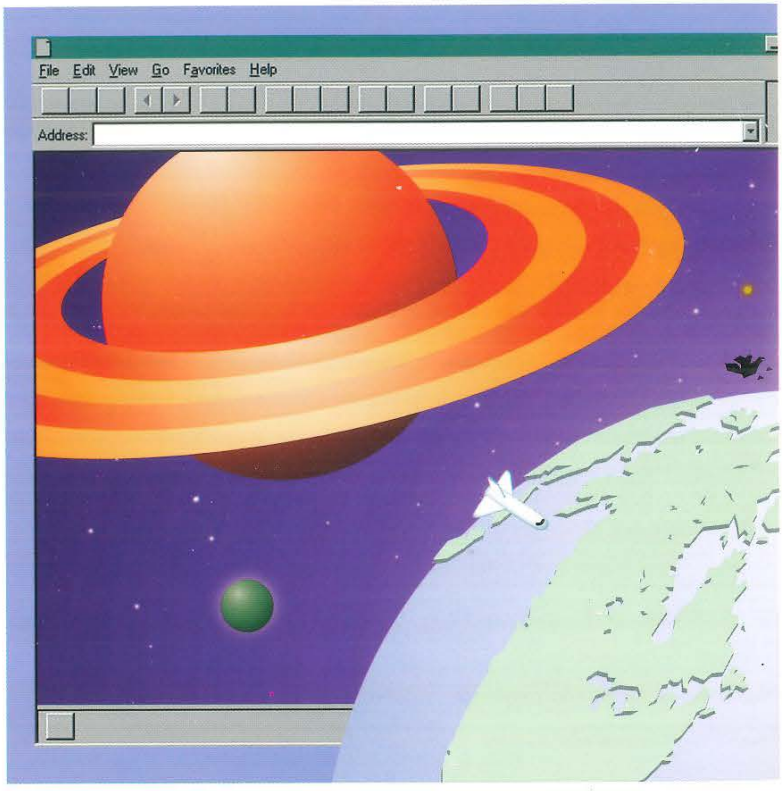
3 A major advantage of the multicast protocol is that when the video packets are sent—for example, from Europe to the United States—they are sent only one time, even though they may be sent to many destinations. Normally, TCP would have to send separate video packets for each destination. The multicast protocol solves the problem by putting information about the many Internet destinations within the packet, so that later on in the transmission, the video signals will be delivered to each of the destinations.




4
The MBone understands the multicast protocol, but most networks and routers on the Internet do not. However, the MBone network often requires that data travel along normal Internet routes. To solve the problem, the MBone data travels in "tunnels" through existing Internet networks and routers. At one end of the tunnel is a UNIX workstation that runs software called *mrouted* (multicast routing demon). This software encapsulates the multicast protocol packets inside normal TCP packets. To the Internet, the data now looks like normal TCP packets, so the data can be properly routed.
IP Multicast Protocol
MBone
MBone
5
Another mrouted program runs at the other end of the tunnel. This mrouted program breaks down the multicast protocol packets into their original form and sends them through an MBone network that understands the multicast protocol.
mrouted
TCP Packet
Internet
6
Based on the address information in the packets, the video information is delivered to a number of different hosts on the Internet. Individuals connected to the host computer can view the video and listen to the sound. In certain applications, such as teleconferencing or whiteboard applications, in which people can work together on the same file on both their screens, users can respond via video and audio as well.
mrouted

CHAPTER

40 Virtual Reality on the Internet



IMAGINE the Internet as a place where you could walk through three-dimensional worlds, pick up objects, examine them, and go to other Internet locations by flying or walking through doors. Picture home pages that were more than flat, two-dimensional surfaces that you could only read. What if you could be inside them, just like you can walk through a city or a building?

That's the promise of virtual reality (VR) on the Internet. In fact, it's more than just a promise—VR is already here. You'll find many virtual worlds you can explore on the Internet. You'll be able to walk through a giant computer, explore bizarre art galleries, visit outer space, go to the sites of what seem like ancient ruins, explore inside the human brain, and much more.

Virtual worlds are created using a computer language called Virtual Reality Modeling Language (VRML). This language instructs computers on how to build 3D geometric objects. Programmers and artists use the language to build complex worlds from these geometric objects. A VRML world is created by an ASCII text file containing VRML language commands—and for greater realism, graphics files can be added to this world as well. Because the virtual world is only an ASCII file, with perhaps a few graphics files, it can be downloaded quickly to your computer from the Internet, although some worlds with many graphics in them can be large.

When a virtual world is created, it is posted on an Internet server. When you want to visit that world, you either type in its URL or click on a link to it, just as you do to visit any other location on the World Wide Web. To display the virtual world, you'll need a program that is able to display the world—either a separate virtual reality browser, or more likely, a plug-in player that configures itself to your normal Web browser.

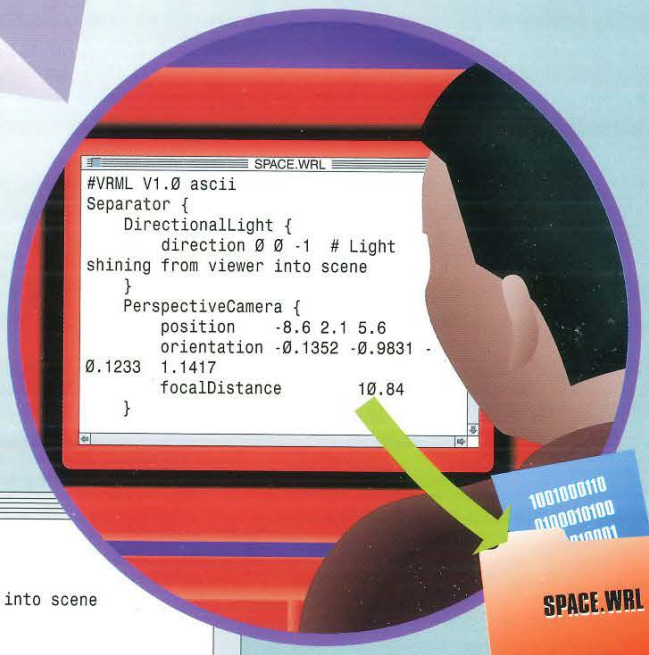
The VRML file describing the virtual world downloads to your computer. This can take a few minutes, or well over half an hour, depending on the size of the world and your connection speed. Once the file is on your computer, your CPU computes the geometry of the world, based on the VRML commands in the file. Again, depending on the size of the world and the speed of your CPU, this can take only a minute or two, or up to ten minutes or more. Once the world is computed, you can walk through it, fly through it, examine objects, and spin them. You can also visit other virtual worlds or places on the Internet by interacting with the world.

VR on the Internet is being used for far more than just creating virtual worlds that people can walk through. For example, it has been used to create views of the brain and of molecules. It has been used by astronomers to show the rotation of molecular gas in a galaxy undergoing active star formation. Finally, as with everything else related to the Internet, VR will be eventually used for things that today none of us can imagine.

Despite all its appeal, one major problem with VR worlds is that they can be very large. Because of that, they can be very slow to load and interact with due to the current limited bandwidth of the Internet. As bandwidth increases, those problems may eventually go away.

How Virtual Reality Works

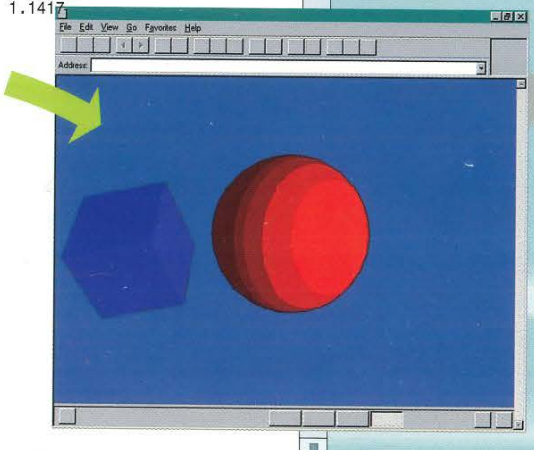
1 When someone wants to create a virtual world, he or she uses the Virtual Reality Modeling Language (VRML). VRML lets people create 3D worlds not by drawing them, but instead by using the VRML computer language to describe the geometry of a scene. VRML files are much smaller than graphics files. VRML files are simply text files that contain instructions for drawing the VRML world. VRML files end in a .WRL extension. After the world is created, it is posted on a Web server.



2 Here is an example of a VRML file describing a scene that has a red sphere and a blue cube in it, lit by a directional light.

OBJECTS.WRL

```
#VRML V1.0 ascii
Separator {
    DirectionalLight {
        direction 0 0 -1  # Light shining from viewer into scene
    }
    PerspectiveCamera {
        position    -8.6 2.1 5.6
        orientation -0.1352 -0.9831 -0.1233  1.1417
        focalDistance       10.84
    }
    Separator {   # The red sphere
        Material {
            diffuseColor 1 0 0   # Red
        }
        Translation { translation 3 0 1 }
        Sphere { radius 2.3 }
    }
    Separator {  # The blue cube
        Material {
            diffuseColor 0 0 1  # Blue
        }
        Transform {
            translation -2.4 .2 1
            rotation 0 1 1  .9
        }
        Cube {}
    }
}
```





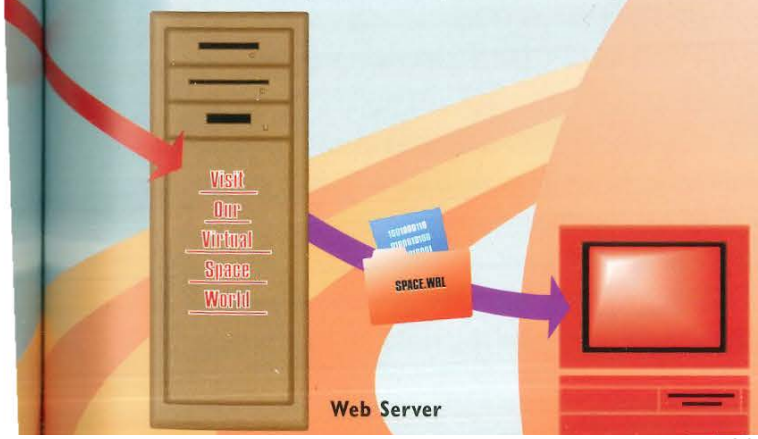
6 For greater realism and detail, graphics files can be "painted" on virtual reality objects, for example, to show paintings in an art gallery. When these graphics files are painted on objects, they must be downloaded along with the .WRL file as .GIF or .JPEG files. When the browser displays the virtual world, it shows those graphics files on top of VR objects so that they look like part of the scene.

5 Objects in the virtual world can be links to sites on the Web, to other virtual worlds, and to animations. So, for example, if you walk through a door, you might be sent to a home page on the Internet, or to another virtual world. If you're sent to another virtual world, that virtual world will have to be downloaded from a Web server to your computer so your browser can compute the new world and you can interact with it.



4 As the file downloads, the VR plug-in is launched. It doesn't run separately from your Web browser. Instead, it takes over your Web browser while you're in the virtual world. After the file is downloaded, your VR plug-in creates the virtual world by taking the VRML commands in the file and having your computer compute the geometry of the scene. Once the computation is done, the scene will appear on your screen. The VRML file contains three-dimensional information that allows you to "walk" or "fly" through the scene using your browser. Depending upon the complexity of the scene, your computer may have to do computations as you move through the scene.

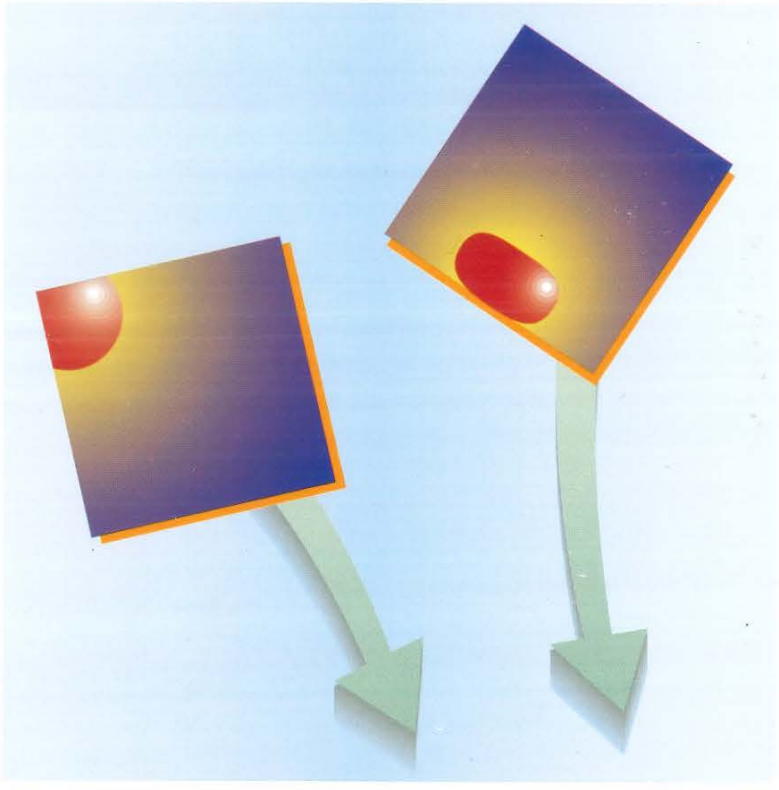




3 When you have a VR plug-in installed on your Web browser, you can visit a virtual world by clicking on its URL. First, the VRML file is sent from the Web server to your computer. Depending on the size of the virtual world and your connection speed, the file can take from a few minutes to a half-hour or more to download

CHAPTER

41 Animation on the Web



ANIMATION on the Web works no differently than animation anywhere else. Just like in a flip-book, animation is a series of still images displayed in succession to create an illusion of fluid motion. The faster the frames advance, the more fluid the animation becomes. Unfortunately, the Web can be a very slow place, and an animation that should run quickly often crawls across the screen unless special technology is used.

There are a number of different ways to create Web animations, including client pull, server push, animated GIFs, and the Shockwave multimedia plug-in. In *client pull*, an HTML page gives the browser instructions to request and load another document automatically. This feature is like a slideshow. Web pages are displayed one after the other with a specified time delay in between. This is useful for step-by-step instructions. But client pull is slowed by the need to load a whole page rather than a single cell of animation, which prevents the illusion of fluid animation.

Client pull requests are embedded within the HTTP response header of a Web page that is sent back from the server to the client. The <META> tag inserts meta-information into a response header. *Meta-information* is used to help parse a Web page, but it is not displayed by the browser. A *response header* is the beginning of each HTTP response that a server sends back to a client with the requested Web page.

Server push is a complement to client pull, though server push is the more complex of the two. Server push requires a CGI (Common Gateway Interface) script that tells the server when to automatically serve a new document or image. It requires the client browser to be able to recognize the MIME-type called multipart/x-mixed-replace. This MIME-type allows for multiple documents to be sent within one message. To understand how server push works, imagine an email message with text, hypertext, a digital movie and sound, and you can see how multiple "documents" (media types) can be sent within a single message. The multipart message is simply a series of images that displays one right after the other, each being sent or "pushed" by the server. In this way, a small animation can be embedded among the text and images of an otherwise static Web page.

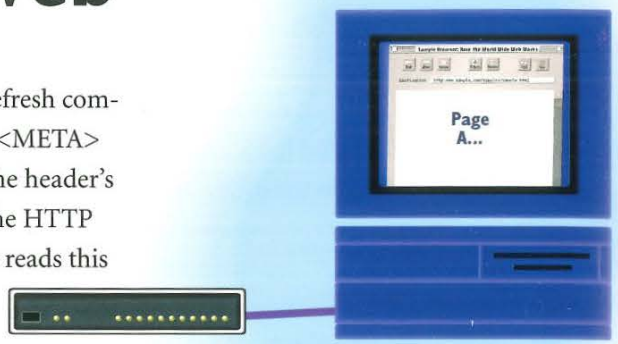
Animated GIFs are a series of graphical GIF images that "roll up" into a single image—much like the flip book "animates" a series of drawings as you thumb through the pages. They load into a browser just like any GIF file, but in a series to give the illusion of motion. Animated GIFs have the benefit of speed because images are cached on the client PC and loaded from memory rather than from the Internet. They represent an easy solution to adding motion to Web pages.

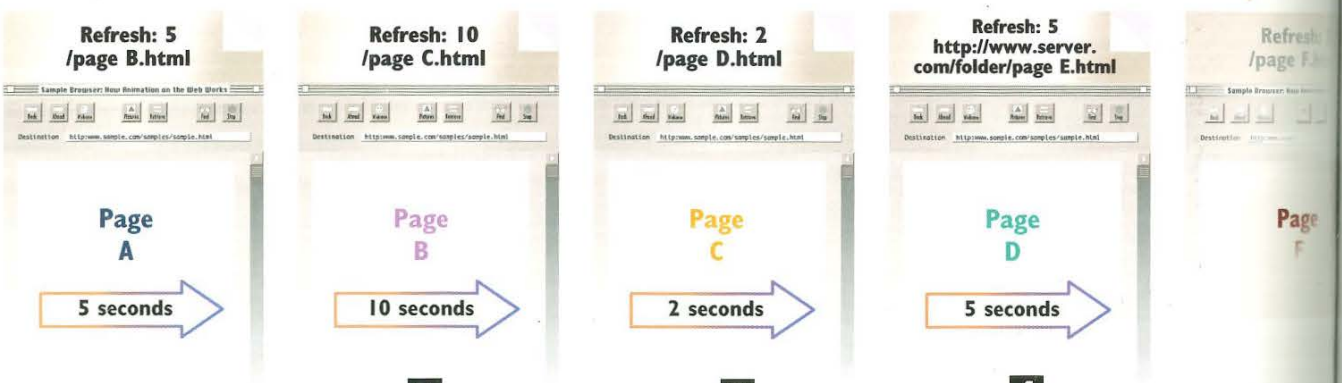
More complex multimedia animation has become possible using Macromedia's Shockwave plug-in. Shockwave plays multimedia files that are created with Macromedia's popular Director and Authorware programs. You must first download and install the Shockwave plug-in before you can view any Web pages that have Shockwave animations.

How Animation Works on the Web

Client Pull

Client pull is executed by the Refresh command. A refresh command is written into an HTML document using the <META> tag. The contents of the <META> tag are added to the header's meta-information sent from the server along with the HTTP response. During a client pull sequence, the browser reads this header information, which instructs it to keep track of the time elapsed between pages retrieved with the help of your PC's internal clock. When the time has elapsed, the browser requests and displays the next page.





1. The Refresh command does two things. First, it indicates the time before the next page request is sent or the same page reloads. For example, page A in the illustration will refresh after five seconds. Secondly, if the URL follows the number of seconds, a request for that page will be sent automatically after five seconds. After the browser parses a document's meta-information and recognizes the Refresh command in the header, it knows to send a request for the page indicated by the URL following the command.

2. If the next document to load also has a Refresh command in the header, the browser will simply repeat the process. In this case, it will retrieve and display page C after 10 seconds.

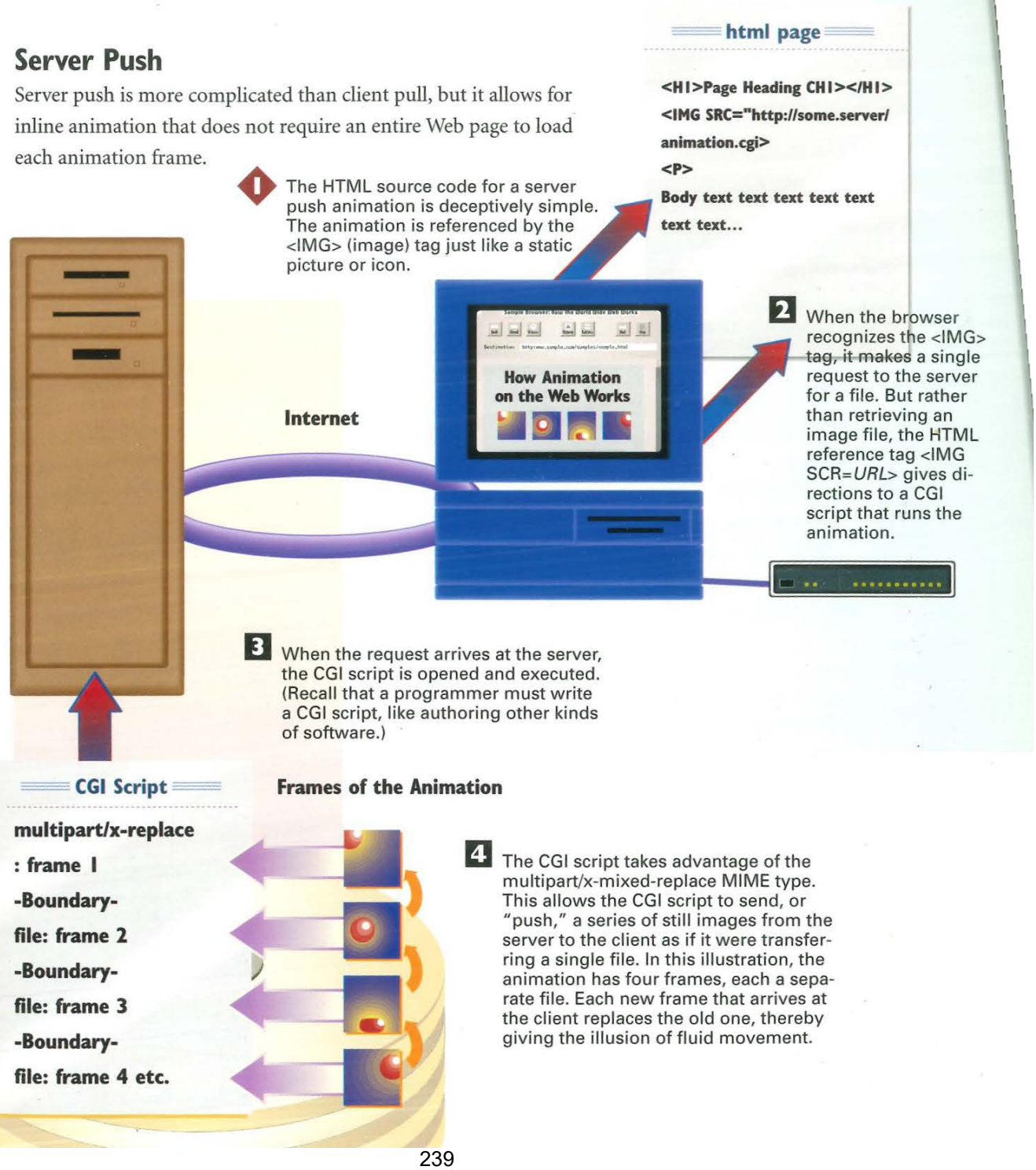
3. Whoever writes the HTML source code can specify how long it will be until the request for the next page is made. Page C will refresh after only two seconds, followed by page D.

4. Each page in a client pull sequence can be located anywhere on the Web. The URL following the Refresh command may lead the browser to any active server. Page E is located on a different server than pages A to D but will still be requested automatically after five seconds.

NOTE A client pull sequence may continue for as many or as few pages as the site designer wants. The last page will simply not have a Refresh command in the header. A user may stop the process manually by clicking the browser's Stop button.

Server Push

Server push is more complicated than client pull, but it allows for inline animation that does not require an entire Web page to load each animation frame.

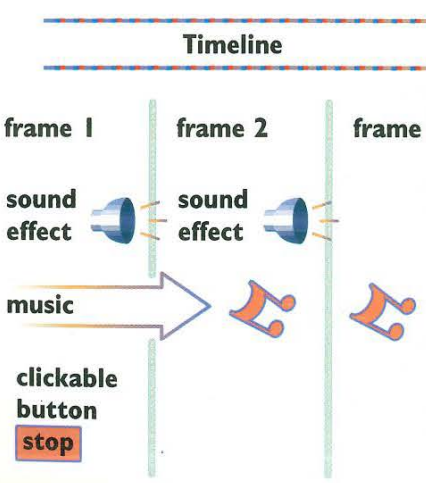


How Shockwave Works

1 The first step in a Shockwave animation happens in a multimedia authoring program such as Director or Authorware. An animation designer must gather the raw materials necessary for a short and compelling animation, such as still images, music, and sound effects.



2 The authoring program then helps arrange the elements frame by frame along a time line. It also allows the designer to match a sound effect with a particular action in the animation.

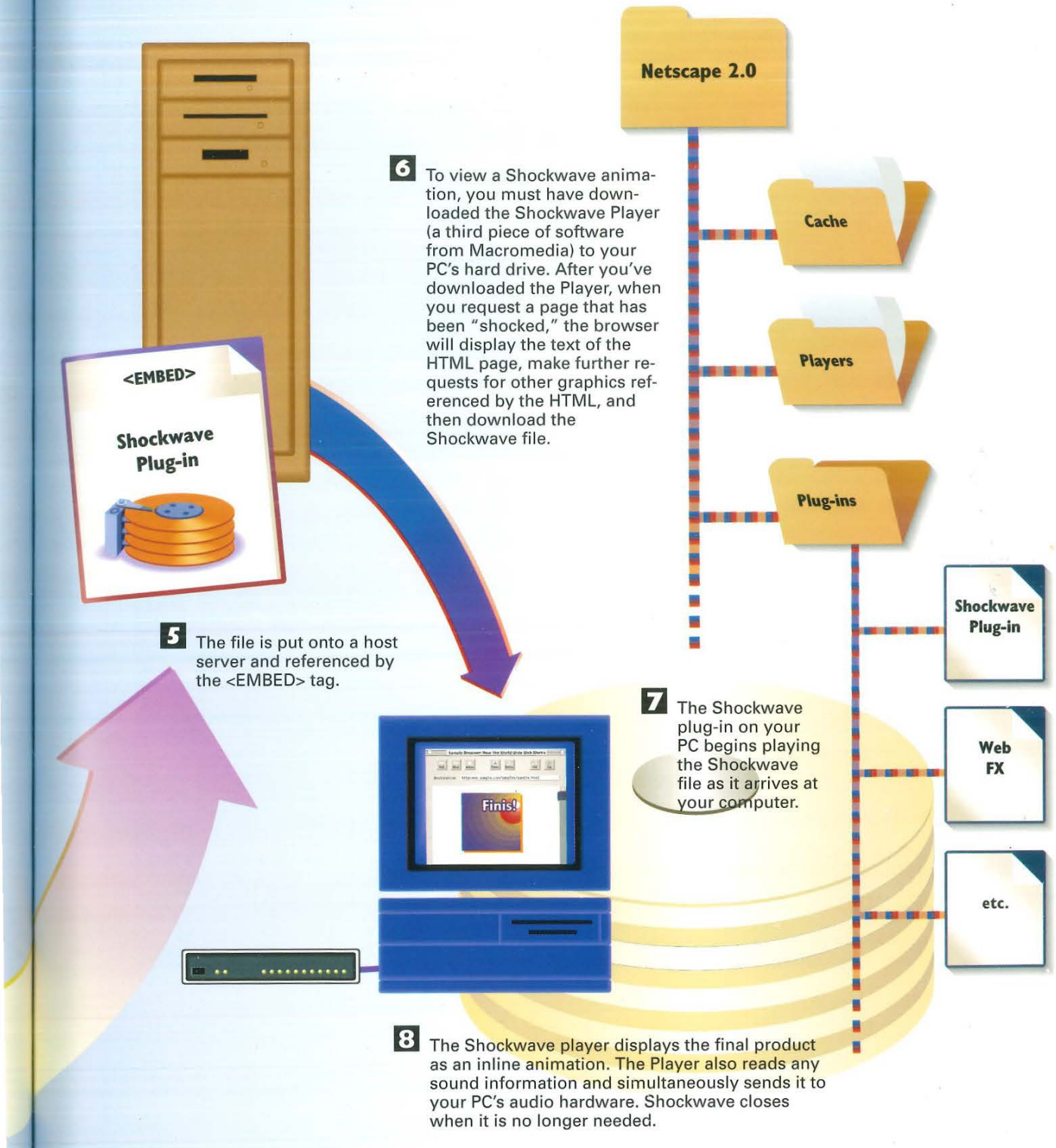


3 When this step is done, the complete animation is saved as a Director or Authorware movie file.



4 Next, the movie file must be converted and compressed into a small file that can be quickly downloaded to a user's PC. This conversion process is done with Afterburner, another Macromedia product.



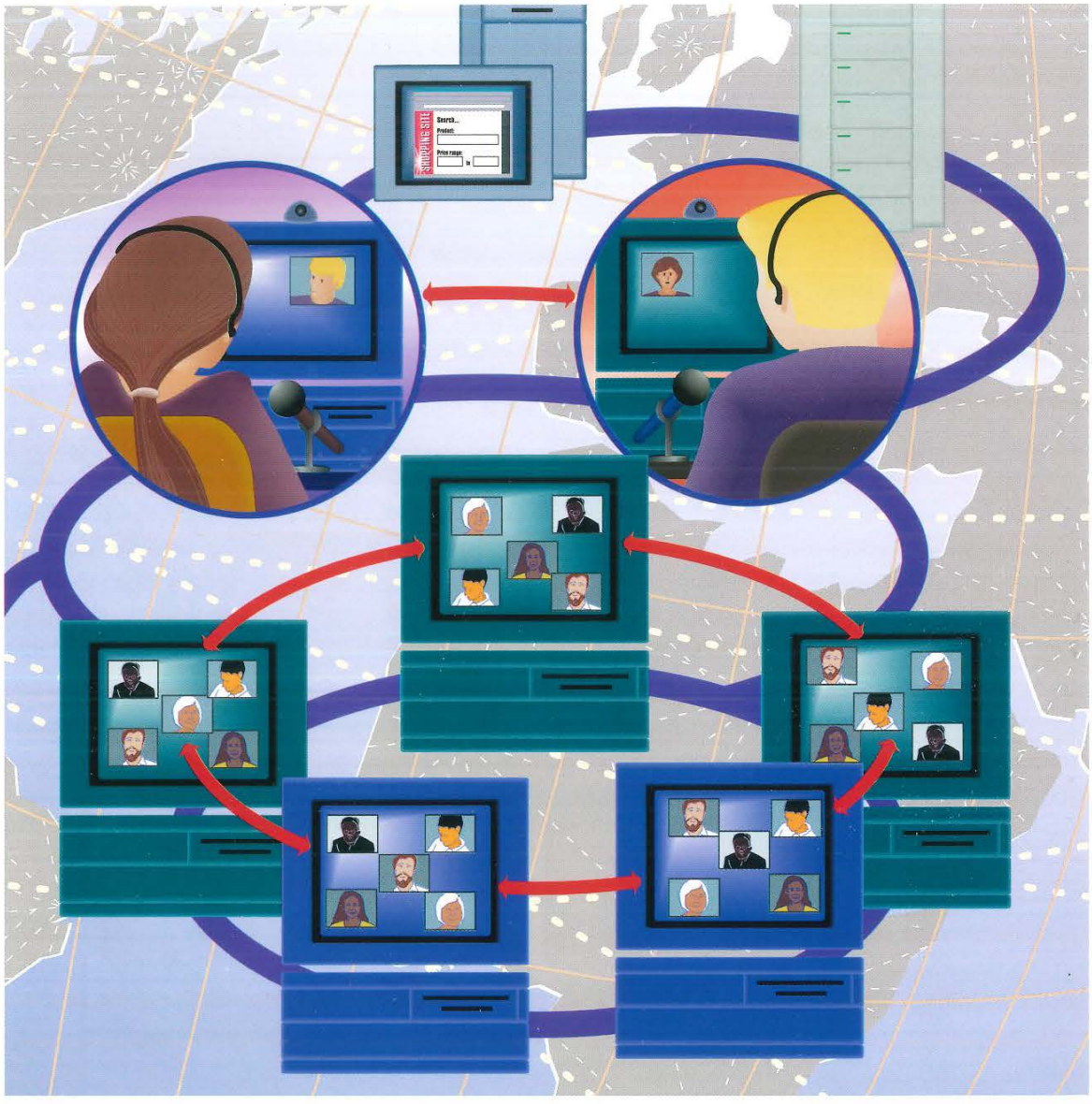


6 To view a Shockwave animation, you must have downloaded the Shockwave Player (a third piece of software from Macromedia) to your PC's hard drive. After you've downloaded the Player, when you request a page that has been "shocked," the browser will display the text of the HTML page, make further requests for other graphics referenced by the HTML, and then download the Shockwave file.

5 The file is put onto a host server and referenced by the <EMBED> tag.

7 The Shockwave plug-in on your PC begins playing the Shockwave file as it arrives at your computer.

8 The Shockwave player displays the final product as an inline animation. The Player also reads any sound information and simultaneously sends it to your PC's audio hardware. Shockwave closes when it is no longer needed.


SHOPPING SITE
Search...
Product:
Price range:
to

PART 9

THE INTERNET, INTRANETS, AND THE OUTSIDE WORLD

Chapter 42: How Intranets Work
246

Chapter 43: How Workgroup Software Works
250

Chapter 44: Shopping on the Internet
254

THE Internet is no longer a self-enclosed club with no connection to the outside world. It has become intimately tied to the way we live and work, and it will become more so with each passing year. At work, for entertainment, to get information, to shop—the Internet is increasingly becoming a part of our daily lives.

The Internet may have its roots in the military and in academia, but its dramatic growth has been fueled in large part by business and consumers. The Internet may become one of the primary places in which businesses operate, and where hundreds of billions of dollars in goods and services will be bought and sold every year.

Thousands of businesses use the Internet to market and sell their products, and so many people buy things from home and from their places of business through the Internet instead of at retail stores. You can use the Internet to browse through catalogs and make purchases online; to buy and sell stock, mortgages and insurance; and even to participate in online auctions. Companies are figuring out ways not only to sell online, but also to hook those online transactions into their internal computer and billing systems.



This part of the book looks at the various ways that the Internet interacts with the outside world. It covers how the Internet is being used by businesses as their primary corporate network, and how business and commerce is being conducted on the Internet.

Chapter 42, "How Intranets Work," covers intranets. *Intranets* are private networks set up by companies for their employees, using Internet technology. Intranets are used for many purposes, including email, group brainstorming, group scheduling, and access to corporate databases and documents, among others.

Although intranets use TCP/IP networks and technologies, the network and its resources are used only by the businesses and are not available to people outside the company. Intranets are separated from the Internet by firewalls that don't allow unauthorized access to the intranet. People who work in the company can access the intranet and use its resources, but intruders are kept out by the firewalls.

Chapter 43, "How Workgroup Software Works," looks at one of the most important parts of an intranet—*workgroup software.* This kind of software ties together everyone in a corporation and enables them to work together better. Among other things, it enables people to share files and information; to cooperate more easily on projects; and in general, to work together in ways never before possible. It allows people to go beyond simply communicating, and enables them to work together on shared documents.

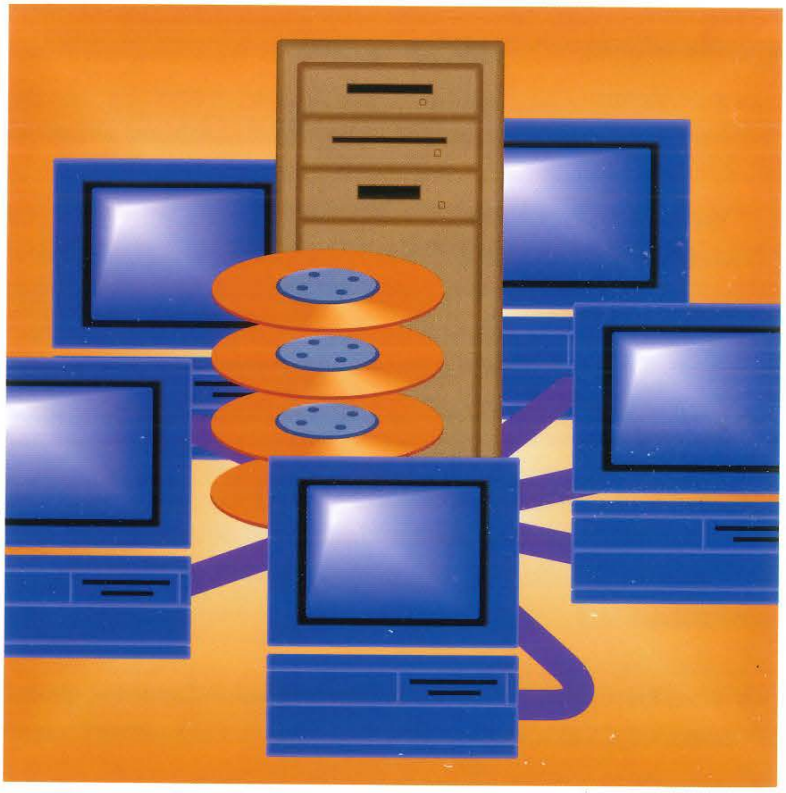
Chapter 43 looks at a variety of workgroup software. It covers messaging software that enables people to publicly participate in group discussions. It also looks at *whiteboard software*, which allows people to see what is on other people's computers and work together on documents. Several people could look at a spreadsheet together, for example, and one person could mark up the spreadsheet while everyone else sees what is being done.

Chapter 44, "Shopping on the Internet," covers shopping. Shopping on the Internet already accounts for billions of dollars a year in revenue, and every year many more billions are spent. One problem remains, however; by its very nature, the Internet is an insecure network, and so ways must be found to safeguard information, such as credit card numbers. Chapter 44 looks at how credit card transactions can be done securely across the Internet, and it looks at technologies such as electronic cash, also called Ecash, and anonymous payments.



CHAPTER

42 How Intranets Work



BUSINESSES will increasingly use Internet technology to create private corporate networks called *intranets* that will replace current local area networks and will be the prime computing resources of corporations.

Corporations can set up intranets for a wide variety of purposes, including email, group brainstorming, group scheduling, access to corporate databases and documents, video conferencing, as well as buying and selling goods and services.

Intranets use TCP/IP networks and technologies as well as Internet resources such as the World Wide Web, email, Telnet, and FTP. However, the network and its resources are used privately by businesses and are not available to people outside the company. An intranet is separated from the rest of the Internet by a *firewall*—a hardware and software combination that prohibits unauthorized access to the intranet. People who work in the company can access the Internet and use its resources, but intruders are kept out by the firewalls. Turn to Chapter 45, "How Firewalls Work," to learn more about firewalls.

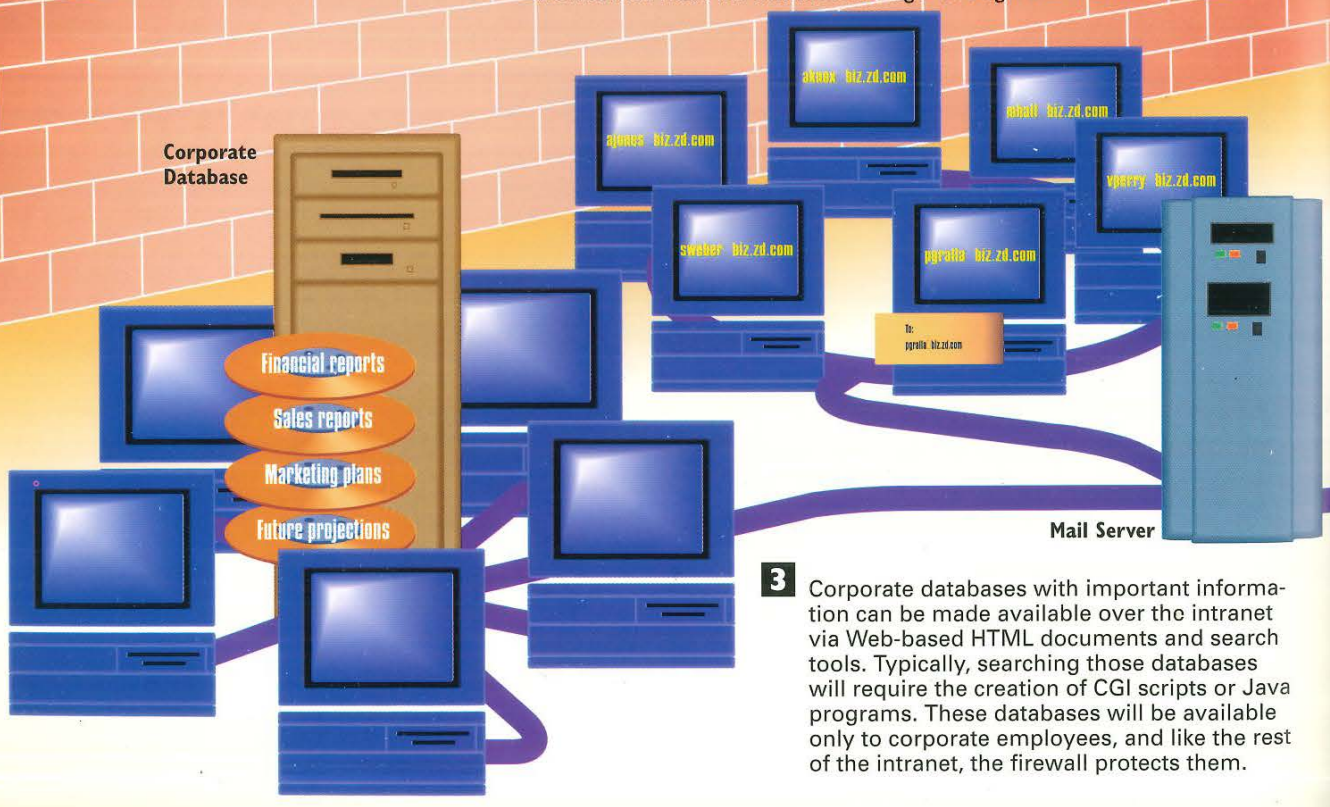
Intranets use a combination of off-the-shelf software such as Web browsers, and customized software such as database querying tools. Because intranets are based on Internet standard protocols, it will always be possible to quickly update them with the latest in network technologies.

In the long term, companies will make the most use of intranets in *workgroup applications*—software that allows people to work cooperatively with their computers. There are many different kinds of workgroup software. These programs allow people to participate in discussions and videoconferencing across the country and across the world, share databases, track documents, and much more. For details on workgroup software, turn to Chapter 43, "How Workgroup Software Works."

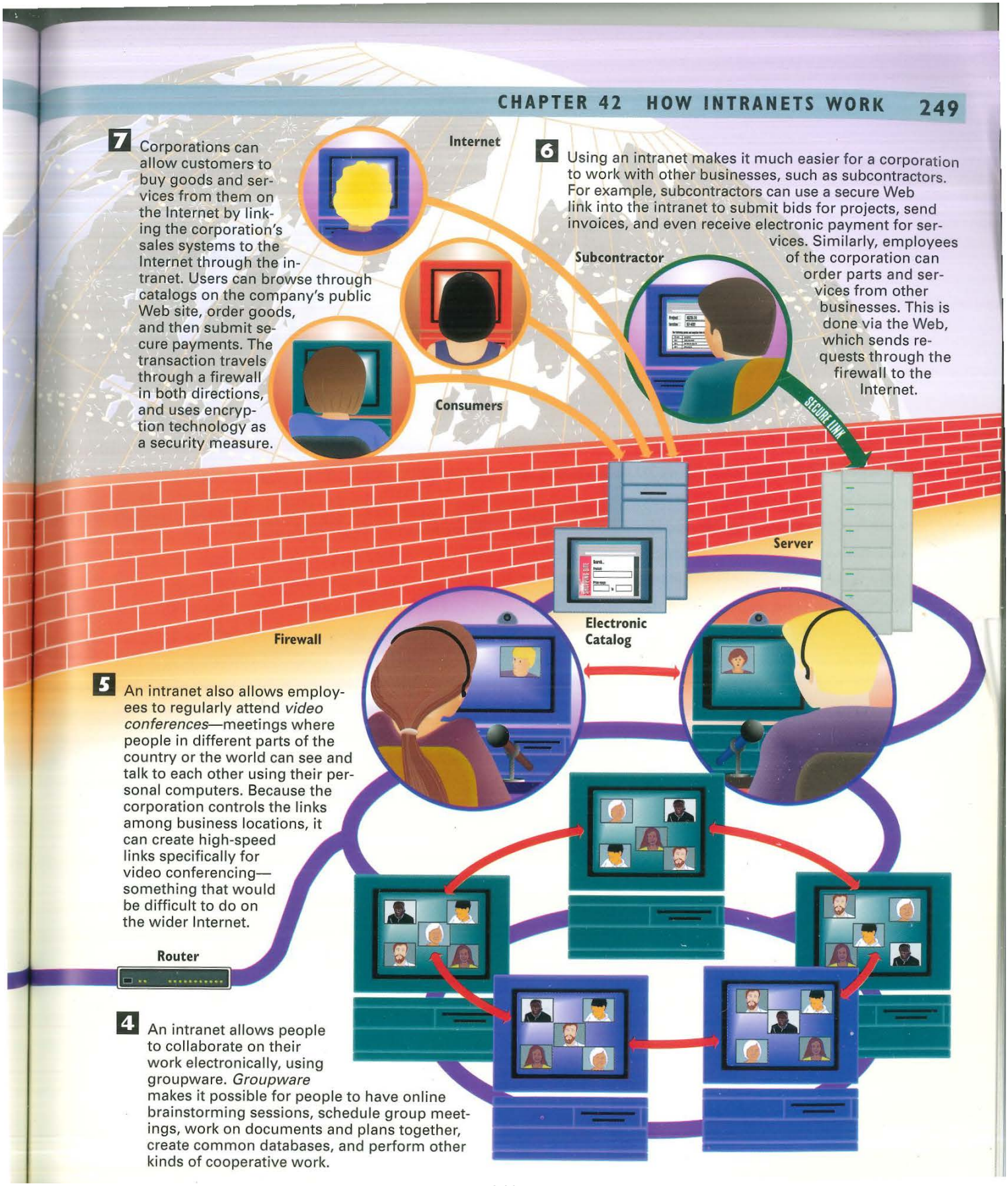
Using an Intranet Within a Company

1 An intranet is separated from the rest of the Internet by a *firewall*—a hardware/software combination that protects the corporate intranet from snooping eyes and malicious attacks. The firewall allows corporate employees to use the Internet and also allows certain parts of the intranet—such as areas designed for electronic commerce—to be accessed by outsiders.

2 A key component of an intranet is an internal email system. The email system works just like Internet email. It can use normal Internet email clients, except that it is designed to route traffic within an organization so the email need not travel outside the intranet. Internal routers and mail servers send the mail to other corporate employees via the intranet. Email that travels to and from the Internet from the intranet must go through the firewall.



3 Corporate databases with important information can be made available over the intranet via Web-based HTML documents and search tools. Typically, searching those databases will require the creation of CGI scripts or Java programs. These databases will be available only to corporate employees, and like the rest of the intranet, the firewall protects them.



7 Corporations can allow customers to buy goods and services from them on the Internet by linking the corporation's sales systems to the Internet through the intranet. Users can browse through catalogs on the company's public Web site, order goods, and then submit secure payments. The transaction travels through a firewall in both directions, and uses encryption technology as a security measure.

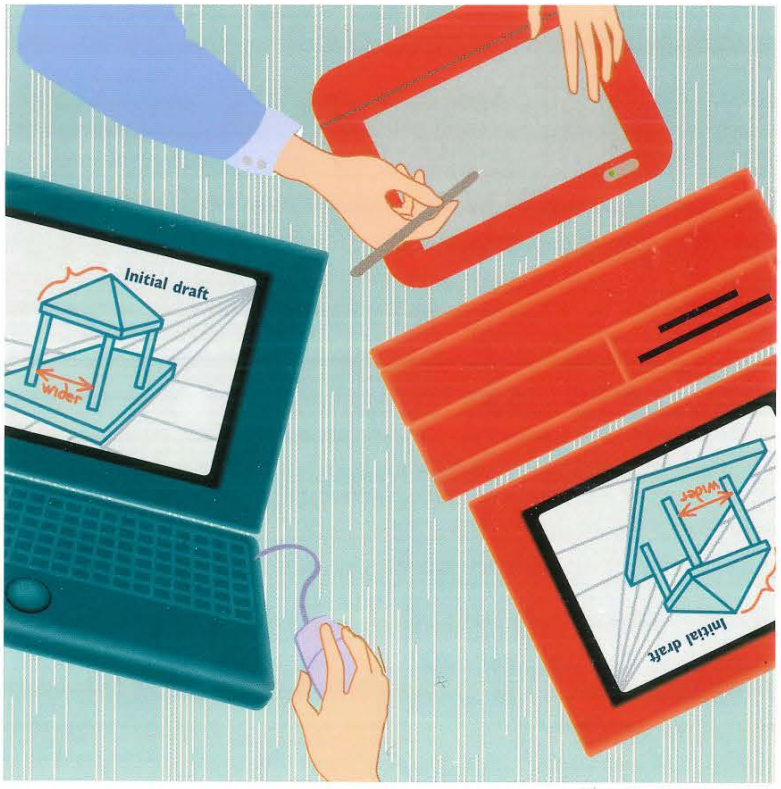
6 Using an intranet makes it much easier for a corporation to work with other businesses, such as subcontractors. For example, subcontractors can use a secure Web link into the intranet to submit bids for projects, send invoices, and even receive electronic payment for services. Similarly, employees of the corporation can order parts and services from other businesses. This is done via the Web, which sends requests through the firewall to the Internet.

5 An intranet also allows employees to regularly attend *video conferences*—meetings where people in different parts of the country or the world can see and talk to each other using their personal computers. Because the corporation controls the links among business locations, it can create high-speed links specifically for video conferencing—something that would be difficult to do on the wider Internet.

4 An intranet allows people to collaborate on their work electronically, using groupware. *Groupware* makes it possible for people to have online brainstorming sessions, schedule group meetings, work on documents and plans together, create common databases, and perform other kinds of cooperative work.

CHAPTER

43 How Workgroup Software Works



WORKGROUP software, also called groupware, is at the heart of intranets. It lets people share files and information; cooperate more easily on projects; and in general, work together in ways never before possible. The key is that it allows people to go beyond simply communicating and lets them work together on shared documents.

One of the most basic pieces of workgroup software is *messaging software*—programs that allow people to publicly participate in group discussions. These group discussions are threaded, which means that people can read and respond to individual subject areas of a discussion. For example, in a message area devoted to corporate finances, there may be one thread concerning research and development finances and another concerning engineering finances.

What makes intranet messaging software especially useful is the way it integrates with other Internet and intranet technologies. For example, some discussion software will allow the use of Hypertext Markup Language (HTML) embedded inside messages. This means that from within a discussion, someone could embed a link to a Web page or other intranet resource.

A more sophisticated workgroup application is desktop videoconferencing. It requires that everyone involved have computer-linked video cameras (which have become quite inexpensive) as well as hardware and software that allows computers to send and receive voice and sound. While sitting at computers, people can see each other and speak to each other.

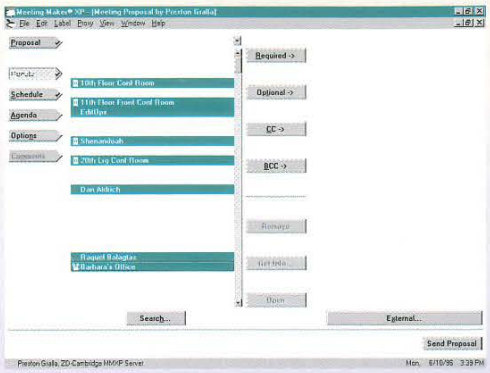
A related technology is called whiteboard software. *Whiteboard software* lets people see what is on someone else's computer on an intranet while sitting at their own computer. More importantly, whiteboard software allows people to use a mouse to highlight parts of the screen, write on the screen, and otherwise mark it up. This means people on the same intranet—even if they're on opposite sides of the country from each other—can comment on each other's work easily.

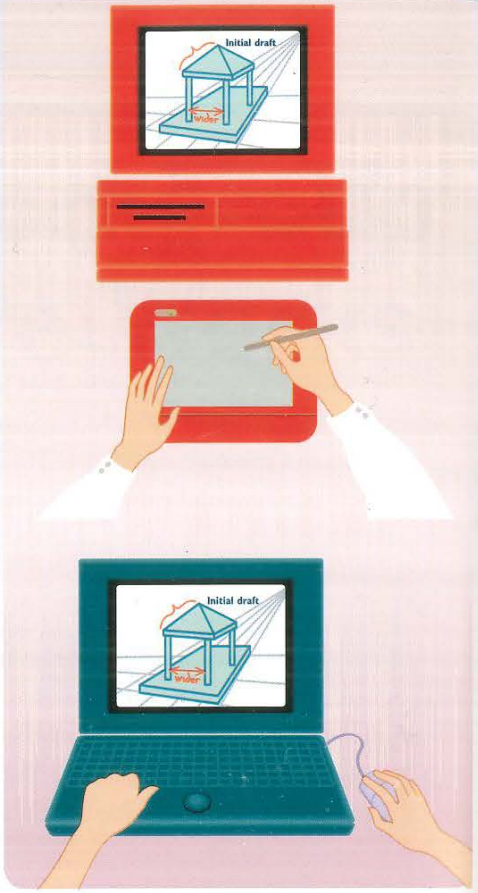
Document management software and work flow groupware are useful for intranets in companies that have complicated work procedures, or where many people must cooperatively put together a single document. In intranet document management software, a document can be "locked" so that only one person at a time can use it, and so people can't overwrite each other's work. It can also give different people different kinds of rights to a particular document, so that some people may only be able to read documents, while others are able to actually work on them, edit them, and otherwise change them. The most powerful document management software allows many different people to work on different parts of the same document simultaneously.

Work flow groupware is similar to document management software. This kind of groupware manages the entire work flow of an organization in addition to handling individual documents. For example, intranet work flow groupware allows procedures for filing expense reports to be easily computerized.

How Workgroup Software Works

1 Discussion software allows people from within a corporation to exchange work and ideas. Included in the software are links to other intranet resources. From within a discussion, people can link out to a Web page on the Internet or intranet, or even link into intranet databases, servers, or shared applications. Additionally, software can replicate intranet discussions onto Internet newsgroups. From one discussion area, people on an intranet can hold discussions with people from within their company or people out on the Internet.





5 Whiteboard applications will be a popular workgroup use of intranets. In a whiteboard application, two or more people can see what is on each other's computer screens across the intranet and they can talk about what they see. Additionally, they can mark up what is displayed on each other's screens.



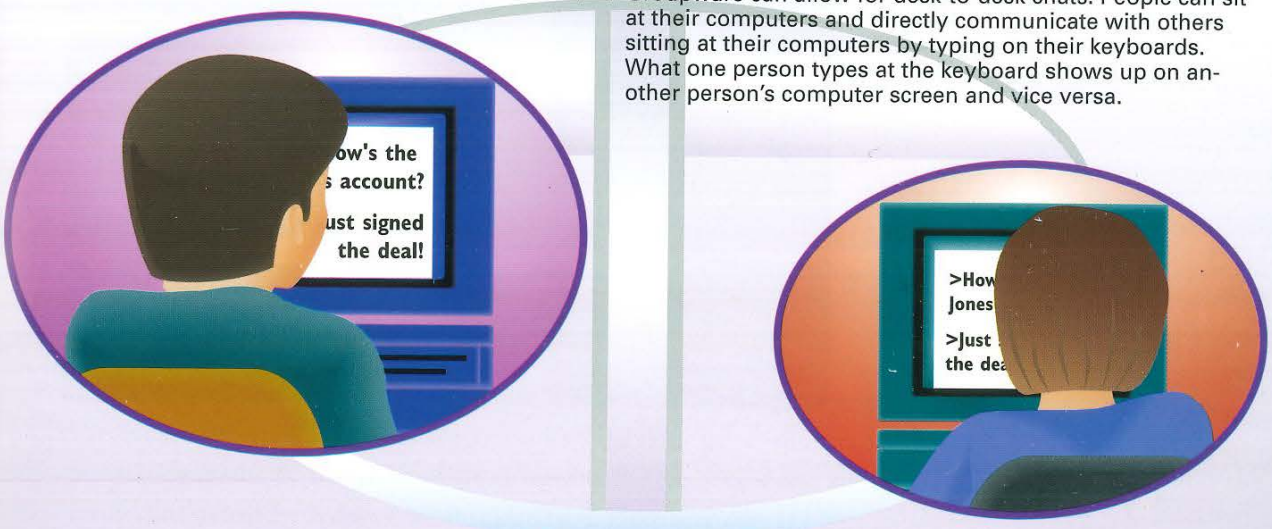


4 With intranet groupware, videoconferencing can finally be a corporate reality. Desktop conferencing software allows two or more people to see each other and talk to each other on their computer screens, as long as they have cameras connected to their computers and sound-equipped computers. Because intranets can be built using very high bandwidth connections, it's possible to have a videoconference across an intranet, while it can be much more difficult to do it across the Internet because of the lower bandwidth of the Internet.



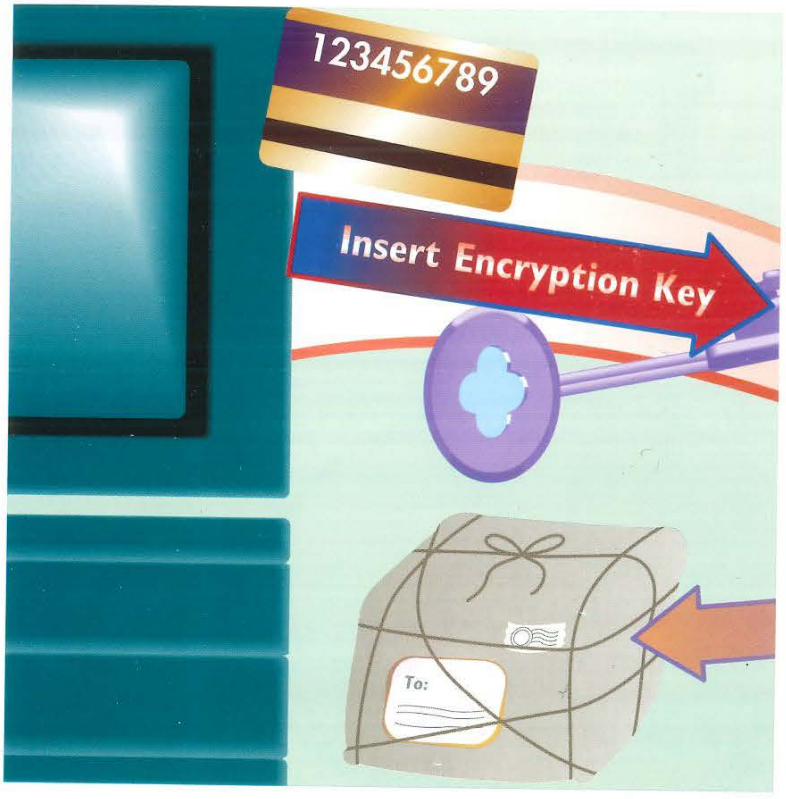
2 Document management software and work flow groupware allows intranet administrators to create systems that track and control access to documents through every aspect of their creation—for example, allowing only one person at a time to check a document out of a library. They can provide a "version history" of every document so anyone can see who has worked on it along with the changes that person made. The administrators can also give certain people the right to "lock" the document so no further changes are allowed.

3 Groupware can allow for desk-to-desk chats. People can sit at their computers and directly communicate with others sitting at their computers by typing on their keyboards. What one person types at the keyboard shows up on another person's computer screen and vice versa.



CHAPTER

44 Shopping on the Internet



BILLIONS of dollars are expected to be spent every year shopping on the Internet. You can already buy almost anything on the Internet—whether it be a book or a new car. Some day, buying over the Internet may become as common as shopping by mail or even in retail stores.

There's one problem with online shopping: The Internet is an insecure network. As packets travel across it, anyone along the way could conceivably examine those packets. Because of that, there are potential dangers to doing business online—if you pay over the Internet with a credit card, someone could snoop at it and steal your credit card number.

A number of ways to make payments across the Internet have sprung up to solve this problem. Most use procedures and protocols designed to make financial transactions on the Internet as confidential as possible, using encryption technology to make sure that no one can steal your credit card number. Dozens of organizations, including Microsoft, Netscape, the major credit card companies, automated finance services, and Internet and Web standards-setting bodies, are developing schemes to provide secure, encrypted financial transactions over the Internet.

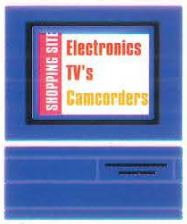
Schemes for secure transactions take two approaches. One approach encrypts personal financial information, such as a credit card number, so that it can be transferred across the Internet in a manner that would not let prying eyes read the data. The second method creates a system of *cyber-dollars*, electronic credits that only authorized merchants can redeem for real money.

The Secure Electronic Transaction protocol (SET) has been endorsed by VISA, MasterCard, American Express, Microsoft, and Netscape, among other companies. It describes a way that people can shop online and have the purchases charged to their credit cards. Because this method of payment has gotten backing from so many major software and financial companies, there is a good chance it will become a standard for buying online in the future.

In addition to secured credit card transactions, a number of companies are working on electronic, or "cyber-dollar," scenarios that will enable consumers to purchase goods and services anonymously. That is, the consumer uses the digital equivalent of paper currency to make purchases and need not provide personal information (such as credit card or bank information) to do so. Using this method of electronic payment, consumers buy electronic "coins" or "tokens" and use these specially marked and encrypted coins to make purchases.

Both credit card systems and electronic cash systems have their advantages and disadvantages. Credit card encryption schemes are convenient and don't ask customers to change their usual purchasing methods. Transactions are charged to the credit card and appear on the customer's credit card statement as does any other purchase. Electronic cash is a little less convenient because customers have to buy electronic currency before they can use it. However, electronic currency also brings a certain amount of privacy to electronic purchases.

How Financial Transactions Work on the Internet



1 Gabriel browses through an electronic catalog on a Web site and he decides to buy a camcorder. To use the Secure Electronic Transaction protocol (SET) to pay for it, he will need a credit card from a participating bank and a unique "electronic signature" for his computer; this will verify that he is making the purchase, not an impostor. (In SET, everyone involved in the transaction needs electronic signatures identifying them.) SET also uses public-key encryption technology to encrypt all the information sent between everyone involved in the transaction.



2 Gabriel fills out an order form detailing what he wants to buy, its price, any shipping and handling fees, and taxes. He then selects the method he wants to use to pay. In this case, he decides to pay electronically over the Internet. At this point, he doesn't send his precise credit card number, but instead indicates which credit card he wants to use. The information he sends includes his electronic signature so the merchant can verify it is really Gabriel who wants to do the ordering.



3 The merchant receives the order form from Gabriel. The merchant's software creates a unique transaction identifier so the transaction can be identified and tracked. The merchant sends this identifier back to Gabriel along with two "electronic certificates," which are required to complete the transaction for his specific bank card. One certificate identifies the merchant and the other certificate identifies a specific *payment gateway*—an electronic gateway to the banking system that processes online payments.



4 Gabriel's software receives the electronic certificates and uses them to create Order Information (OI) and Payment Instructions (PI). It encrypts these messages and includes Gabriel's electronic signature in them. The OI and the PI are then sent back to the merchant.



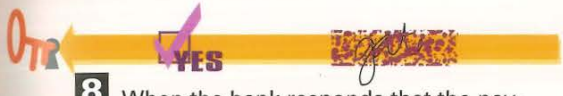
5 The merchant's software decrypts Gabriel's OI and uses the electronic signature that Gabriel sent to verify that the order is from him. The merchant sends verification to Gabriel that the order has been made.



6 The merchant's software creates an authorization request for payment and includes with the merchant's digital signature the transaction identifier and the PI received from Gabriel. He encrypts all of it and sends the encrypted request to the payment gateway.



7 The payment gateway decrypts the messages and uses the merchant's digital signature to verify that the message is from the merchant. By examining the PI, it verifies that they have come from Gabriel. The payment gateway then uses a bank card payment system to send an authorization request to the bank that issued Gabriel his bank card, asking if the purchase can be made.



8 When the bank responds that the payment can be made, the payment gateway creates, digitally signs, and encrypts an authorization message, which is sent to the merchant. The merchant's software decrypts the message and uses the digital signature to verify that it came from the payment gateway. Assured of payment, the merchant now ships the camcorder to Gabriel.



9 Some time after the transaction has been completed, the merchant requests payment from the bank. The merchant's software creates a *capture request*, which includes the amount of the transaction, the transaction identifier, a digital signature, and other information about the transaction. The information is encrypted and sent to the payment gateway.



10 The payment gateway decrypts the capture request and uses the digital signature to verify it is from the merchant. It sends a request for payment to the bank, using the bank card payment system. It receives a message authorizing payment, encrypts the message, and then sends the authorization to the merchant.



11 The merchant software decrypts the authorization and verifies that it is from the payment gateway. The software then stores the authorization that will be used to reconcile the credit card payment routinely when it is received from the bank.



PAYMENT GATEWAY PAYMENT GATEWAY PAYMENT GATEWAY PAYMENT

BANK BANK BANK BANK BANK BANK BANK BANK

LEGEND

- Internet
- ORDER FORM — Order form
- Gabriel's digital signature
- Merchant digital signature
- Payment gateway electronic signature
- ID — Unique transaction identifier
- 101100 — Merchant electronic certificate
- 101010 — Payment gateway electronic certificate
- Encryption key
- 411 — Order information
- $ — Payment/instructions
- YES / NO — Authorization request
- YES — Authorization message
- CAPTURE REQUEST

How Electronic Cash and Credit Card Encryption Work

1 First, the customer must download an electronic payment helper application to use when shopping on the Web with "electronic cash." Some browser software already includes this application. The helper application includes an encryption key that encodes personal financial information so that it can only be decoded by the electronic payment service using a special second key.



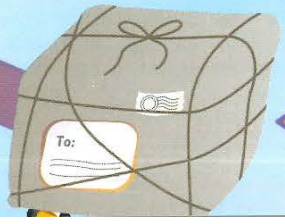
Download Helper App
(Not necessary if browser has built-in encryption)

Internet

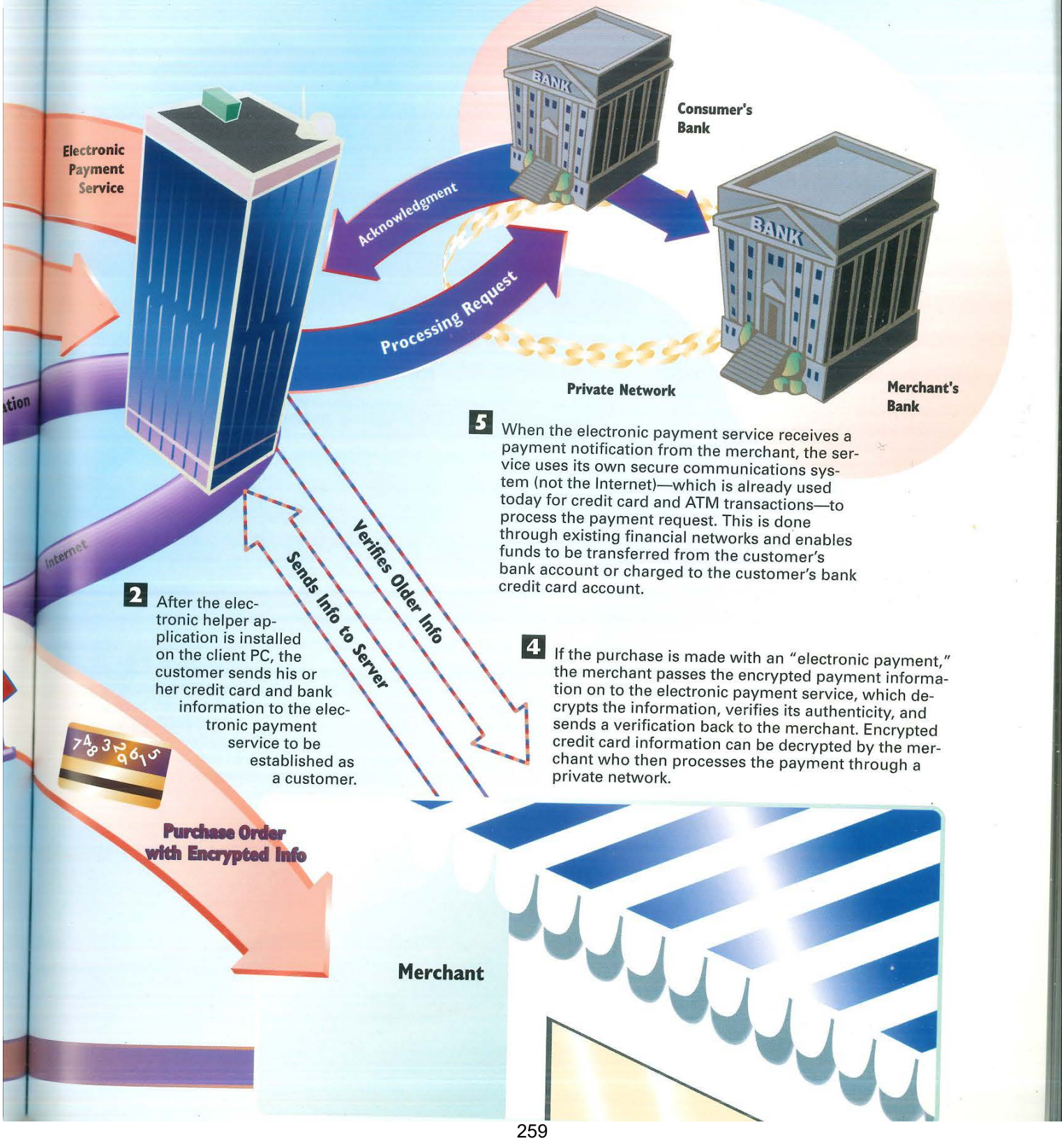
Send Set-up Information



3 When a customer makes a purchase, the encrypted payment information is sent along with the purchase order to the merchant. Information is encrypted by applying a unique key to scramble the information. This scrambled information can only be unscrambled by using a special second key.



6 The merchant receives verification in a matter of seconds and then can fulfill the order and send the goods to the customer.

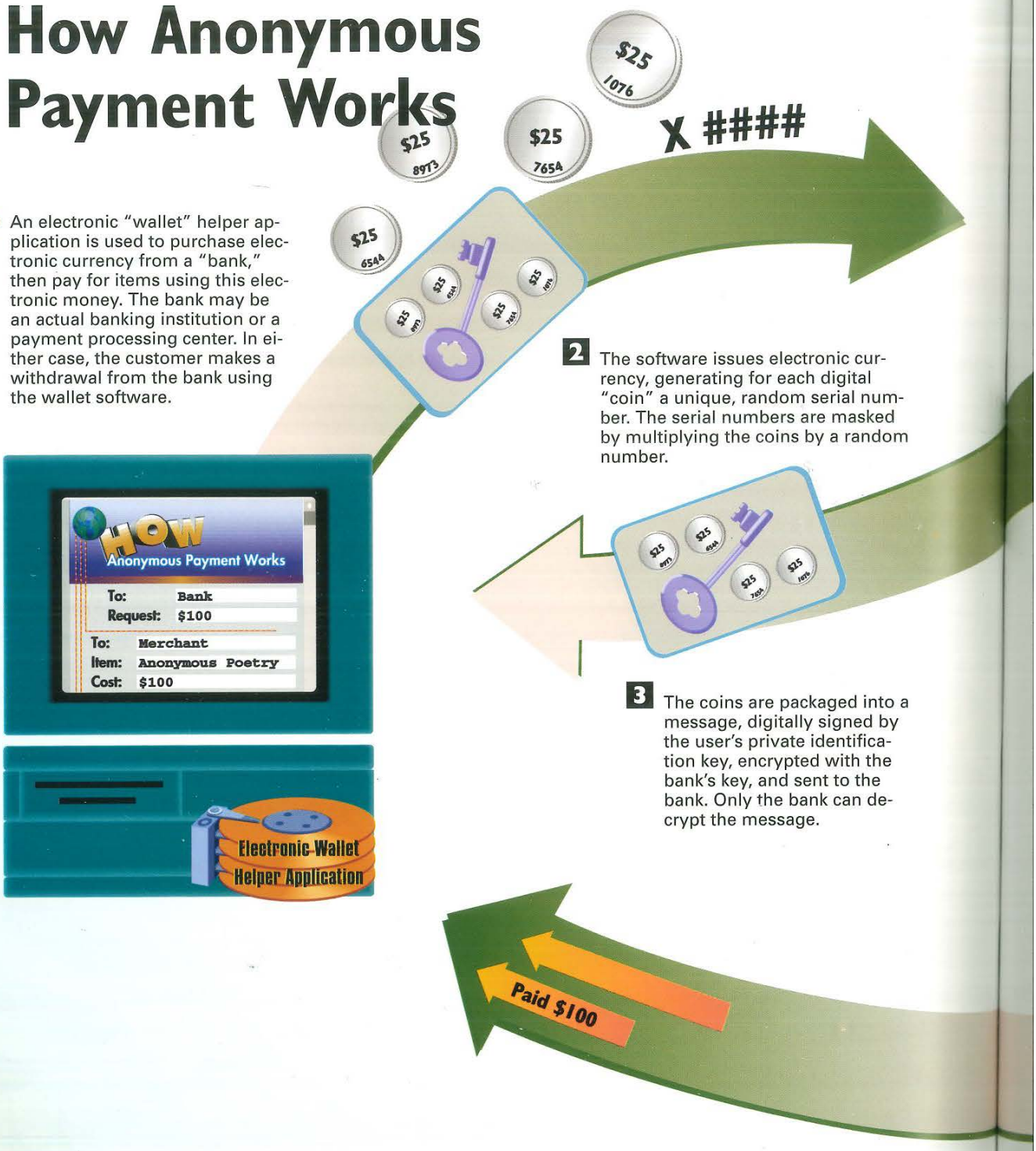


5 When the electronic payment service receives a payment notification from the merchant, the service uses its own secure communications system (not the Internet)—which is already used today for credit card and ATM transactions—to process the payment request. This is done through existing financial networks and enables funds to be transferred from the customer's bank account or charged to the customer's bank credit card account.

2 After the electronic helper application is installed on the client PC, the customer sends his or her credit card and bank information to the electronic payment service to be established as a customer.

4 If the purchase is made with an "electronic payment," the merchant passes the encrypted payment information on to the electronic payment service, which decrypts the information, verifies its authenticity, and sends a verification back to the merchant. Encrypted credit card information can be decrypted by the merchant who then processes the payment through a private network.

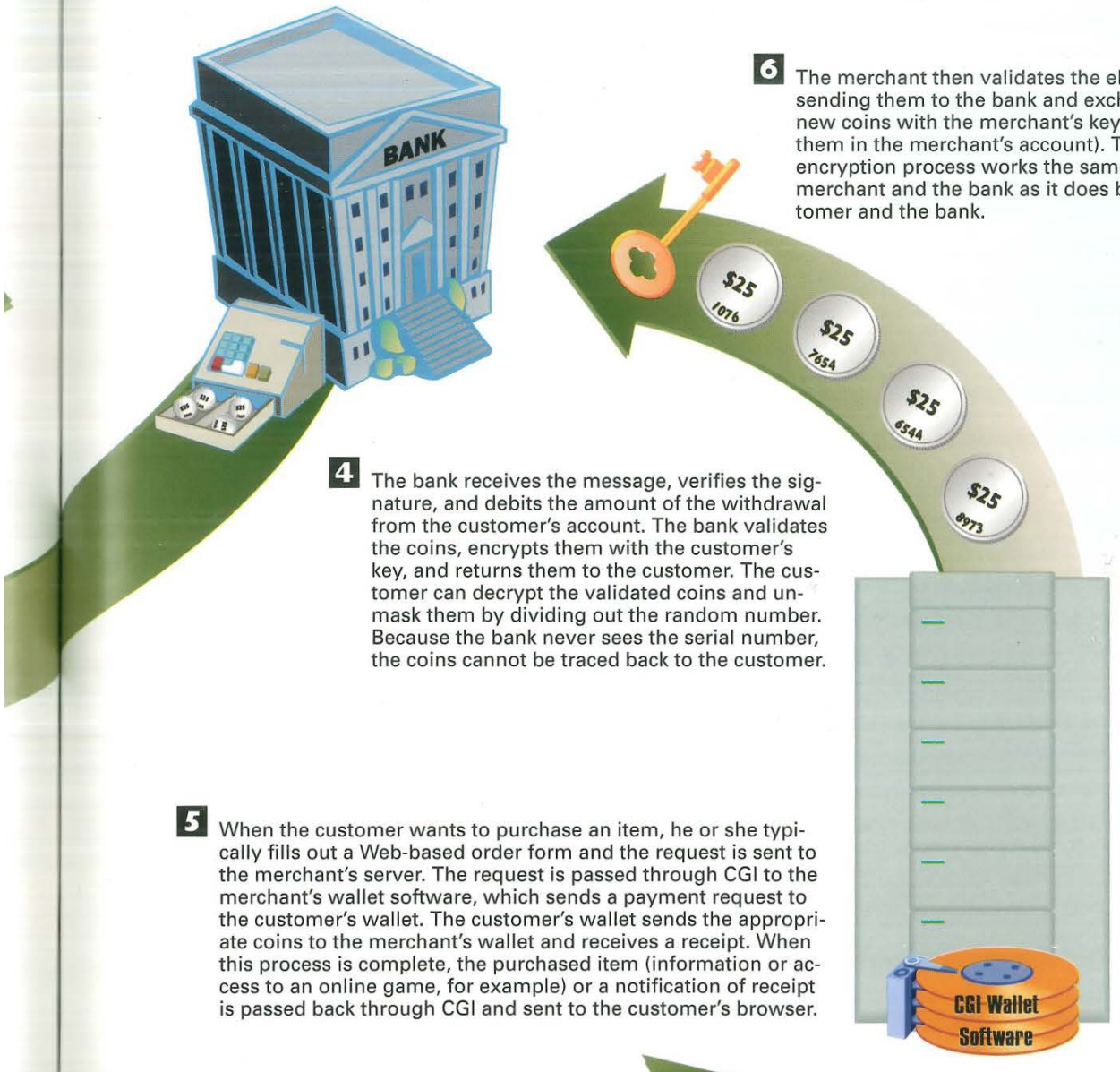
How Anonymous Payment Works



1 An electronic "wallet" helper application is used to purchase electronic currency from a "bank," then pay for items using this electronic money. The bank may be an actual banking institution or a payment processing center. In either case, the customer makes a withdrawal from the bank using the wallet software.

2 The software issues electronic currency, generating for each digital "coin" a unique, random serial number. The serial numbers are masked by multiplying the coins by a random number.

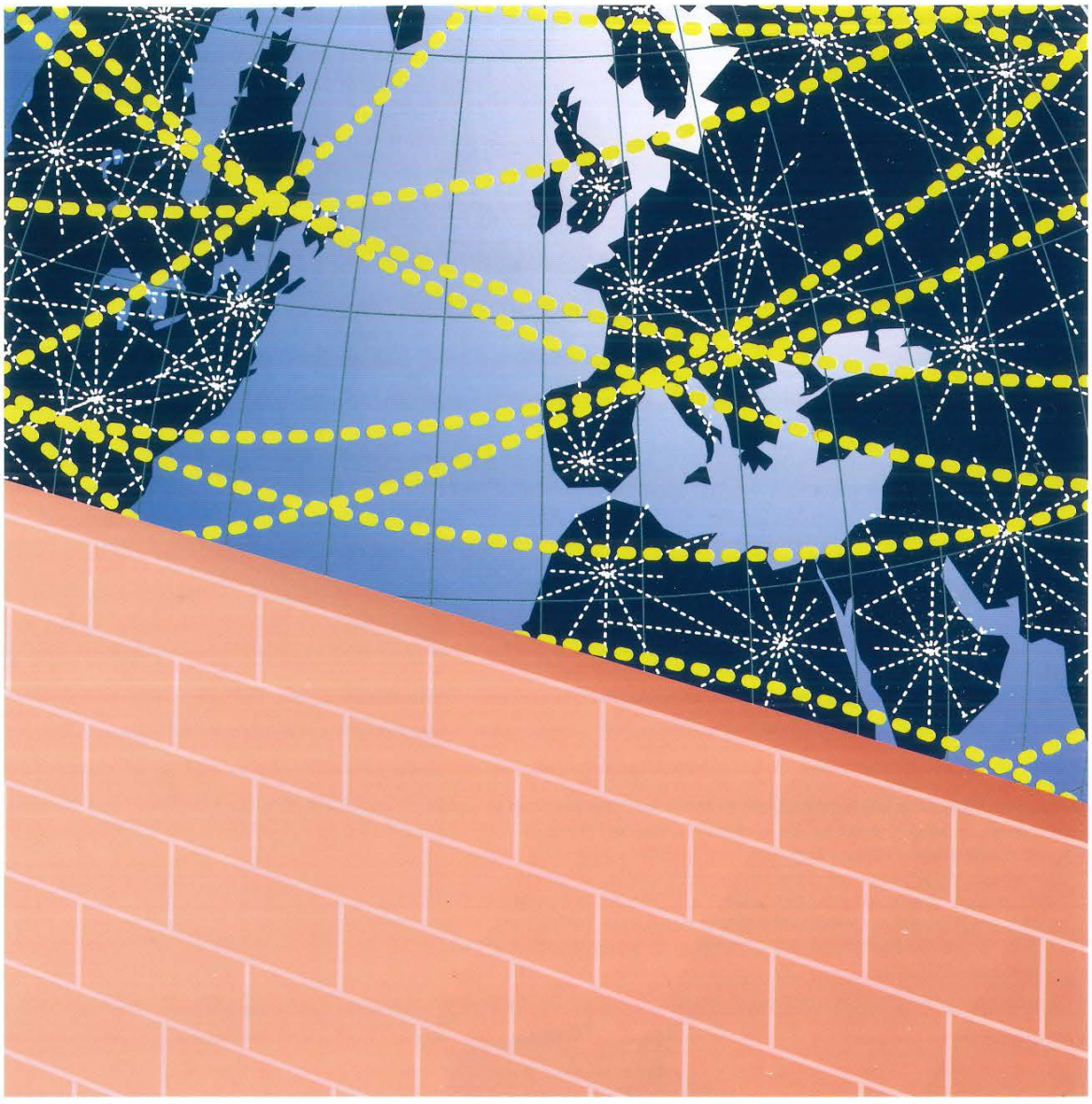
3 The coins are packaged into a message, digitally signed by the user's private identification key, encrypted with the bank's key, and sent to the bank. Only the bank can decrypt the message.



6 The merchant then validates the electronic coins by sending them to the bank and exchanging them for new coins with the merchant's key (or by depositing them in the merchant's account). The masking and encryption process works the same between the merchant and the bank as it does between the customer and the bank.

4 The bank receives the message, verifies the signature, and debits the amount of the withdrawal from the customer's account. The bank validates the coins, encrypts them with the customer's key, and returns them to the customer. The customer can decrypt the validated coins and unmask them by dividing out the random number. Because the bank never sees the serial number, the coins cannot be traced back to the customer.

5 When the customer wants to purchase an item, he or she typically fills out a Web-based order form and the request is sent to the merchant's server. The request is passed through CGI to the merchant's wallet software, which sends a payment request to the customer's wallet. The customer's wallet sends the appropriate coins to the merchant's wallet and receives a receipt. When this process is complete, the purchased item (information or access to an online game, for example) or a notification of receipt is passed back through CGI and sent to the customer's browser.



PART 10

SAFEGUARDING THE INTERNET

Chapter 45: How Firewalls Work
266

Chapter 46: How "Smurf Attacks" Can Cripple ISPs
272

Chapter 47: How Viruses Work
276

Chapter 48: How Cookies, Passports, and Web Tracking Work
280

Chapter 49: Cryptography and Privacy
288

Chapter 50: How Digital Certificates Work
292

Chapter 51: Parental Controls on the Internet
296

THE very nature of the Internet makes it vulnerable to attack. It was designed to allow for the freest possible exchange of information, data, and files—and it has succeeded admirably, far beyond its designers' wildest expectations. However, that freedom carries a price: Hackers and virus-writers try to attack the Internet and computers connected to the Internet; those who want to invade others' privacy attempt to crack databases of sensitive information or snoop on information as it travels across Internet routes; and distasteful and pornographic sites have sprung up on the Web and on Usenet newsgroups.

In this section of the book, we'll look at a variety of security-related issues. We'll see how various tools have been developed for making transactions on the Net more secure and helping companies protect their sensitive data. We'll examine the thorny issue of pornography versus free speech, and we'll see how software can block children from visiting obscene sites or getting obscene materials. We'll also look at some of the more controversial technologies on the Internet, such as cookies, which allow Web servers to track you as you move through their sites. And we'll look at how viruses work and how hackers attack Internet service providers (ISPs).



Chapter 45, "How Firewalls Work," looks at firewalls. Many companies whose networks are connected to the Internet have a great deal of sensitive information on their networks and want to make sure that their data and computers are safe from attack. The answer is to use *firewalls*—systems that allow people from inside a company to use the Internet, but also stop people on the Internet from getting at the company's computers.

Chapter 46, "How 'Smurf Attacks' Can Cripple ISPs," looks at attacks launched by hackers that can cripple Internet service providers. In a *Smurf attack*, also called *Smurfing*, a hacker targets an ISP and floods it with so much "garbage" traffic that none of the ISP's customers are able to use the service. Smurfing can become one of the most common kinds of hacking attacks on the Internet.

Chapter 47, "How Viruses Work," looks at viruses and how they are detected. Any program you download from the Internet has the potential for being infected with a virus, and it could, in turn, infect your computer. We'll see just how these nasty data-killers work, and we'll look at antivirus tools that can detect and kill them.

Chapter 48, "How Cookies, Passports, and Web Tracking Work," explores two controversial technologies—cookies and Web tracking—as well as a technology that may help preserve people's privacy—Internet "passports." Some people worry that cookies and Web tracking may invade people's privacy. Others say they don't and instead can help customize the Web to people's interests. *Cookies* are bits of data put on a hard disk when someone visits certain Web sites. That data

can be used for many purposes. One common use is to make it easier for people to use Web sites that require a username and password by storing that information and then automatically sending the information whenever it's requested. *Passports* enable people to decide what kind of information about them may be tracked by Web sites. *Web tracking* enables those who run Web sites to see how people use their site.

In Chapter 49, "Cryptography and Privacy," we'll examine *cryptosystems.* An enormous amount of information is sent across the Internet every day—everything from personal email to corporate data to credit card information and other highly sensitive material. All that information is vulnerable to hackers and snoopers. Because the information is sent in packets along public routers, the possibility exists that someone could intercept and decipher it. As a way to ensure that the sensitive material can't be looked at, sophisticated cryptosystems have been developed so that only the sender and receiver know what's in the packets.

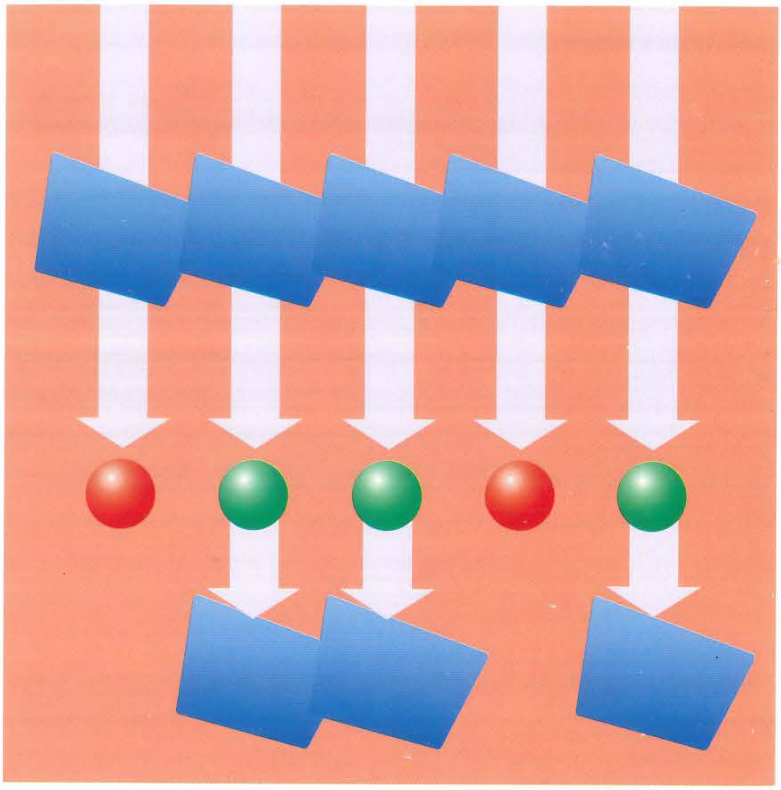
Chapter 50, "How Digital Certificates Work," describes digital certificates. On the Internet, no face-to-face communication takes place, so it can be difficult to know whether people really are who they say they are. *Digital certificates* are used to absolutely identify someone. If someone sends you an email, for example, a digital certificate will let you know that the person is who he says he is.



Finally, Chapter 51, "Parental Controls on the Internet," takes a detailed look at the issues of pornography and free speech on the Internet. Explicit sexual material is posted on the Internet, and some people would like to fine and jail people and organizations that allow such material to be posted. Passing those kinds of laws raises a whole host of constitutional issues about free speech. As a way to solve the problem, companies create and sell software for parents that enables them to block their children from seeing obscene and violent material on the Internet. In this chapter, we'll see how one of the most popular pieces of parental control software works.

CHAPTER 45

How Firewalls Work



EVERY time a corporation connects its internal computer network or LAN (local area network) to the Internet it faces potential danger. Due to the Internet's openness, every corporate network connected to it is vulnerable to attack. Crackers on the Internet could theoretically break into the corporate network and do harm in a number of ways: They could steal or damage important data; damage individual computers or the entire network; use the corporate computer's resources; or use the corporate network and resources as a way of posing as a corporate employee. The solution isn't to cut off the network from the Internet. Instead, the company can build firewalls to protect its network. These firewalls allow anyone on the corporate network to access the Internet, but they stop crackers, hackers, or others on the Internet from gaining access to the corporate network and causing damage.

Firewalls are hardware and software combinations that are built using routers, servers, and a variety of software. They sit at the most vulnerable point between a corporate network and the Internet, and they can be as simple or complex as system administrators want to build them. There are many types of firewalls, but most of them have a few common elements.

One of the simplest kinds of firewalls utilizes packet filtering. In packet filtering, a screening router examines the header of every packet of data traveling between the Internet and the corporate network. Packet headers have information in them, including the IP address of the sender and receiver, the protocol being used to send the packet, and other similar information. Based on that information, the router knows what kind of Internet service—such as FTP or rlogin—is being used to send the data, as well as the identity of the sender and receiver of the data. (The command, rlogin, is similar to Telnet, allowing someone to log into a computer. It can be dangerous because it allows users to bypass having to type in a password.) After this information is determined, the router can bar certain packets from being sent between the Internet and the corporate network. For example, the router could block any traffic except for e-mail. Additionally, it could block traffic to and from suspicious destinations or from certain users.

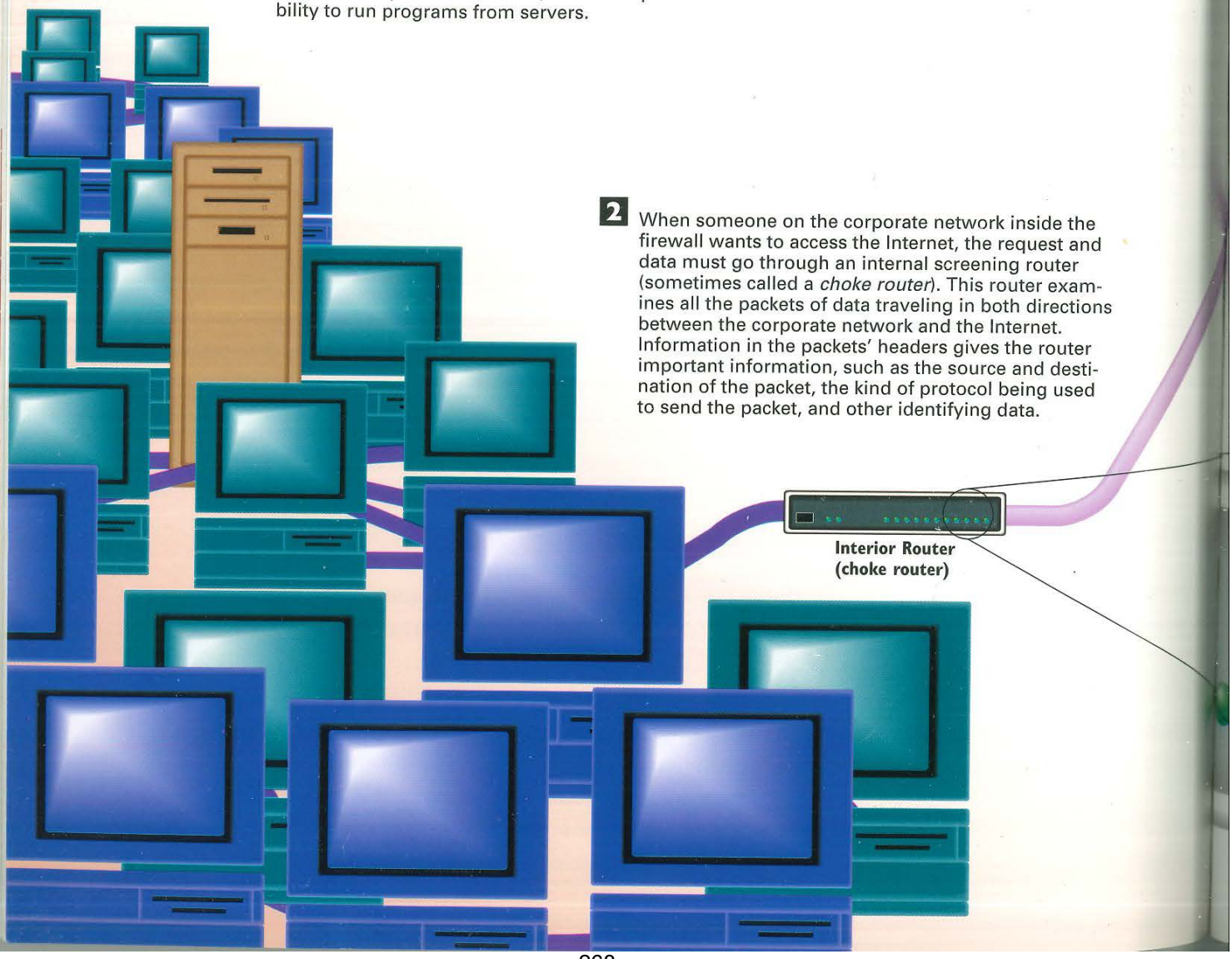
Proxy servers are also commonly used in firewalls. A *proxy server* is server software that runs on a host in a firewall, such as a bastion host. Because only the single proxy server (instead of the many individual computers on the network) interacts with the Internet, security can be maintained. That single server can be kept more secure than can hundreds of individual computers on a network.

When someone inside the corporate network wants to access a server on the Internet, a request from the computer is sent to the proxy server, the proxy server contacts the server on the Internet, and then the proxy server sends the information from the Internet server to the computer inside the corporate network. By acting as a go-between, proxy servers can maintain security as well as log all traffic between the Internet and the network.

A *bastion host* (explained in the illustration) is another common component of firewalls.

How Firewalls Work

1 The firewall shields the internal corporate network from the Internet. The internal network works as networks normally do, with servers providing internal services such as email, access to corporate databases, and the capability to run programs from servers.



2 When someone on the corporate network inside the firewall wants to access the Internet, the request and data must go through an internal screening router (sometimes called a *choke router*). This router examines all the packets of data traveling in both directions between the corporate network and the Internet. Information in the packets' headers gives the router important information, such as the source and destination of the packet, the kind of protocol being used to send the packet, and other identifying data.

Interior Router (choke router)

Internet

6 An *exterior screening router* (also called *an access router*) screens packets between the Internet and the perimeter network. It adds an extra level of protection by screening packets based on the same rules as the internal screening router. This protect the network even if the internal router fails. However, it may also add more rules for screening packets specifically designed to protect the bastion host.

Firewall

5 The bastion host is placed in a perimeter network in the firewall, so it is not on the corporate network itself. This further shields the corporate network from the Internet. If the bastion host were on the normal corporate network, an intruder could conceivably gain access to every computer on the network and to all network services. Isolating the bastion server from the corporate network by putting it in a perimeter network prevents an intruder from gaining access to the internal corporate network, even if there is a server break-in.

Exterior Router (access router)

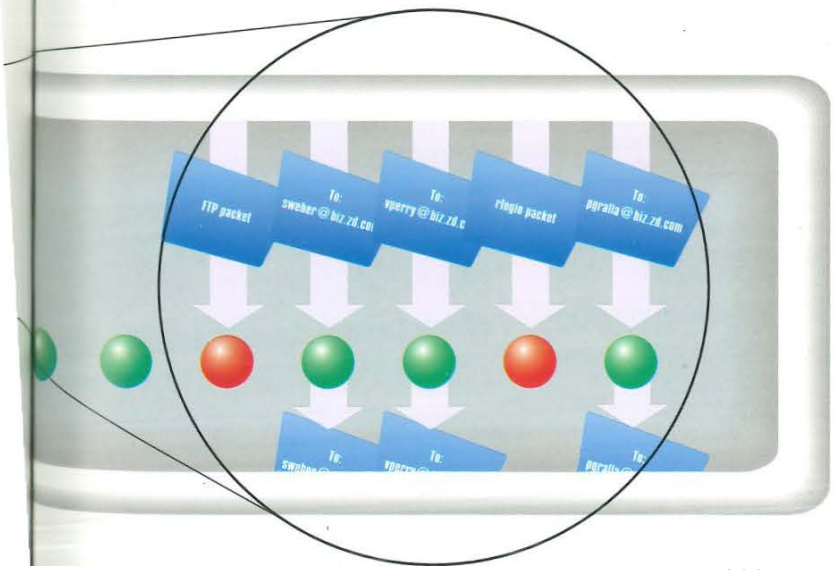
Bastion Host



4 A *bastion host* in the firewall is the primary point of contact for connections coming in from the Internet for services such as receiving email and allowing access to the corporation's FTP site. The bastion host is a heavily protected server with many security provisions built in and it is the only contact point for incoming Internet requests. In this way, none of the computers or hosts on the corporate network can be contacted directly for requests from the Internet, providing a level of security. Bastion hosts can also be set up as *proxy servers*—servers that process any requests from the internal corporate network to the Internet, such as browsing the Web or downloading files via FTP. See the next illustration for an explanation of how proxy servers work.

Screened Subnet Firewall

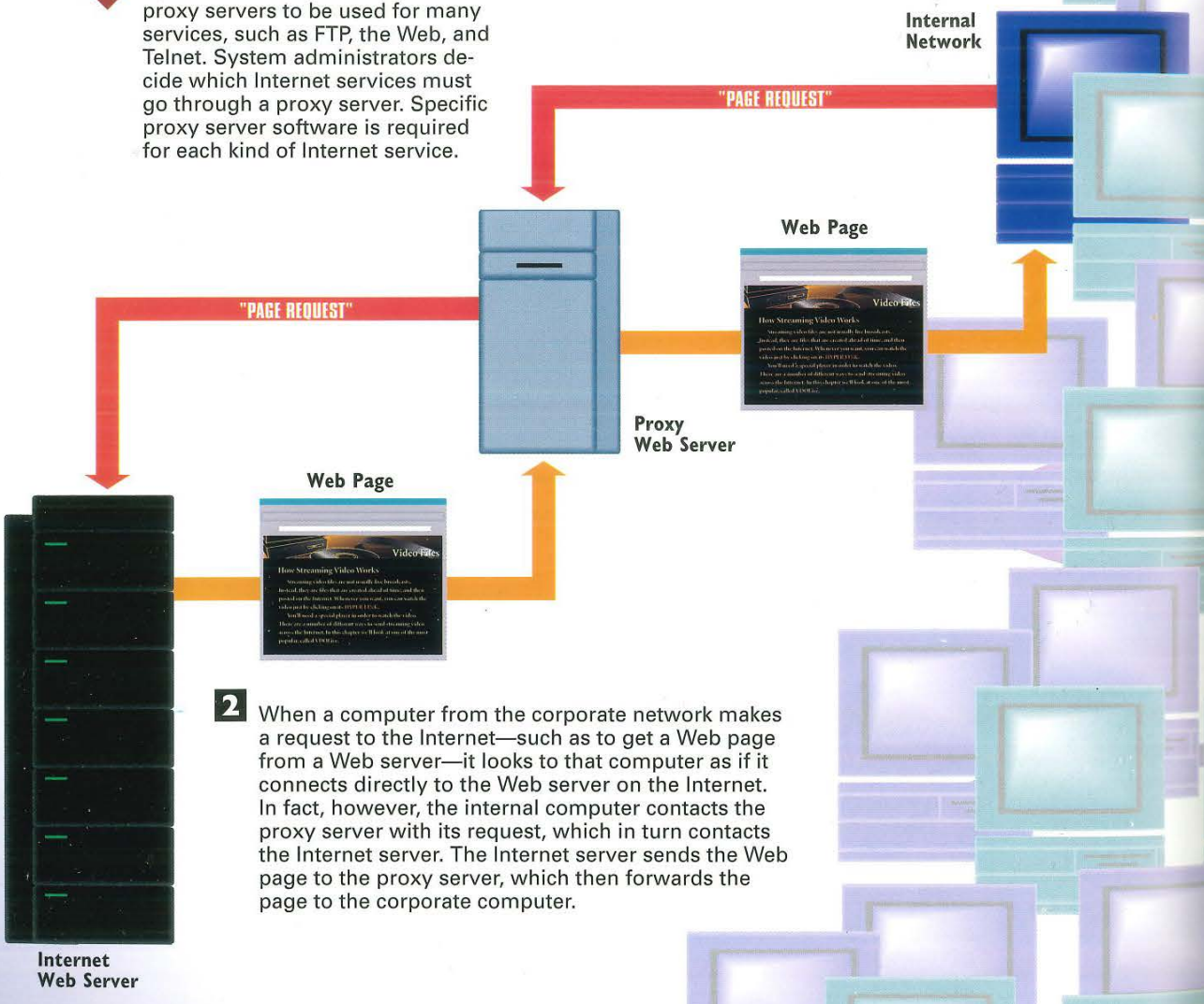
Internal Network



3 Based on the information in the headers, the screening router will allow certain packets to be sent or received, but will block other packets. For example, it might not allow some services such as rlogin to be run. The router also might not allow packets to be sent to and from specific Internet locations because those locations have been found to be suspicious. Conceivably, a router could be set up to block every packet traveling between the Internet and the internal network except for email. System administrators set the rules for determining which packets to allow in and which ones to block.

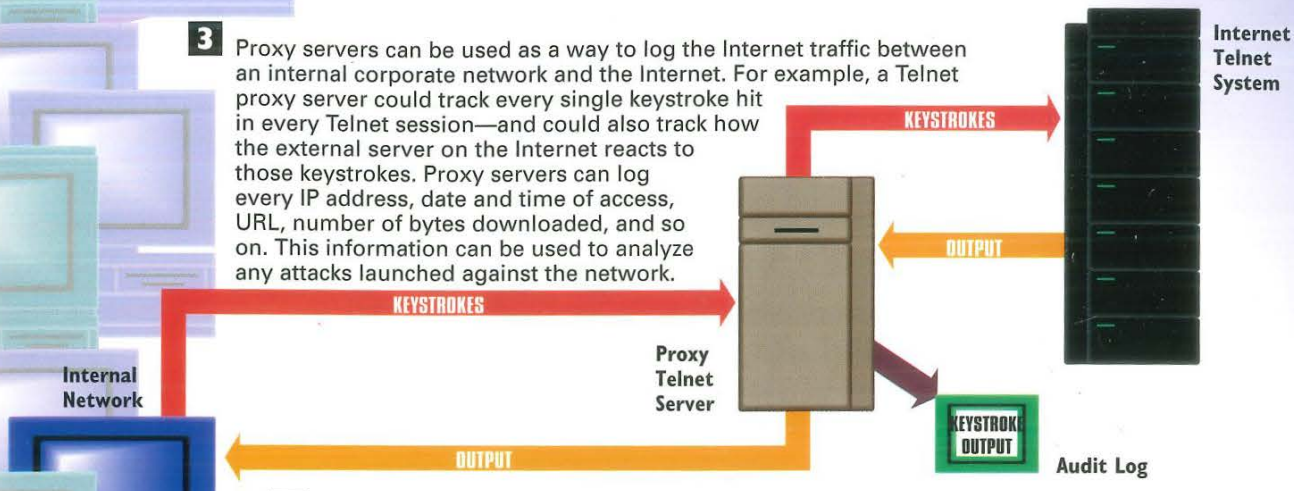
How Proxy Servers Work

1 System administrators can set up proxy servers to be used for many services, such as FTP, the Web, and Telnet. System administrators decide which Internet services must go through a proxy server. Specific proxy server software is required for each kind of Internet service.

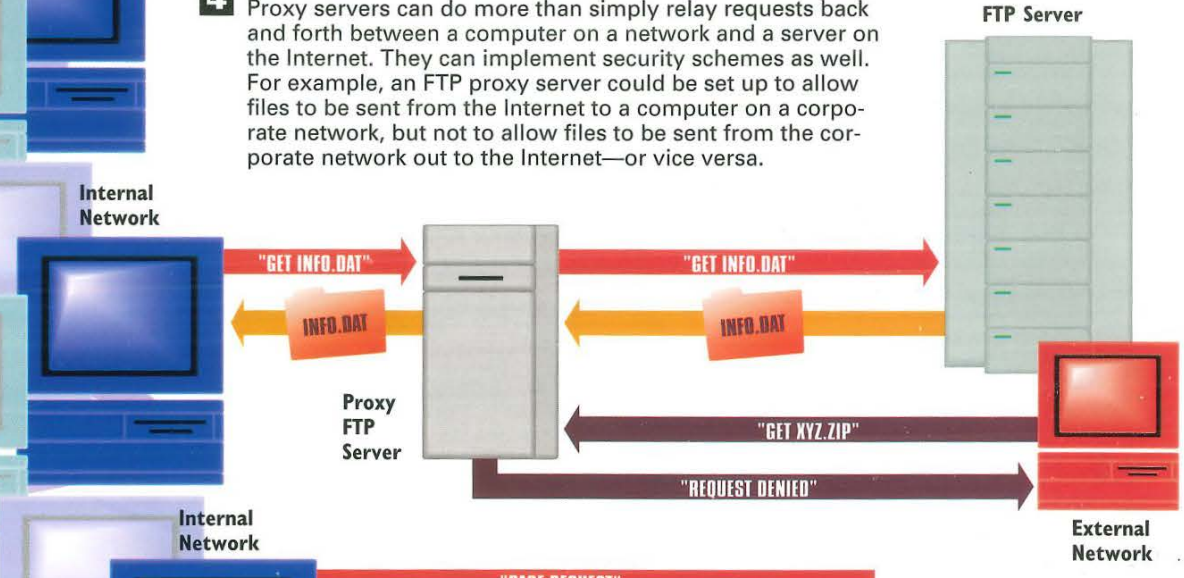


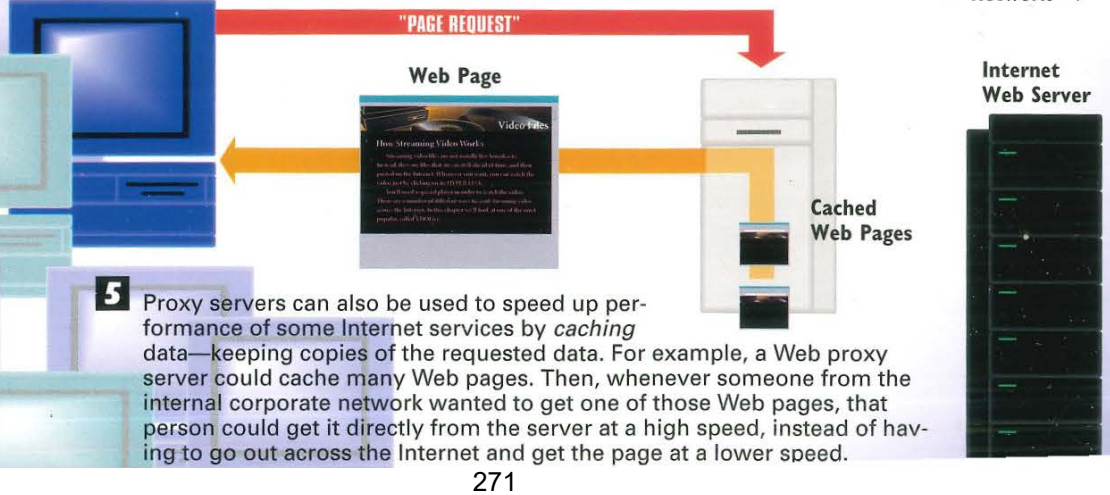
2 When a computer from the corporate network makes a request to the Internet—such as to get a Web page from a Web server—it looks to that computer as if it connects directly to the Web server on the Internet. In fact, however, the internal computer contacts the proxy server with its request, which in turn contacts the Internet server. The Internet server sends the Web page to the proxy server, which then forwards the page to the corporate computer.

3 Proxy servers can be used as a way to log the Internet traffic between an internal corporate network and the Internet. For example, a Telnet proxy server could track every single keystroke hit in every Telnet session—and could also track how the external server on the Internet reacts to those keystrokes. Proxy servers can log every IP address, date and time of access, URL, number of bytes downloaded, and so on. This information can be used to analyze any attacks launched against the network.



4 Proxy servers can do more than simply relay requests back and forth between a computer on a network and a server on the Internet. They can implement security schemes as well. For example, an FTP proxy server could be set up to allow files to be sent from the Internet to a computer on a corporate network, but not to allow files to be sent from the corporate network out to the Internet—or vice versa.

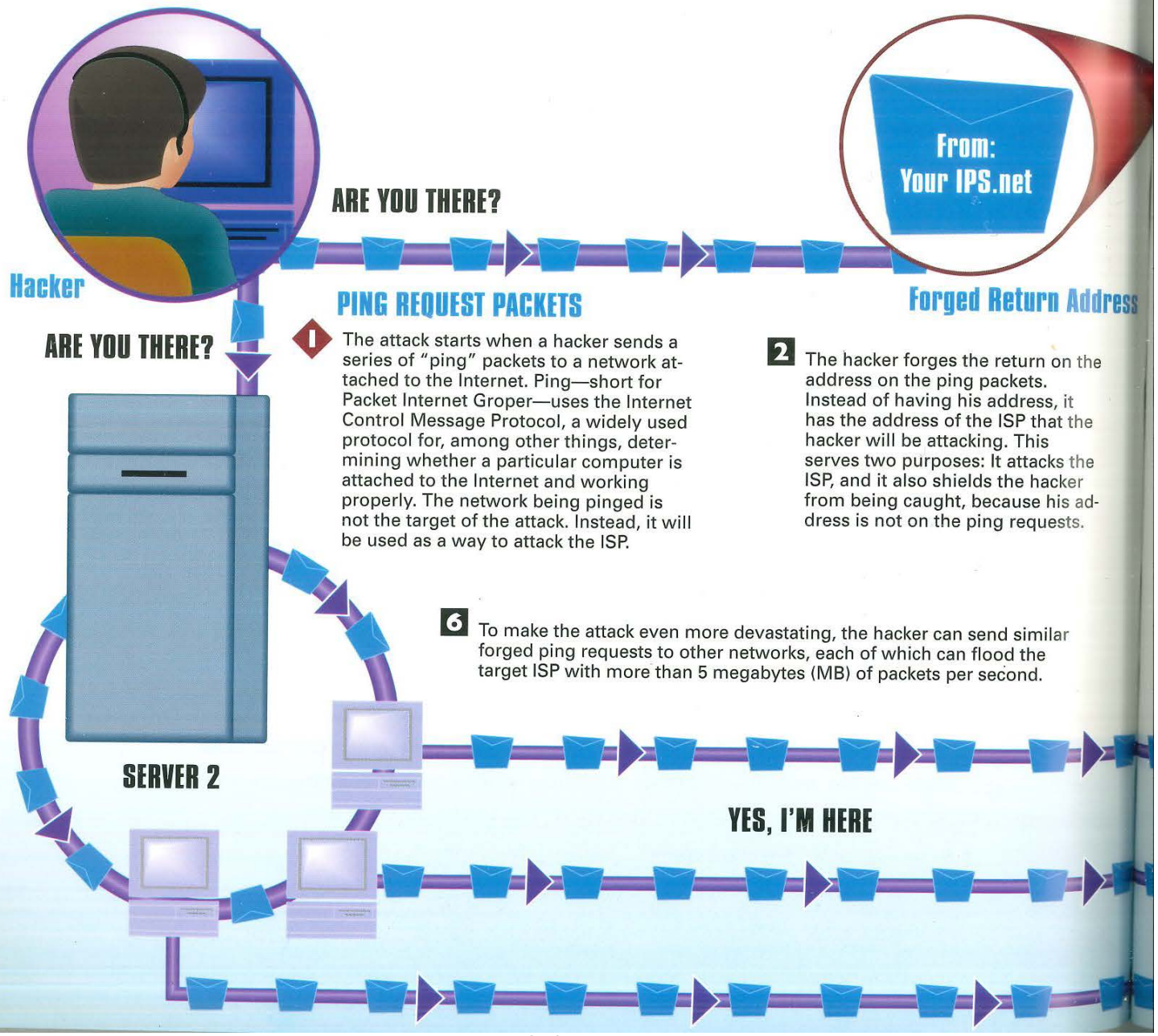


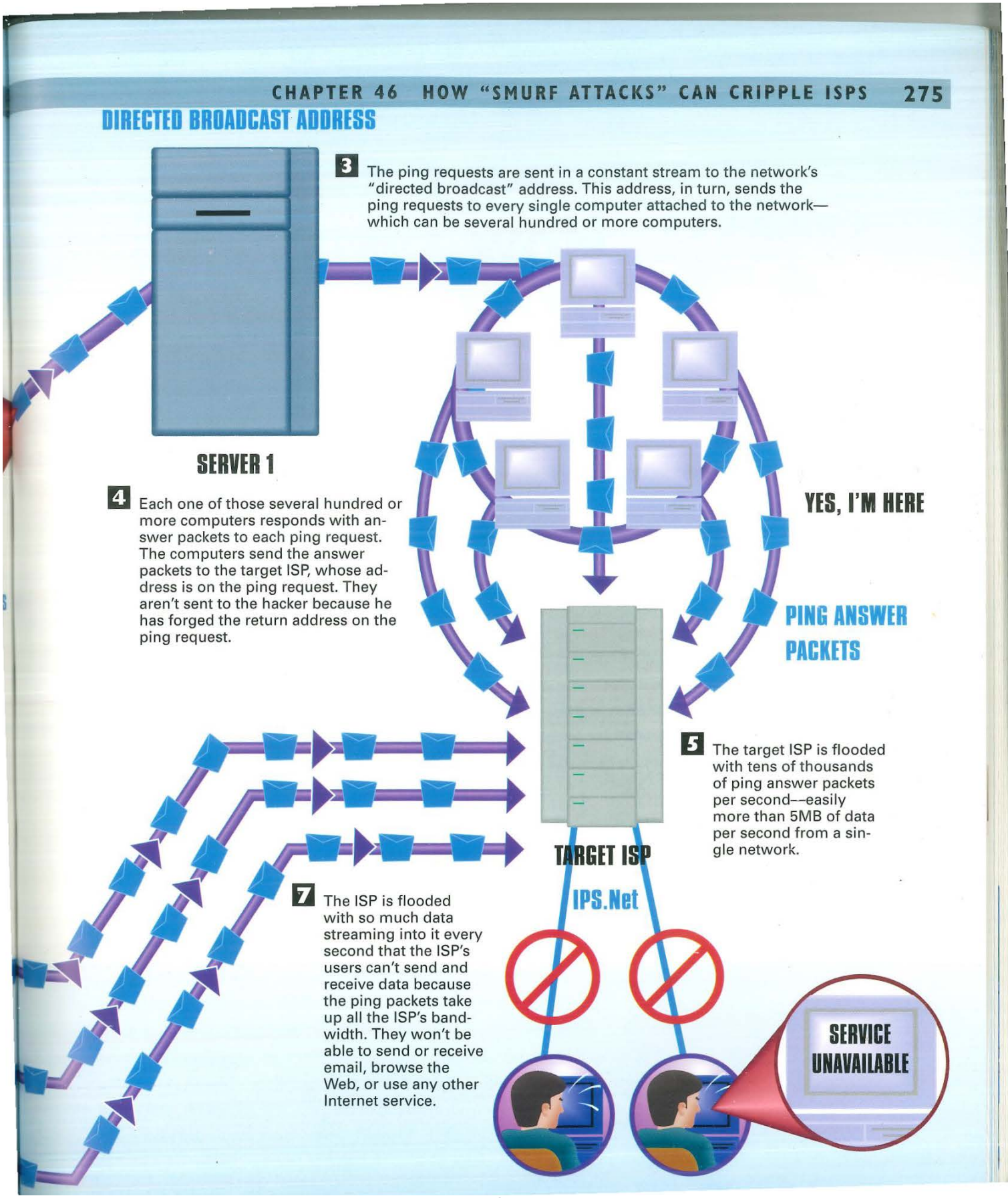


5 Proxy servers can also be used to speed up performance of some Internet services by *caching* data—keeping copies of the requested data. For example, a Web proxy server could cache many Web pages. Then, whenever someone from the internal corporate network wanted to get one of those Web pages, that person could get it directly from the server at a high speed, instead of having to go out across the Internet and get the page at a lower speed.

How Hackers Attack Internet Service Providers with "Smurf Attacks"

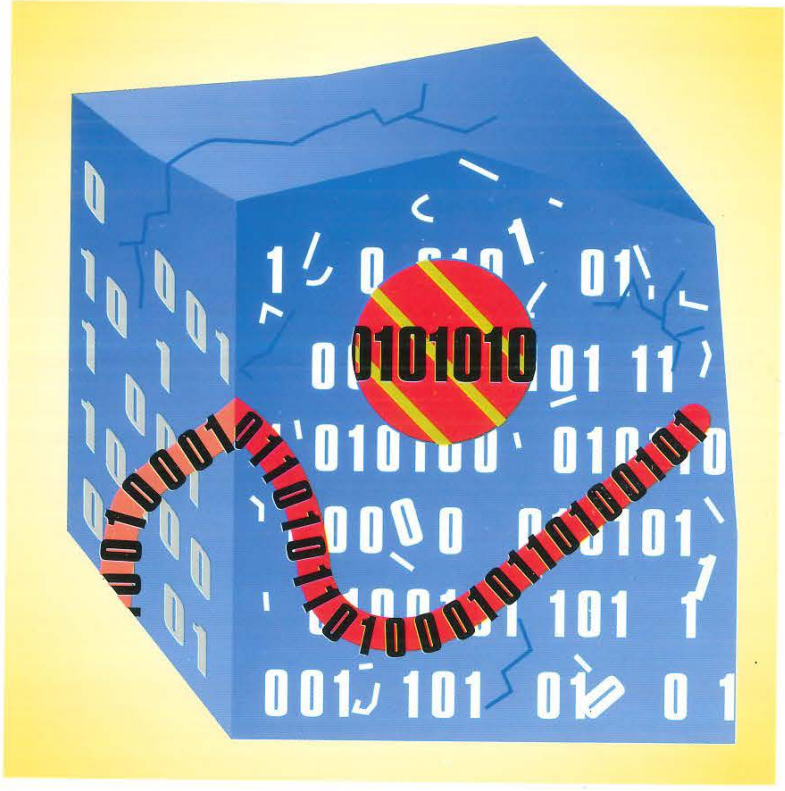
In a "smurf attack," also called "smurfing," a hacker targets an Internet Service Provider (ISP) and floods it with so much "garbage" traffic that none of the ISP's customers are able to use the service. It has become one of the most popular kinds of hacker attacks on the Internet.




DIRECTED BROADCAST ADDRESS
3 The ping requests are sent in a constant stream to the network's "directed broadcast" address. This address, in turn, sends the ping requests to every single computer attached to the network—which can be several hundred or more computers.
SERVER 1
4 Each one of those several hundred or more computers responds with answer packets to each ping request. The computers send the answer packets to the target ISP, whose address is on the ping request. They aren't sent to the hacker because he has forged the return address on the ping request.
YES, I'M HERE
PING ANSWER PACKETS
5 The target ISP is flooded with tens of thousands of ping answer packets per second—easily more than 5MB of data per second from a single network.
TARGET ISP
IPS.Net
7 The ISP is flooded with so much data streaming into it every second that the ISP's users can't send and receive data because the ping packets take up all the ISP's bandwidth. They won't be able to send or receive email, browse the Web, or use any other Internet service.
SERVICE UNAVAILABLE

CHAPTER 47

How Viruses Work



THE Internet, just like the rest of the world, is not a perfectly safe place to visit. If you download files from the Internet, there is a chance—a very small chance, but nonetheless a chance—that your computer could become infected with a virus.

Viruses are malicious programs that invade your computer. They can cause many different kinds of damage, such as deleting data files, erasing programs, or destroying everything they find on your hard disk. Not every virus causes damage; some simply flash annoying messages.

Although you can get a virus from the Internet by downloading files to your computer, the Internet is not the only place that viruses can be picked up. If you've sent files via email or on your company's internal network, you can get them that way as well. There have even been instances when commercially bought, shrink-wrapped software has contained viruses.

The term "virus" is a somewhat generic term applied to a wide variety of programs. Viruses are written for specific kinds of computers, such as PCs or Macintoshes, because the files they infect run only on one kind of computer.

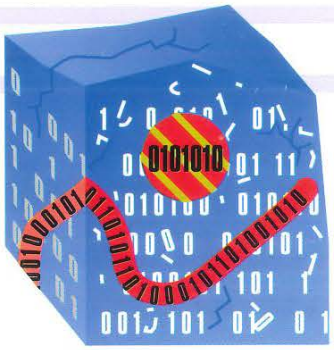
Traditional viruses attach themselves to programs or data files, infect your computer, replicate themselves on your hard disk, and then damage your data, hard disk, or files. Viruses usually attack four parts of your computer: its executable program files; its file-directory system that tracks the location of all your computer's files (and without which, your computer won't work); its boot and system areas that are needed to start your computer; and its data files. There was a time when it was believed that data files could not be infected by viruses, but recently, viruses have been written that infect data files, too. For example, some viruses attach themselves to Word for Windows macros inside a Word for Windows data file and are launched whenever a particular macro is run.

Trojan horses are programs that disguise themselves as normal, helpful programs, but in fact are viruses. For example, if a program purported to be a financial calculator, but really deleted every file on your hard disk ending, that program would be called a Trojan horse.

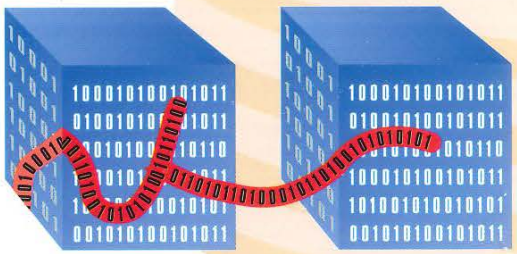
Worms are programs designed to infect networks such as the Internet. They travel from networked computer to networked computer, replicating themselves along the way. The most infamous worm of all was released on November 2, 1988, and copied itself to many Internet host computers, eventually bringing the Internet to its knees.

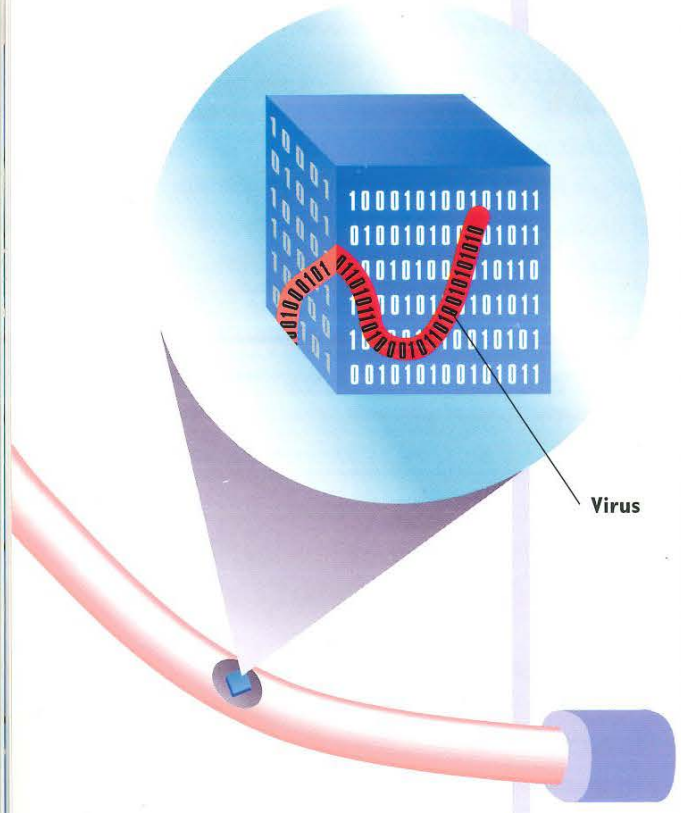
The best way to protect your computer against viruses is to use antiviral software. There are several kinds. A *scanner* checks to see if your computer has any files that have been infected, whereas an *eradication program* will wipe the virus from your hard disk. Sometimes eradication programs can kill the virus without having to delete the infected program or data file, whereas other times those infected files must be deleted. Still other programs, sometimes called *inoculators*, do not allow a program to be run if it contains a virus, and stop your computer from being infected.

How Viruses Infect Computers

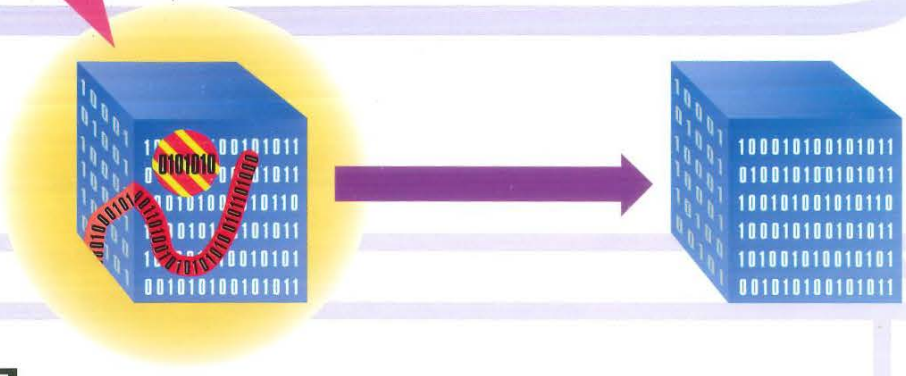


3 Viruses can corrupt program or data files so that they work oddly, not at all, or cause damage when they do run. They can destroy all the files on your computer, change the system files that your computer needs when it is turned on, and cause other types of damage.





1 A virus hides inside a legitimate program where it remains dormant until you run the infected program. The virus springs into action when you actually run the infected program. Sometimes the first thing the virus will do is infect other programs on your hard disk by copying itself into them.

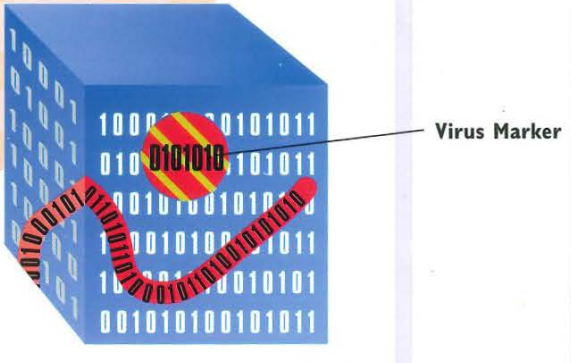


5 *Eradication programs* disinfect, or remove, viruses from software. Sometimes they can eradicate the virus without damaging the program that the virus has infected. In other instances, they have to destroy the program as well as the virus.

4 Software programs called *scanners* check for viruses and alert you to the viruses' presence. They work in many different ways. One method of detection is to check your program files for tell-tale virus markers that indicate the presence of a virus. Other methods include checking to see whether a program's file size has changed. Some types of antiviral programs run continuously on your computer and check any program for the presence of a virus before the program is run.



2 Some viruses place messages called *v-markers* or *virus markers* inside programs that they infect, and these help manage the viruses' activities. Each virus has a specific v-marker associated with it. If a virus encounters one of these markers in another program, it knows that the program is already infected so it doesn't replicate itself there. When a virus cannot find more unmarked files on a computer, that can signal to the virus that there are no more files to be infected. At this point, the virus may begin to damage the computer and its data.



CHAPTER

48 How Cookies, Passports, and Web Tracking Work



PRIVACY issues are a big concern on the Net. Much information can be gathered about people when they use the Net, and it's not always clear who will use that information or how it will be used. In particular, two technologies that concern people are cookies and Web tracking. Both serve useful purposes, but many people worry that there is a "Big Brother" aspect to them. One technology, Internet "passports," may enable people to ensure that their privacy isn't invaded while still enabling Web sites to gather information that can be used to deliver specialized services to Web surfers.

Cookies are bits of data put on a hard disk when someone visits certain Web sites. The most common use of this data is to make it easier for people to use Web sites that require a username and password. The cookie on the hard disk has the username and password in it, so people don't have to log in to every page that requires that information. Instead, the cookie sends the information to the server and the person can visit the page freely.

Cookies can contain virtually any kind of information, such as the last time a person visited the site, the person's favorite sites, and similar, customizable information. They can be used to track people as they go through a Web site and to help gain statistics about what kinds of pages people like to visit. Although some people view them as invaders of privacy, they can also make the Web a much better place to visit by doing things such as making it easier to conduct electronic commerce.

Web sites can't read cookies that other sites have placed on a hard disk, so if you visit a certain news site, for example, no other site can get the information out of the cookie. Users always have the option of not allowing cookies to be placed on their hard disks—browsers give people this option.

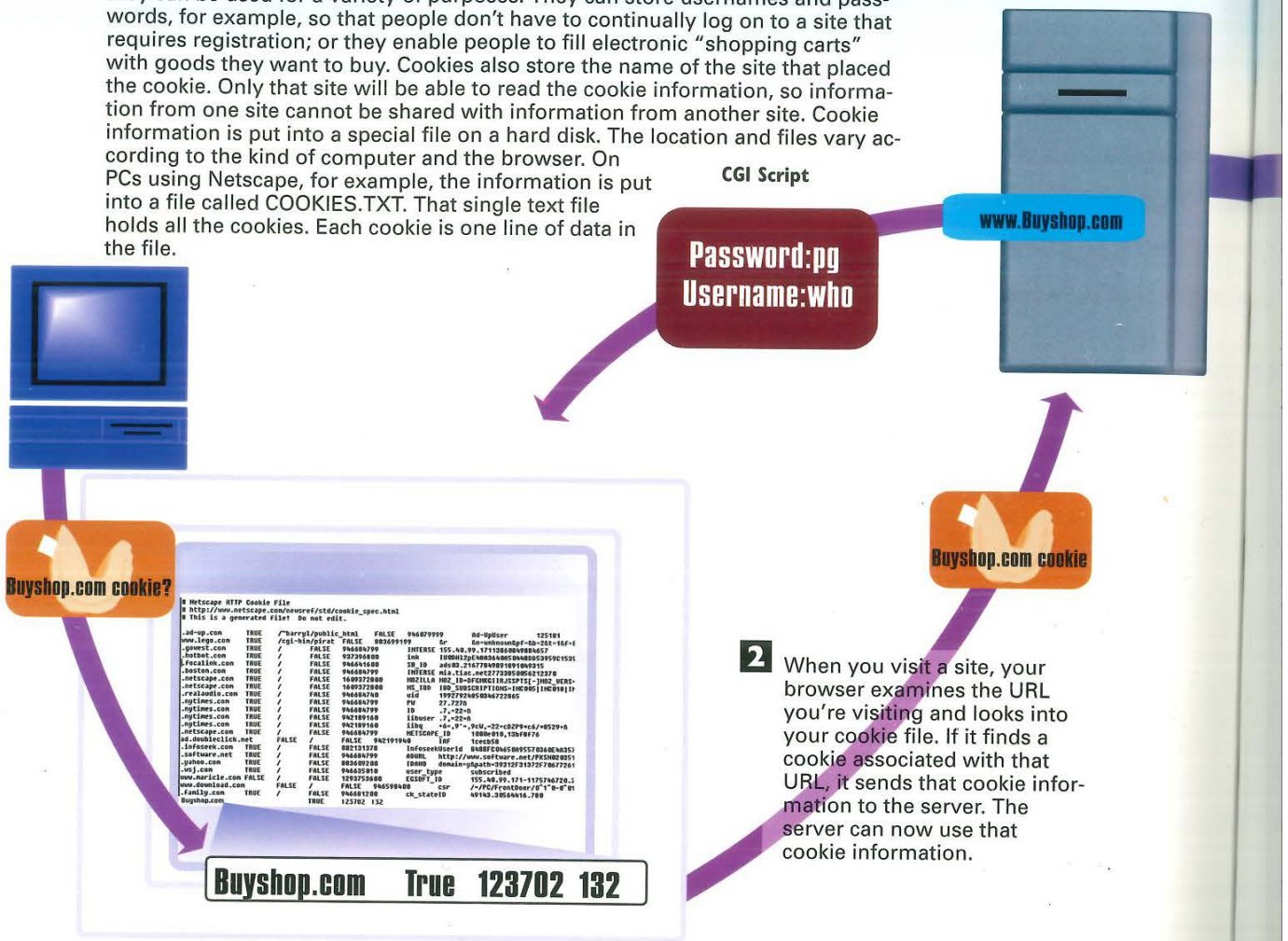
Although cookies can be used to track how people use a Web site, many other methods can be used, as well. In one method, Web server logs are examined in detail. This would make it possible, for example, to identify the most popular pages on the site, the sites people have just visited, how many pages people read in a typical visit, and similar information. Other methods include using software "sniffers" that examine every packet coming into or going out of a Web site. Webmasters can use this tracking information to help create better sites—but they can also use it to assemble demographic information to sell to advertisers.

In fact, Webmasters need to know a good deal of information about how their site is being used, such as the total number of visitors per day, the total number of pages viewed, how people travel through the site, where people come to the site from, and where they go when they leave. The third illustration in this chapter shows the functionality of Web tracking software from a company called Accrue.

To allay people's privacy concerns, a variety of technologies and standards are being developed. They include the Platform for Privacy Preferences (P3P), the Internet Content and Exchange standard (ICE), and the Open Profiling Standard (OPS). Generically, we'll call them Internet passports. These Internet passports let people control which information about themselves they'll allow to be released to Web sites—and how that information can be used. And they let people control what kind of information can be gathered about their surfing and how that information can be used, as well. In general, the more information that people allow to be gathered about them, the more specialized services they'll gain on the Web, such as customized news feeds. We'll look at Internet passports in the second illustration in this chapter.

How Cookies Work

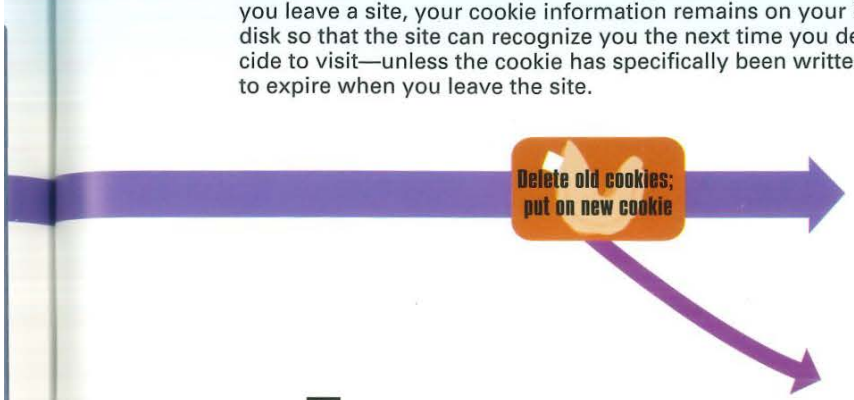
1 Cookies are pieces of data placed on a computer's hard drive by a Web server; they can be used for a variety of purposes. They can store usernames and passwords, for example, so that people don't have to continually log on to a site that requires registration; or they enable people to fill electronic "shopping carts" with goods they want to buy. Cookies also store the name of the site that placed the cookie. Only that site will be able to read the cookie information, so information from one site cannot be shared with information from another site. Cookie information is put into a special file on a hard disk. The location and files vary according to the kind of computer and the browser. On PCs using Netscape, for example, the information is put into a file called COOKIES.TXT. That single text file holds all the cookies. Each cookie is one line of data in the file.



2 When you visit a site, your browser examines the URL you're visiting and looks into your cookie file. If it finds a cookie associated with that URL, it sends that cookie information to the server. The server can now use that cookie information.

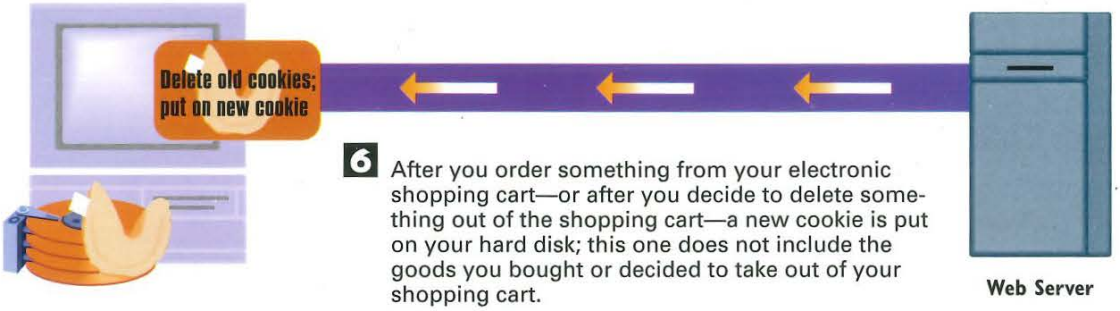
3 If no cookie is associated with the URL, the server will place a cookie inside the cookie file. Some sites may first ask a series of questions, such as name and password, and will then place a cookie on the hard disk with that information in it. This is typical of sites that require registration. Commonly, a CGI script on the server takes the information that the user has entered and then writes the cookie onto the hard disk.

4 As you travel through a Web site, more information may need to be put into your cookie. On a site where you can purchase goods online, for example, you might put goods into an electronic shopping cart. Every time you did this, new cookie information would be added, detailing the goods you wanted to buy. When new cookie information is put in, a CGI script deletes the old cookie information and puts in a new cookie. When you leave a site, your cookie information remains on your hard disk so that the site can recognize you the next time you decide to visit—unless the cookie has specifically been written to expire when you leave the site.

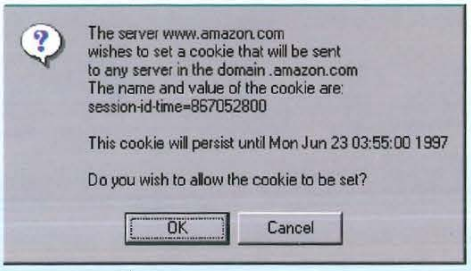




5 The server takes actions based on your cookies—for example, displaying your electronic shopping cart. If the site enables you to buy online, it might have asked for your credit card number. For security reasons, that number will not be stored in your cookie. Instead, it will be stored on a secure server. When you decide to buy something, you enter a secure area with your browser. Your cookie then sends an ID to the server that identifies you, and the server then displays your credit card information, enabling you to buy online.



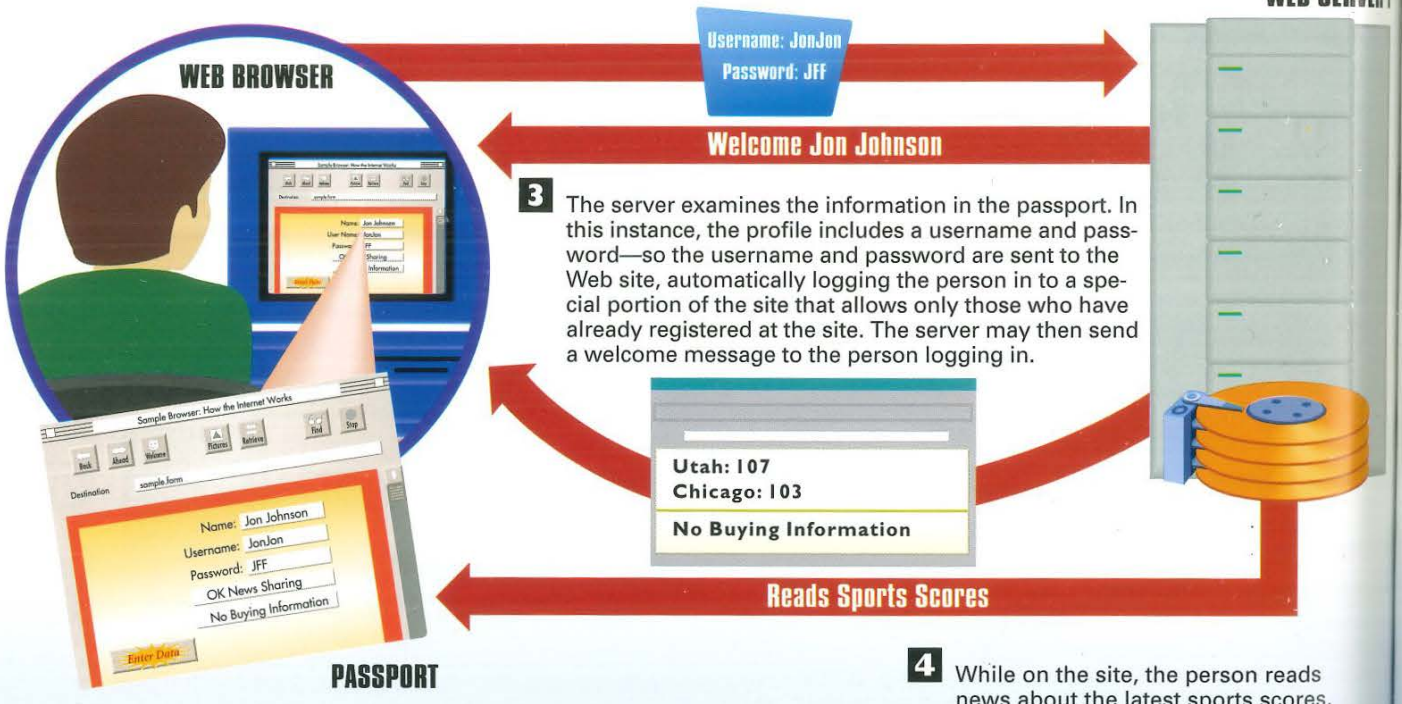
6 After you order something from your electronic shopping cart—or after you decide to delete something out of the shopping cart—a new cookie is put on your hard disk; this one does not include the goods you bought or decided to take out of your shopping cart.



7 Because some people don't like cookies to be placed on their hard disk, browsers give people control over whether to accept cookies, to not accept cookies, or to ask each time a cookie is being placed on the hard disk. Pictured is the message you get if you've asked to be told each time a cookie is placed on your hard disk.

How Internet "Passports" Work

Internet "passports" are designed to let people decide what personal information they will allow to be released to Web sites. The more information that people will allow to be released, the more sophisticated the services that can be delivered to them. The user also determines what information about their surfing habits can be gathered by Web sites and what information that's been gathered can be shared among Web sites. A variety of technologies are involved with Internet passports, including the Platform for Privacy Preferences (P3P), the Internet Content and Exchange standard (ICE), and the Open Profiling Standard (OPS).



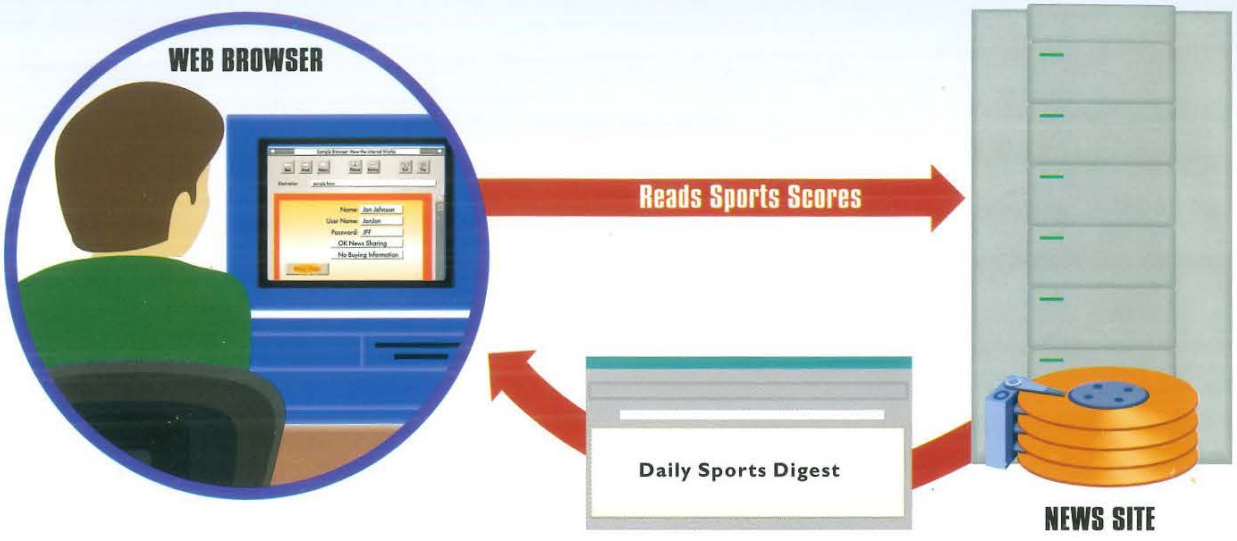
2 When the user visits a Web site, the information in the profile the person has put into his passport is sent to the site's Web server.

3 The server examines the information in the passport. In this instance, the profile includes a username and password—so the username and password are sent to the Web site, automatically logging the person in to a special portion of the site that allows only those who have already registered at the site. The server may then send a welcome message to the person logging in.

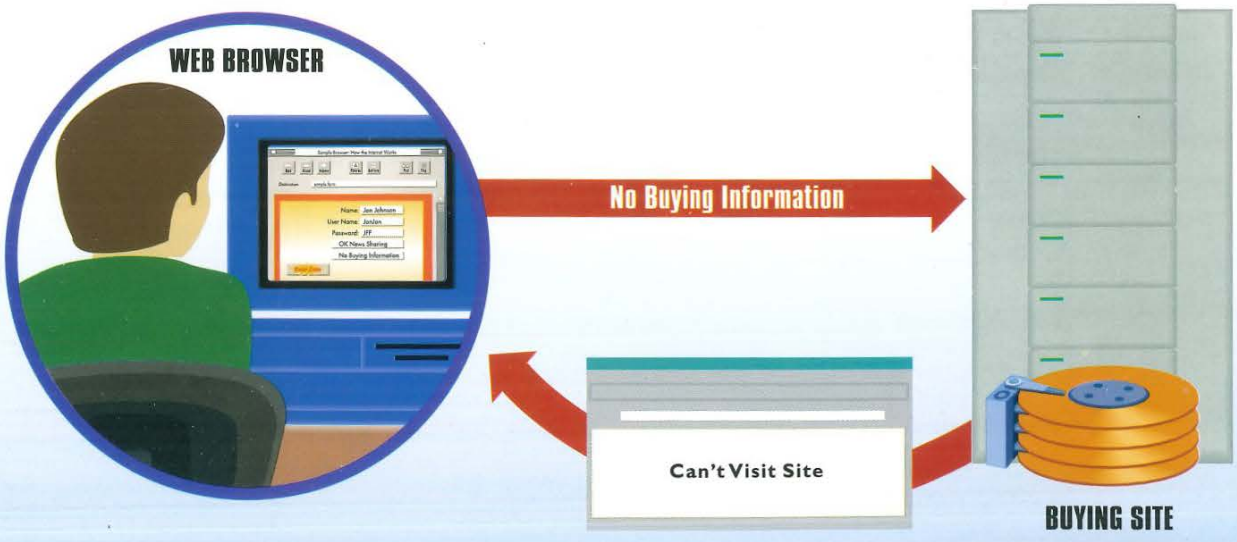
1 The passport lives inside a Web browser. A user fills out a profile in the browser, determining what information can be made available to Web sites, such as name, address, occupation, username and password, and age. The user also decides which kind of information about his surfing habits can be shared among Web sites—and which can't. In this instance, the person has decided that information about what news stories he reads can be shared, but not information about what products he buys.

4 While on the site, the person reads news about the latest sports scores, then buys an electric razor and a book about sailing. The Web site puts into the person's profile that he's read sports scores, but not that he's bought a razor and a book—because the person's profile said it would allow information about what news stories he reads to be shared, but not information about what he buys.

5 The person visits another Web site. The information in the profile the person has put into his passport is sent to the site's Web server. The server sees that the person has recently read a story about sports scores, and so it sends to him a daily digest of the latest sports news. Because the profile doesn't have any information about what the person has bought, it doesn't send any information about special sales on the site.

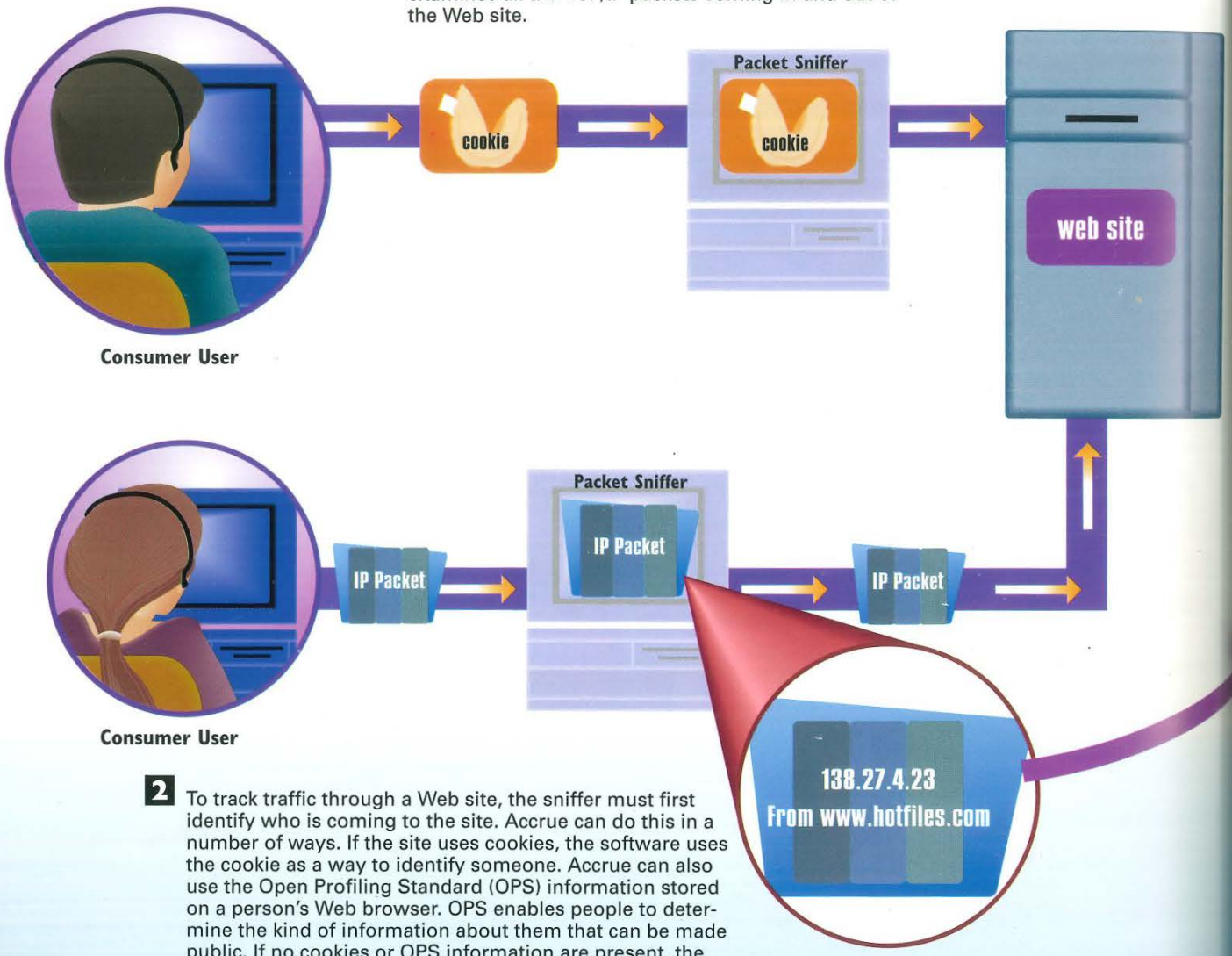


6 The person now surfs to a different Web site. This site only allows people in who have agreed, in their profile, to allow their online buying habits to be shared among sites. Because the person has said he doesn't want that information to be shared, the person is not allowed onto the site.



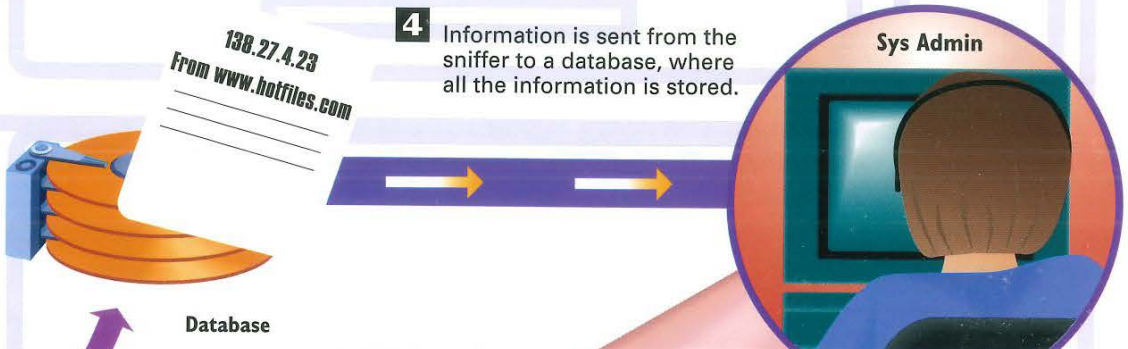
How Web Tracking Works

1 Accrue's software doesn't rely on analyzing Web logs to figure out how a Web site is being used. Instead, a "sniffer" sits on the Internet and analyzes traffic to the site. This sniffer is a computer that runs software that examines all the TCP/IP packets coming in and out of the Web site.



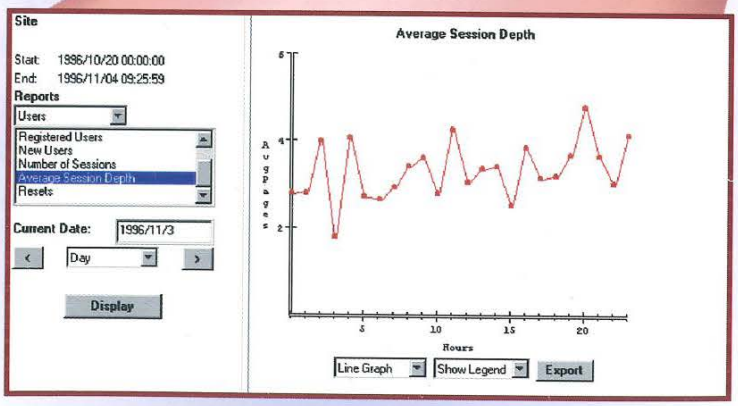
2 To track traffic through a Web site, the sniffer must first identify who is coming to the site. Accrue can do this in a number of ways. If the site uses cookies, the software uses the cookie as a way to identify someone. Accrue can also use the Open Profiling Standard (OPS) information stored on a person's Web browser. OPS enables people to determine the kind of information about them that can be made public. If no cookies or OPS information are present, the software uses the person's IP address.

3 The sniffer examines packets as they come into and go out of the site. It notes any time an action is taken, such as when someone requests a Web page, and whenever that action is completed, such as when the final packets from the page are delivered. It tracks who is making the requests, where they are coming from, where they are going, and similar information. This information is contained in the TCP/IP packets. The sniffer discards all the intermediate packets transmitted during each action—only the beginning and ending packets are needed. It discards all the intermediate packets because they provide no useful information.



4 Information is sent from the sniffer to a database, where all the information is stored.

5 Many kinds of reports can be created out of the database, such as the average amount of time people spend on a site, the average number of pages they read per visit, the most popular pages on a site, sites people have just visited, sites they're going to visit, and other information.



CHAPTER

49 Cryptography and Privacy



EVERY packet of data sent over the Internet traverses many public networks, which means access to those packets is not private. This is of particular concern when highly confidential information, such as corporate data and credit card numbers, is transmitted across the Internet. Unless there is some way to protect that kind of information, the Internet will never be a secure place to do business or send private, personal correspondence.

Fortunately, software engineers have developed ways to send confidential information securely. The information needs to be *encrypted*—that is, altered so that to anyone other than the intended recipient it will look like meaningless garble. The information also needs to be *decrypted*—that is, turned back into the original message by the recipient, and only by the recipient. Many complex *cryptosystems* have been created to allow for this kind of encryption and decryption.

The heart of understanding how cryptosystems work is to understand the concept of keys. *Keys* are secret values that computers use in concert with complex mathematical formulas called *algorithms* to encrypt and decrypt messages. The idea behind keys is that if someone encrypts a message with a key, only someone else with a matching key will be able to decrypt the message.

There are two kinds of common encryption systems: *secret-key cryptography*, also called *symmetric cryptography*, and *public-key cryptography*, also called *asymmetric cryptography*. The most common secret-key cryptography system is the Data Encryption Standard (DES). The best known public-key system is known as RSA.

In secret-key cryptography, only one key is used to encrypt and decrypt messages; both the sender and the receiver need copies of the same secret key. By contrast, in public-key cryptography, two keys are involved: a public key and a private key. Every person has both a public key and a private key. The public key is made freely available, whereas the private key is kept secret on the person's computer. The public key can encrypt messages—but only the private key can decrypt messages that the public key has encrypted. If someone wanted to send a message to you, for example, he would encrypt it with your public key. But only you, with your private key, would be able to decrypt the message and read it. Your public key could not decrypt it.

It's not feasible to use private-key cryptosystems widely on the Internet for things such as electronic commerce. For a company to conduct business over the Internet with a private-key system, it would mean creating millions of different private keys—one for each person who wanted to do business—and then figuring out some way to send those private keys securely over the Internet. With a public-key system, the business only needs to create a single public/private key combination. The business would post the public key for anyone to use for encrypting information—but only the business, with the private key, would be able to decrypt the data.

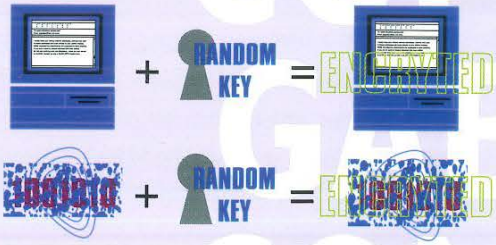
How Cryptosystems Work



1 Gabriel wants to send a confidential message over the Internet to Mia. Mia will need some way to decrypt the message as well as a way to guarantee that Gabriel, and not an imposter, has actually sent the message. First, Gabriel runs his message through an algorithm called a *hash function*. This produces a number known as the *message digest*. The message digest acts as a sort of "digital fingerprint" that Mia will use to ensure that no one has altered the message.

2 Gabriel now uses his private key to encrypt the message digest. This produces a unique digital signature that only he, with his private key, could have created.





3 Gabriel generates a new random key. He uses this key to encrypt his original message and his digital signature. Mia will need a copy of this random key to decrypt Gabriel's message. This random key is the only key in the world that can decrypt the message—and at this point only Gabriel has the key.

4 Gabriel encrypts this new random key with Mia's public key. This encrypted random key is referred to as the *digital envelope*. Only Mia will be able to decrypt the random key because it was encrypted with her public key, so only her private key can decrypt it.





5 Gabriel sends a message over the Internet to Mia that is composed of several parts: the encrypted confidential message, the encrypted digital signature, and the encrypted digital envelope.



6 Mia gets the message. She decrypts the digital envelope with her private key and out of it gets the random key that Gabriel used to encrypt the message.

7 Mia uses the random key to decrypt Gabriel's message. She can now read the confidential message that he sent to her. However, she can't yet be sure that the message hasn't been altered en route to her or that Gabriel was definitely the sender.



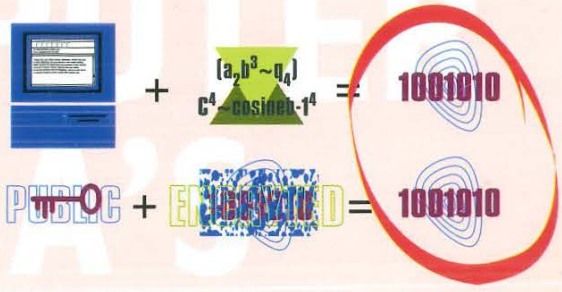
8 Mia now uses the random key and Gabriel's public key to decrypt his encrypted digital signature. When she does this, she gets his message digest, the message's "digital fingerprint."



9 Mia will use this message digest to see whether Gabriel indeed sent the message and that it was not altered in any way. She takes the message that she decrypted and runs it through the same algorithm—the hash function—that Gabriel ran the message through. This will produce a new message digest.



10 Mia compares the message digest that she calculated to the one that she got out of Gabriel's digital signature. If the two match precisely, she can be sure that Gabriel signed the message and that it was not altered after he composed it. If they don't match, then she knows that either he didn't compose the message, or that someone altered the message after he wrote it.





CHAPTER

50 How Digital Certificates Work



TO a great extent, the Internet is built on trust. It's a virtual world online; you don't actually see the people or institutions with whom you're communicating and getting and sending information. You don't actually see the person who's sending you email, for example, but you trust that they are who they say they are.

When it comes to financial transactions and other important communications, though, trust is not enough. There are hackers and crackers out there, con artists and scamsters who would be only too happy to take your credit card number, or who would like to learn your personal, financial, or business secrets. In the same way, businesses need to know that the person sending a credit card number really is who he says he is, and not an imposter who has managed to steal a credit card number from someone.

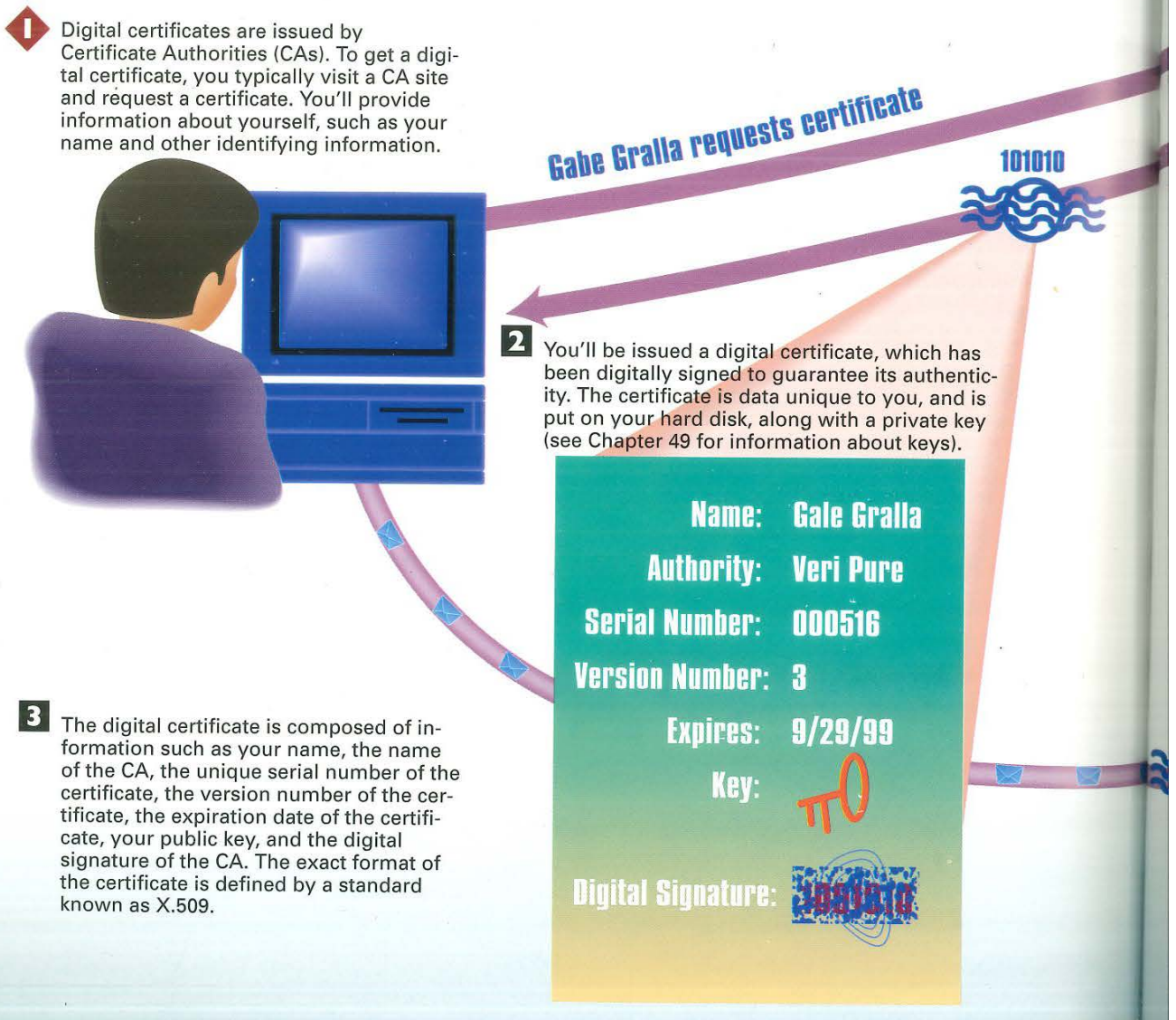
One of the major ways around the problem of trust is the use of "Digital Certificates." Digital certificates are used to verify that the person sending information, a credit card number, a message, or anything else over the Internet really is who he says he is. The certificates place information on a person's hard disk, and use encryption technology to create a unique digital certificate for each person. When someone with a digital certificate goes to a site, or sends email, that certificate is presented to the site or attached to the email, and it verifies that the user is who he claims to be.

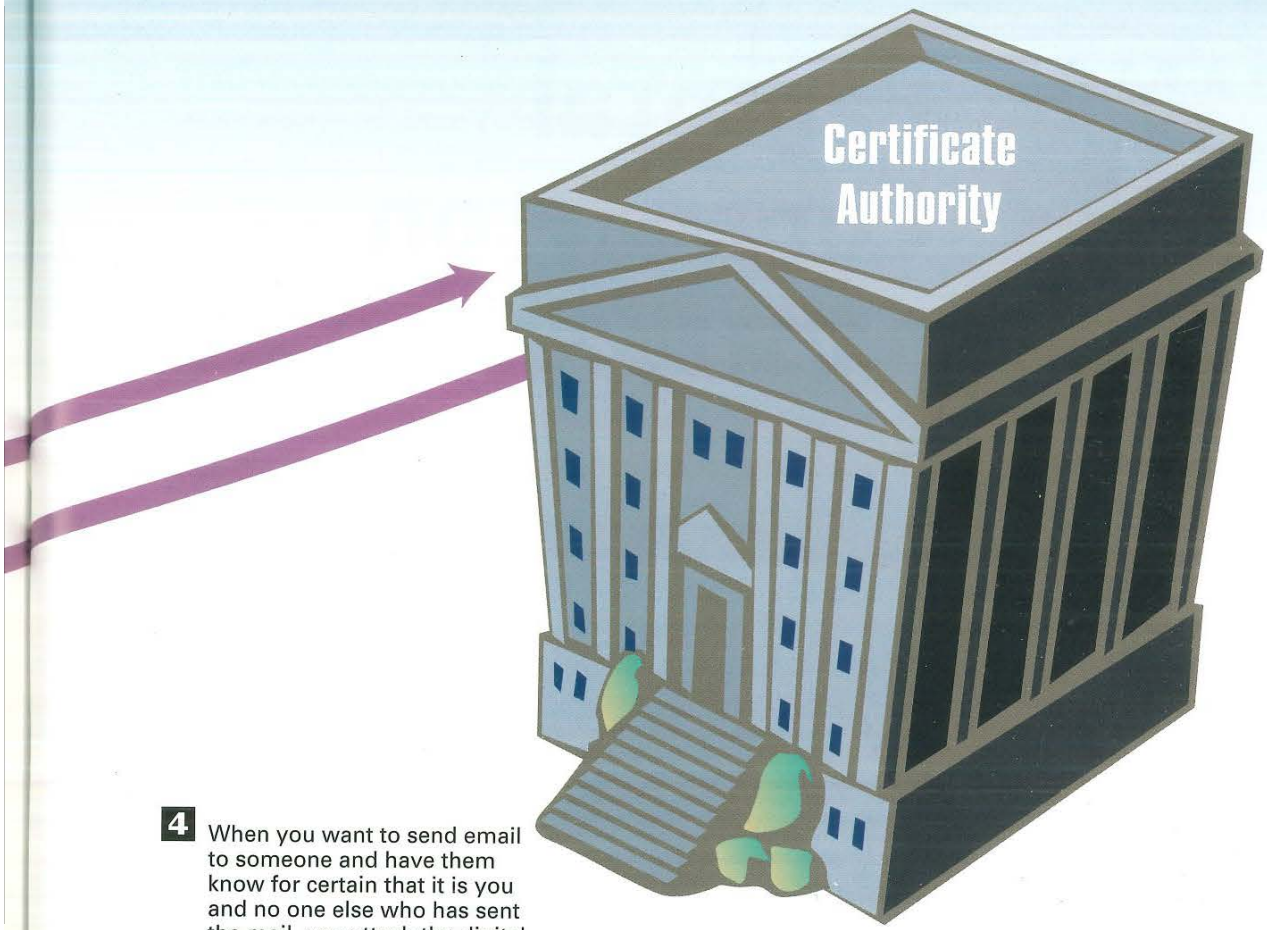
Because they use powerful encryption technology, digital certificates are quite secure. In fact, they are more secure than real-life signatures. In real life, a signature can be forged. On the Internet, a digital certificate can't be.

Digital certificates are issued by Certificate Authorities. These Certificate Authorities are private companies who charge either users or companies for the issuance of the certificates. You may be familiar with one such Certificate Authority, called VeriSign. Digital certificates contain information such as your name, the name of the Certificate Authority, the certificate's serial number, and similar information. The information has been encrypted in a way that makes it unique to you. As with many other things on the Internet, there is a standard that governs digital certificates—in this case, known as X.509.

How Digital Certificates Ensure Internet Security

A digital certificate is used to guarantee that the person who sends information or email over the Internet or who makes a financial transaction really is who he says he is. This illustration shows how a digital certificate would be used to guarantee that the person sending an email is being truthful about his identity.





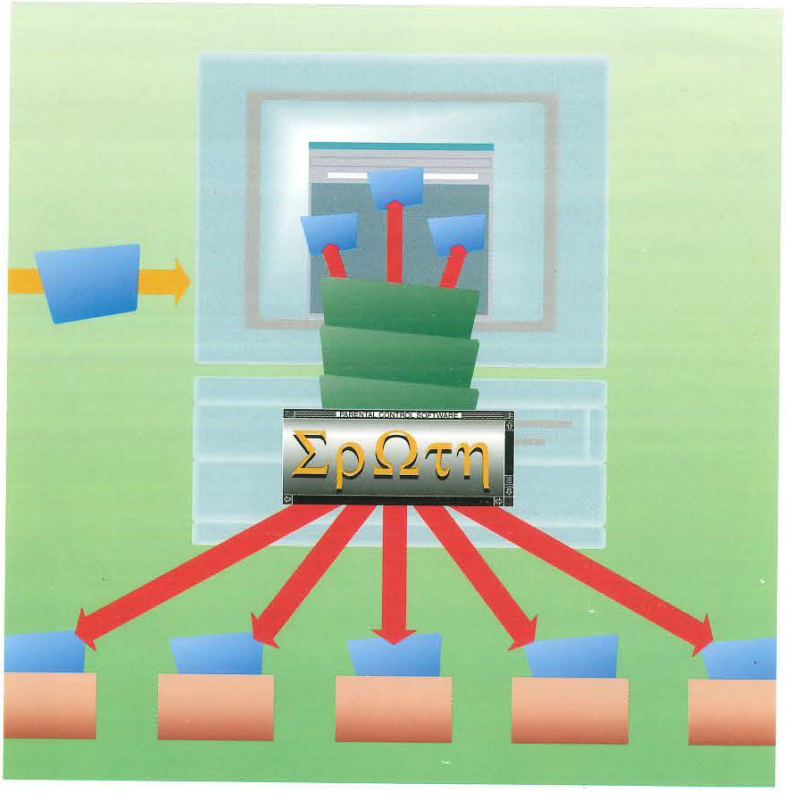
4 When you want to send email to someone and have them know for certain that it is you and no one else who has sent the mail, you attach the digital certificate to your email message. One of the things that the certificate does is sign the message with a private key that you were given as part of the digital certificate.



5 The person you're sending email to gets your digital certificate along with your email. The key is used to read the private key's signature. That signature matches information found in the digital certificate, and so the receiver is assured that the message really came from you.

CHAPTER

51 Parental Controls on the Internet



THE very nature of the Internet—the way it allows the free, unfettered flow of information among people—has gotten it a lot of bad publicity. Much has been made of the fact that erotic and pornographic information are available on the Internet, everything from pictures to discussions about subjects that many people find objectionable. The truth is, that kind of content makes up a very small part of what's available on the Internet. Furthermore, the objectionable content is not exactly in public view—you have to do a bit of digging to find it.

However, just the fact that this type of information is available to anyone who wants to see it, including children, has made people uncomfortable. In fact, Congress and other legislative bodies have tried to take steps to ban certain kinds of content from being available on the Internet. As a result of these efforts, a controversial law was passed against online pornography. The law is called the Communications Decency Act, which the Supreme Court ruled unconstitutional.

The real answer to the problem, though, doesn't lie with legislation. The answer lies with technology—software that will allow parents to make sure that their children are not seeing objectionable material. A number of companies make and sell software that will do this, such as SurfWatch, CyberNanny, and CyberPatrol. They each check sites for content and then bar children from getting to those sites that contain content that is unsuitable for them.

Online services such as CompuServe, America Online, and Prodigy have a variety of ways to block access to objectionable material on the Internet. Some allow parents to block children from using services such as the World Wide Web, chat, or newsgroups completely. Others, such as America Online, license technology from software makers like those that manufacture SurfWatch to allow anyone on their service to block Internet sites they don't want their children to visit.

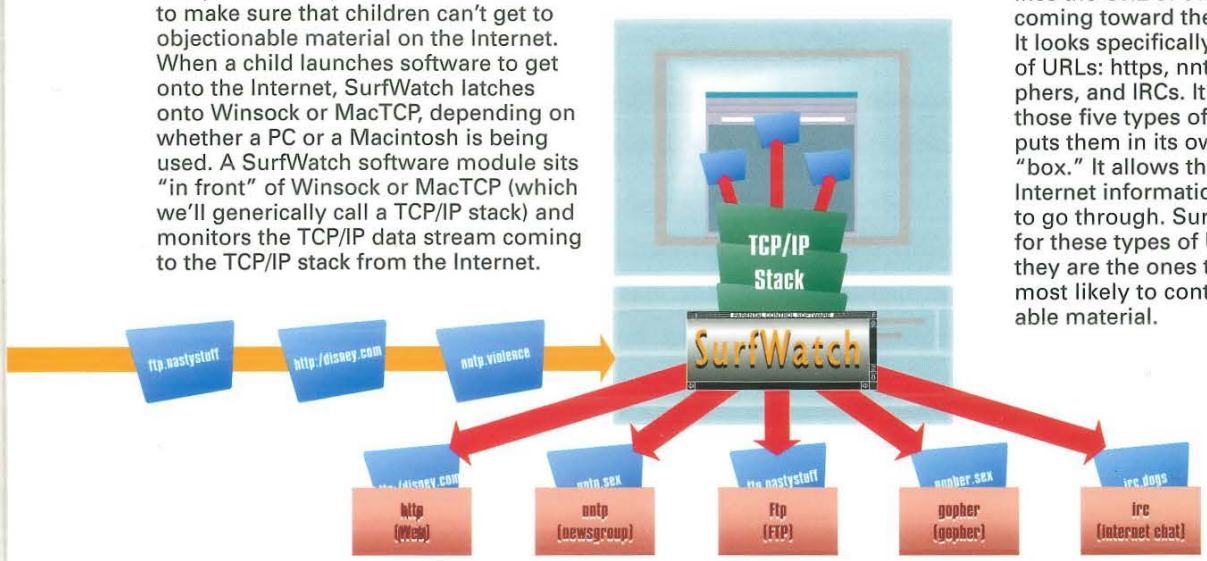
One group working on the issue is PICS (Platform for Internet Content Selection), which is trying to give parents control over the kind of material that their children have access to. The group is trying to develop industry standards for technology that would allow the content of all sites and documents on the Internet to be rated according to its suitability for children. Additionally, the group would create standards to allow software to be developed for blocking sites based on those suitability ratings.

Businesses are also concerned with the type of Internet material their workers are accessing over corporate networks. There is a feeling that getting at and displaying sexual material could be interpreted as sexual harassment. Furthermore, some companies simply don't want their workers accessing that material on company time. Some companies now lease the same software that parents are buying. Instead of installing the software on individual computers, though, the software is installed on a server, and it checks all incoming Internet traffic to every computer in the company.

How Parental Controls Work

1 SurfWatch software is installed on a computer that a parent wants to monitor to make sure that children can't get to objectionable material on the Internet. When a child launches software to get onto the Internet, SurfWatch latches onto Winsock or MacTCP, depending on whether a PC or a Macintosh is being used. A SurfWatch software module sits "in front" of Winsock or MacTCP (which we'll generically call a TCP/IP stack) and monitors the TCP/IP data stream coming to the TCP/IP stack from the Internet.

2 The SurfWatch module examines the URL of every address coming toward the TCP/IP stack. It looks specifically for five kinds of URLs: https, nntps, ftps, gophers, and IRCs. It takes each of those five types of URLs and puts them in its own separate "box." It allows the rest of the Internet information coming in to go through. SurfWatch checks for these types of URLs because they are the ones that are the most likely to contain objectionable material.



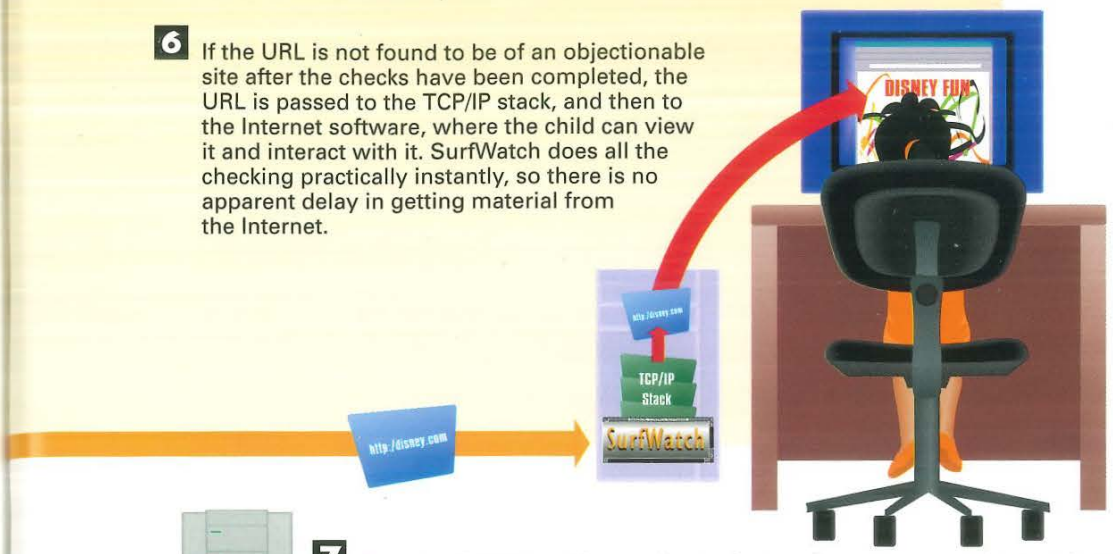


3 Every URL in each of the boxes is checked against a database of the URLs of objectionable sites. If SurfWatch finds that any of the URLs are from objectionable sites, it won't allow that information to be passed on to the TCP/IP stack, blocking the site and preventing information from being viewed. It alerts the child that the site has been blocked. SurfWatch checks thousands of sites and lists in its database the ones that are found objectionable.

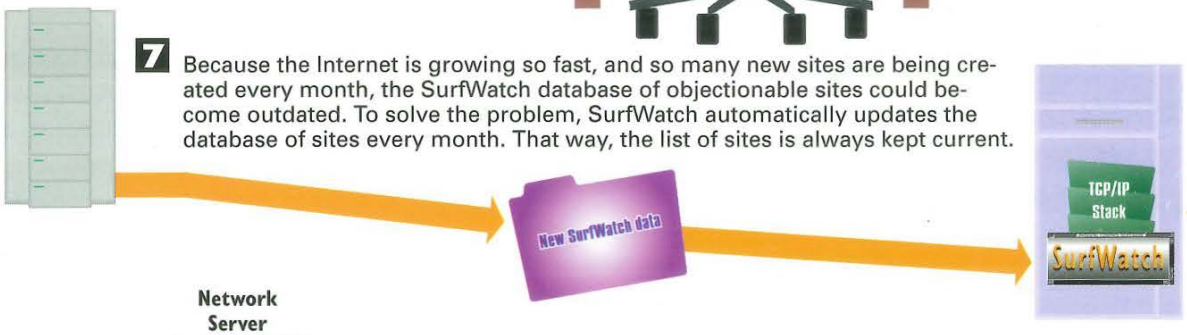
4 If the URL is not in the database, SurfWatch does another check of the URL. This is called *pattern matching*. It looks at the words in the URL and checks them against a database of words to see if any of them indicates a request for objectionable material. Often, people creating objectionable material put representative words in the URL to draw attention to the site. If SurfWatch finds a matching pattern, it won't allow that information to be passed on to the TCP/IP stack, blocking the site and preventing information from being viewed. It alerts the child that the site has been blocked.

5 There is another way that SurfWatch may eventually check for objectionable sites. A rating system called PICS (Platform for Internet Content Selection) is being developed that will embed information about the content in its documents—saying, for example, whether objectionable material can be found there. If SurfWatch uses this system and finds that the URL is of a site that contains objectionable material, it won't allow that information to be passed on to the TCP/IP stack, blocking the site and information from being viewed. It alerts the child that the site has been blocked.

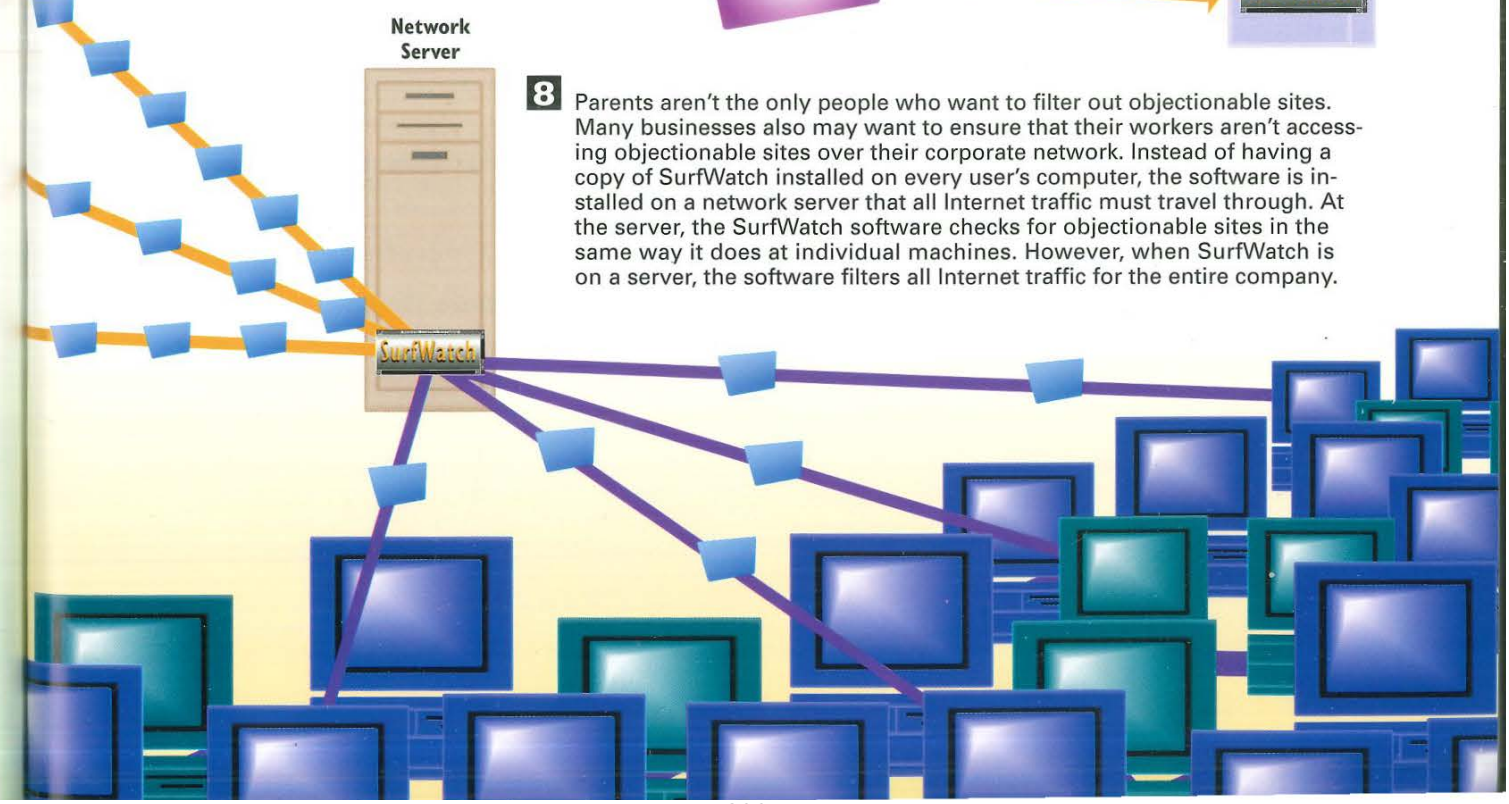
6 If the URL is not found to be of an objectionable site after the checks have been completed, the URL is passed to the TCP/IP stack, and then to the Internet software, where the child can view it and interact with it. SurfWatch does all the checking practically instantly, so there is no apparent delay in getting material from the Internet.



7 Because the Internet is growing so fast, and so many new sites are being created every month, the SurfWatch database of objectionable sites could become outdated. To solve the problem, SurfWatch automatically updates the database of sites every month. That way, the list of sites is always kept current.



8 Parents aren't the only people who want to filter out objectionable sites. Many businesses also may want to ensure that their workers aren't accessing objectionable sites over their corporate network. Instead of having a copy of SurfWatch installed on every user's computer, the software is installed on a network server that all Internet traffic must travel through. At the server, the SurfWatch software checks for objectionable sites in the same way it does at individual machines. However, when SurfWatch is on a server, the software filters all Internet traffic for the entire company.



Symbols

@ (at sign), 17-19
.AU file extension, 217
.com (commercial) domain, 33
. (dot), 17
.edu (education) domain, 33
.gov (government) domain, 33
.mil (military) domain, 33
.MPG file extension, 27
.net (ISPs and networks) domain, 33
.org (organization) domain, 33
.PDF (Adobe Acrobat) file extension, 26
.PS (PostScript) file extension, 26
.WAV file extension, 217
.WAV (Windows) file extension, 27
.WRL file extension, 234-235
.ZIP (PKZIP) file extension, 27

A

absolute hyperlinks, 167
access routers, 269
accounts, shell, 50
Active Desktop (Internet Explorer)
  desktop components, 157
  layers, 156
ActiveX controls, 197
  <OBJECT> tag (HTML), 200-201
  security, 201
addresses
  DNS (Domain Name System), 17
  email
    @ (at sign), 17-19
    . (dot), 17
    harvesting, 92
    LDAP (Lightweight Directory Access Protocol), 88-89
    omitting, 95
  Internet, for telephone calls, 109
  IP (Internet Protocol), 13, 18-19, 34-35, 169
    domain names, 179
    port numbers, 179
    Telnet, 125
    typing, 124
  Telnet, 123-125
ADSL (Asymmetric Digital Subscriber Line), 72
  channels, 73
  DSLs (Digital Subscriber Lines), 71
  Internet, connecting to, 49
  modems, 72-73
  POTS (plain old telephone system), 73
  telephone lines, 22, 73
agents
  excluding, 207
  knowbots, 205
  meta-search software, 136-137
  ROBOTS.TXT file, 207
  Web maintenance spiders, 207
  Web sites, 205-207
AIM (Instant Messenger), buddy lists, 106-107
algorithms, encryption, 289
American Standard Code for Information Interchange, *see* ASCII
Animated GIFs (Graphics Interchange Formats), 237

animations
client pull, 238
files, 27
GIFs (Graphics Interchange Formats), Animated GIFs, 237
server push, 239
Shockwave, 237, 240
viewing, 241
anonymous
FTP (File Transfer Protocol), 128
payment, *see* electronic cash
anti-virus software, 277, 279
AOL (America Online), AIM (Instant Messenger), 106-107
applets (Java), 197-199
applications
CGI (Common Gateway Interface), comparing, 209
helper, 153
whiteboard, 151
architecture, client/server, 41-43
archived messages (newsgroups), 97
ASCII (American Standard Code for Information Interchange), 25
email messages, 80
files, reading, 27
mode for data connections, 129
asymmetric cryptography, 289
Asymmetric Digital Subscriber Line, *see* ADSL
at sign (@), 17-19
ATMs (Automated Teller Machines), 68-69
attachments, email, 80
attacks
smurf, 273-275
viruses, 277-279
audio
files
extensions, 217
playing, 217-219
MBone, multicast IP (Internet Protocol) broadcasts, 230-231
RealAudio, metafiles and streaming, 218-219
RMF (Rich Music Format), 217
streaming, 27, 217
Automated Teller Machines (ATMs), 68-69

B

backbones, 5
networks, connecting, 11
vBNS (very high-speed Backbone Network Services), 6, 9
band signaling, 59
Basic Rate Interface, *see* BRI
bastion hosts, 267, 269
binary
files, 25
converting for email, 79
modes for data connections, 129
BlindRequest, 89
blocking Web sites with parental control software, 297-299
<BODY> tag (HTML), 202
BRI (Basic Rate Interface), channels, 57-58
band signaling, 59

bridges, 9
broadcasts, multicast IP (Internet Protocol), 229-231
browsers, *see* Web browsers
button, Stop, 239
bytecodes, 198
  verifying, 199

C

cables
  coaxial, 62-63
  modems, 51, 62-63
  television, 62
    head ends, nodes, 63
    Usenet newsgroups, 63
    WebTV, RGB (*Red-Green-Blue*), 64
caching data, proxy servers, 271
calls (telephone)
  compressing voice files, 111
  digitalized voice data, 109
  directories, 110
  IP (Internet Protocol), 110, 112-113
  placing on Internet, 109-111
  sending and receiving, 110-111
cameras (digital), NetCams, 221, 226
  JPEG (Joint Photographic Experts Group), 227
capture request, online shopping, 257
cards, network, 13
Cascading Style Sheets (CSS), 163
CDF (Channel Definition Format), 191
Certificate Authorities, 293-295
certificates (digital), 293-295
CGI (Common Gateway Interface), 43
  applications, comparing, 209
  databases and Web sites, linking, 181-183
  directories, 210
  forms, interactive, 209
  imagemaps, 173-175
  Perl (Practical Extraction and Reporting Language), 209-210
  programs, launching, 211
  query strings, 182
  security, directories, 210
  servers, 177, 179
    push, 237
Channel Definition Format (CDF), 191
Channel Properties dialog box (Netscape Netcaster), 192
channels
  ADSL (Asymmetric Digital Subscriber Line), 73
  BRI (Basic Rate Interface), 57-58
    band signaling, 59
  CDF (Channel Definition Format), 191
  IRC (Internet Relay Chat), 103, 105
  Netcasting (Netscape)
    crawling, 193
    subscribing, 192
  PointCast, 194-195
  push technology, 189, 191
  subscribing, 192-193
  Web casting (Microsoft), 190-191
chatting
  AIM (Instant Messenger), 103, 106-107
  IRC (Internet Relay Chat), 103-105
checksum (packets), 14-15

children, parental control software, 297, 299
objectionable Web sites, blocking, 298
choke routers, 268
circuit-switched networks, 13
client-pull animations, 237-238
client/server
architecture, 41-43
models
Gophers, 119-121
IRC (Internet Relay Chat), 103-104
WWW (World Wide Web), 143
modes
FTP (File Transfer Protocol), 127
Telnet, 123
client-side scripting languages, 163
clients
software
downloading, 128-129
Internet, connecting to, 55
Web browsers, 41
coaxial cables, 62-63
codec, video files, 222
Collabra, 155
commands
JavaScript, adding parameters, 203
links, 128-129
Refresh (client pull animations), 238
Common Gateway Interface, *see* CGI
Communications Decency Act, 297
compiled languages, Java
applets, 197-199
bytecodes, 198
verifying, 199
components, ActiveX controls, 197, 200-201
compressing files, 27, 130-131
computers
desktop, 21
dumb terminals, 50
hosts, 21, 42-43
Internet, connecting to, 49-51, 53-55
ISDN (Integrated Services Digital Network), 57-59
monitors, RGB (*Red-Green-Blue*), 64
NCs (Network Computers), 51, 67
telephone lines, connecting, 22-23
terminal emulation, 50
WWW (World Wide Web), connecting to, 21
cookies, 281-283
countries outside U.S., domain names, 33
crawlers (search engines), 134
channels, Netcasting (Netscape), 193
Web pages, 190
credit card transactions, online shopping, 255, 258-259
cryptography, 289-291
CSS (Cascading Style Sheets), 163
customizing receiving information, 189-190
CDF (Channel Definition Format), 191
Netcasting (Netscape), 193
PointCast, 194-195
subscribing to channels, 192
Web casting (Microsoft), 191
cyber-dollars (electronic cash), 255, 258-261
CyberNanny and CyberPatrol (parental control software), 297

D

daemons, FTP (File Transfer Protocol), 128
data
  connections, modes, 129
  corrupted in packets, 15
  repeaters, 9
Data Encryption Standard (DES), 289
databases
  CGI (Common Gateway Interface), 209-211
  on intranets, 248
  Web sites, linking, 181-183
decompressing files, 27, 131
dedicated telephone lines, 10
DES (Data Encryption Standard), 289
desktops
  computers, 21
  videoconferencing, 221, 249, 251-252
    reflectors, 224-225
DHTML (Dynamic HTML), 159, 162-163
dialog boxes, Channel Properties (Netscape Netcaster), 192
digital
  cameras, NetCams, 221, 226
    JPEG (Joint Photographic Experts Group), 227
  certificates, 293-295
  envelopes (encryption), 290
  signatures, 290-291
  switches, ISDN (Integrated Services Digital Network), 58
  voice data, 109
Digital Subscriber Lines (DSLs), 10, 71
direct connections, 50
directories
  CGI (Common Gateway Interface), 210
  LDAP (Lightweight Directory Access Protocol), 88-89
  telephone, 110
  URLs (uniform resource locators), 170
discussion software, 252
DNS (Domain Name System), 17, 19, 179
  servers
    IP (Internet Protocol) addresses, 33-35
    URLs (uniform resource locators), 34-35
    Web browsers, 34-35
document management software, 251, 253
Document Object Model (DOM), 163
documents
  HTML (Hypertext Markup Language), 159
  linking, hypertext and hyperlinks, 165
  Web pages
    DHTML (Dynamic HTML), 159, 162-163
    displaying (parsing), 161
  WWW (World Wide Web)
    file types, 171
    retrieving with URLs (uniform resource locators), 171
DOM (Document Object Model), 163
Domain Name System, *see* DNS
domains, 18-19
  Internet Registrars and Registries, 7
  InterNIC, 7
  names
    changes, tracking, 33
    countries outside U.S., 33, 179
    servers, 33
  organization of, 33

registering, 5
URLs (uniform resource locators), 33, 170
Web sites, 21
see also DNS (Domain Name System)
dot (.), 17
downloading
client software, 128-129
files, 127
see also FTP (File Transfer Protocol); HTTP (Hypertext Transfer Protocol)
DSLs (Digital Subscriber Lines), 10, 71
dumb terminals, 50
dynamic routing tables, 38

E

EBCDIC (Extended Binary Coded Decimal Interchange Code), 25
EGP (Exterior Gateway Protocol), 38
electronic cash, 255, 258-261
email
addresses
@ (at sign), 17-19
. (dot), 17
harvesting, 92
LDAP (Lightweight Directory Access Protocol), 88-89
omitting your address, 95
attachments, 80
binary files, converting, 79
DNS (Domain Name System), 17
files
attaching, 79
encoding and decoding, 80
firewalls, 87
gateways, 79
Internet, 82
intranets, 248
LDAP (Lightweight Directory Access Protocol), 79, 88
BlindRequest, 89
list servers (listservs), 79
mail reflector, 83
mailing lists, 84
listservs, 83
moderated and unmoderated, 79
subscribing and canceling, 86
messages
ASCII (American Standard Code for Information Interchange), 80
preparing, 81
MIME (Multipurpose Internet Mail Extensions), 79
Netscape Messenger, 155
readers, 84
sending between networks, 87
signature areas, 81
software, 84-85
Spam
blockers and filters, 91, 94-95
sending, 92-93
Spamouflauge, 93
uuencode, 79
<EMBED> tag (HTML), 241
emulation (terminals), VT-100, 123-125
encryption, 289-291
digital certificates, 293-295
online shopping, 255-257

credit card transactions, 258-259
electronic cash, 260-261
envelopes, IP (Internet Protocol), 14-15
eradication programs, anti-virus software, 277, 279
Ethernet networks, 10
Extended Binary Coded Decimal Interchange Code (EBCDIC), 25
extensions of files
.AU, 217
compressed, 130
.MPG, 27
.PDF (Adobe Acrobat), 26
.PS (PostScript), 26
.WAV, 217
.WAV (Windows), 27
.WRL, 234-235
.ZIP (PKZIP), 27
Exterior Gateway Protocol (EGP), 38
exterior screening routers, 269

F

FAQs (frequently asked questions), 27, 97
File Transfer Protocol, *see* FTP
files
Adobe Acrobat, .PDF extension, 26
animation, 27
ASCII (American Standard Code for Information Interchange), 25, 27
audio, playing, 217-219
binary, 25
converting for email, 79
posting to newsgroups, 97
compressing, 27, 130-131
decompressing, 27
downloading, 127
EBCDIC (Extended Binary Coded Decimal Interchange Code), 25
email
attaching, 79
encoding and decoding, 80
extensions
.AU, 217
compressed, 130
.MPG, 27
.PDF (Adobe Acrobat), 26
.PS (PostScript), 26
.WAV, 217
.WAV (Windows), 27
.WRL, 234-235
.ZIP (PKZIP), 27
formats, graphics, 26
FTP (File Transfer Protocol), downloading, 127
GIF (Graphics Interchange Format), 26
headers, 26
HTTP (Hypertext Transfer Protocol), downloading, 127
JPEG (Joint Photographic Experts Group), 26
MIME (Multipurpose Internet Mail Extensions), 79
MPEG (Motion Picture Experts Group), 27
multimedia, 27
helper applications, 153
plug-ins, 153
viewing, 25
NCs (Network Computers), saving, 67, 69
PKZIP, 27
PostScript, 25

printing, 25
readers, 26
ROBOTS.TXT, 207
streaming audio, 27
types, 25-27
uuencode, 79
videos, codec, 222
viewing, online and offline, 25
voice, compressing, 111
Web browsers, cannot be displayed, 153
Windows sound, 27
WWW (World Wide Web) documents, x-html or x-text, 171
financial transactions, *see* online, shopping
firewalls, 54-55, 87, 247-248, 267-269
forms, interactive, 173-175, 209
From line (email), 81
FTP (File Transfer Protocol), 119
anonymous, 128
client software, downloading, 128-129
client/server models, 127
command links, 128-129
daemons, 128
data connections, modes, 129
files, downloading, 127
servers, command links, 128-129
functions, interactive forms, 173-175
funding, Internet, 5-6

G

gateways, 9
CGI (Common Gateway Interface), 173-175
email, 79
voice, 113
GIF (Graphics Interchange Format), 26
Animated GIFs, 237
gophers
client/server models, 119-121
FTP (File Transfer Protocol), 119
Internet, connecting to, 53
root, 119
selectors, 120
Telnet, 120
user display strings, 121
graphics
files
GIF (Graphics Interchange Format), 26, 237
JPEG (Joint Photographic Experts Group), 26, 227
imagemaps, 173-175
Graphics Interchange Format, *see* GIF
groupware, 249, 251-253

H

hackers, smurf attacks, 273-275
harvesting email addresses, 92
hash functions (encryption), 290
head ends (cable television), nodes, 63
headers, files, 26
compressing, 130
heads (packets), 14
Helper applications, 153
hierarchical organization
Usenet newsgroups, 100-101
Web sites, 147

home pages
hyperlinks, 146-147
Web sites, 143
hop counts, 37
hosts, 123-125
computers, 21, 42-43
server software, 177-179
HTML (Hypertext Markup Language), 144-145, 160-161
DHTML (Dynamic HTML), 159
imagemaps, 173
Internet, searching, 135
tags, 153
<BODY>, 202
<EMBED>, 241
<IMG>, 239
<META>, 237-238
<OBJECT>, 200-201
Web browsers, 152-153
Web pages, 143, 151
HTTP (Hypertext Transfer Protocol), 41-43, 127, 145, 169
Web documents, retrieving with URLs (uniform resource locators), 171
hubs, 9
hyperlinks, 165-166
home pages, 146-147
imagemaps, 173
relative and absolute, 167
hypertext, 165
Hypertext Markup Language, *see* HTML
Hypertext Transfer Protocol, *see* HTTP

I

IAB (Internet Activities Board), Internet Engineering Task Force, 5
ICE (Internet Content and Exchange standard), 281, 284
imagemaps, 173-175
<IMG> tag (HTML), 239
in-band signaling, 59
indexes, searching Internet, 133
information
customizing receiving, 189, 190, 192
CDF (Channel Definition Format), 191
Netcasting (Netscape), 193
PointCast, 194-195
Web casting (Microsoft), 191
Internet, sending, 9
repeaters, 9
routers, 9
see also packets
inoculators, anti-virus software, 277
input
ports, 38
queues, 37, 39
instant messaging, 103, 106-107
integrated services digital network, *see* ISDN
interactive forms, 173-175, 209
interfaces
BRI (Basic Rate Interface), 57
CGI (Common Getaway Interface), 173-175
proprietary, 53
user, 53
internal screening routers, 268

Internet
addresses
@ (at sign), 17-19
. (dot), 17
backbones, vBNS (very high-speed Backbone Network Services), 6
bridges, 9
domains
InterNIC, 7
registering, 5
Registrars and Registries, 7
funding, 5-6
hubs, 9
IAB (Internet Activities Board), 5
ICE (Internet Content and Exchange standard), 281, 284
information, sending in packets, 9
Internet Control Message Protocol, 274
Internet Engineering Task Force, TCP/IP (Transmission Control Protocol/Internet Protocol), 5
Internet Explorer, Active Desktop
desktop components, 157
layers, 156
Internet Protocol, *see* IP
Internet Relay Chat, *see* IRC
Internet service providers, *see* ISPs
Internet Society, 5, 7
InterNIC, 7
LANs (local area networks), 9
networks, 5-6
packets, 9
passports, 281, 284-285
Registrars and Registries, 7
RFCs (requests for comments), 5
running, 6-7
searching
indexes, 133
meta-search software, 136-137
search engines, 133-135
TCP (Transmission Control Protocol), 9
TCP/IP (Transmission Control Protocol/Internet Protocol), 5, 7
television, merging, 62
W3C (World Wide Web Consortium), 5
see also Web browsers; Web pages; Web sites; WWW (World Wide Web)
InterNIC, 7
intranets, 247-249
workgroup applications, 251-253
IP (Internet Protocol), 13, 18-19
addresses, 34-35, 169
domain names, 179
port numbers, 179
Telnet, 125
typing, 124
DNS (Domain Name System), 33
envelopes, 14-15
multicast broadcasts, 229-231
packets, routers, 37
telephony, 112-113
voice gateways, 113
WWW (World Wide Web), connecting to, 23
IRC (Internet Relay Chat)
AIM Instant Messenger), buddy lists, 106-107
channels, 103, 105
client/server models, 103-104
Internet, connecting to, 53

ISDN (integrated services digital network), 11
Internet, connecting to, 49, 51, 57-59
lines, 22
MP (Multilink PPP), 59
PPP (Point-to-Point Protocol), 59
ISPs (Internet service providers), 5, 7, 21
Internet, connecting to, 49, 53
smurf attacks, 273-275
Spam, blocking, 94-95
user interfaces, 53
Web pages, posting, 149
WWW (World Wide Web), connecting to, 22-23

J

Java
applets, 197-199
bytecodes, 198
verifying, 199
JavaScript, 197
<BODY> tag (HTML), 202
commands, adding parameters, 203
objects, methods, 203
JPEG (Joint Photographic Experts Group), 26
videos, NetCams, 227
junk mail, *see* Spam

K-L

Kbps (kilobits-per-second), 22
keys, encryption, 289
knowbots, *see* agents

languages
compiled, Java, 198-199
markup
DHTML (Dynamic HTML), 159
HTML (Hypertext Markup Language), 159
modeling, VRML (Virtual Reality Modeling Language), 233-235
object-oriented, JavaScript, 202-203
programming, Perl (Practical Extraction and Reporting Language), 209-210
scripting, 210
client-side, 163
JavaScript, 197, 202-203
LANs (local area networks)
bridges and gateways, 9
Internet, connecting to, 49-50
leased lines, 5
LDAP (Lightweight Directory Access Protocol), 79
BlindRequest, 89
email, 88-89
leased lines (networks), 5
Lightweight Directory Access Protocol (LDAP), 79
linear (Web sites)
organization, 147
structures, 143
lines (telephone)
ADSL (Asymmetric Digital Subscriber Line), 22, 73
backbones, 5
dedicated, 10
DSLs (Digital Subscriber Lines), 71
Internet, connecting to, 49
ISDN (integrated services digital network), 22
leased, 5
POTS (plain old telephone service), 22
T1 and T3, 10

WWW (World Wide Web), connecting to, 22-23
linking
databases and Web sites, 181-183
networks, 10-11
see also hyperlinks
Links command, 128-129
list servers (listservs), 79, 83
local area networks, *see* LANs

M

Macintosh QuickTime, 27
mailing
lists (listservs), 83
moderated and unmoderated, 79
subscribing and canceling, 86
mailers, 84
reflectors, 83
markup languages
DHTML (Dynamic HTML), 159, 162-163
HTML (Hypertext Markup Language), 144-145, 160-161
<BODY> tag, 202
<EMBED> tag, 241
imagemaps, 173
<IMG> tag, 239
Internet, searching, 135
<META> tag, 237-238
<OBJECT> tag, 200-201
tags, 153
Web browsers, 152-153
Web pages, 143, 151
Massachusetts Institute of Technology (MIT), W3C (World Wide Web Consortium), 5
MBone (Multicast Backbone), 221
mrouted software, 231
multicast IP (Internet Protocol) broadcasts, 229-231
tunnels, 231
media, Web pages, 149
memory, RAM (random-access memory) and ROM (read-only memory), 69
messages
email
ASCII (American Standard Code for Information Interchange), 80
delivering over Internet, 82
preparing, 81
digest (encryption), 290
instant messaging, 103, 106-107
messaging software, 251
newsgroups
archived, 97
moderated and unmoderated, 97-98
<META> tag (HTML), 237-238
metafiles, RealAudio, 218-219
meta-information, 237
meta-search software, searching Internet, 136-137
methods (JavaScript), 203
MIME (Multipurpose Internet Mail Extensions), 79
MIT (Massachusetts Institute of Technology), W3C (World Wide Web Consortium), 5
modeling languages, VRML (Virtual Reality Modeling Language), 233-235

modems (MOdulate-DEModulate)
ADSL (Asymmetric Digital Subscriber), 71-73
cable, 51, 62-63
DSLs (Digital Subscriber Lines), 71
ISDN (Integrated Services Digital Network), 58
moderated and unmoderated newsgroup messages, 97-98
moderated mailing lists, 79
modes for data connections, 129
MOdulate-DEModulate, *see* modems
monitors, RGB (Read-Green-Blue), 64
Motion Picture Experts Group (MPEG), 27
mouse, x and y coordinates, 173-175
MP (Multilink PPP), 59
MPEG (Motion Picture Experts Group), 27
mrouted software, 231
Multicast
Backbone, *see* MBone
IP (Internet Protocol) broadcasts, 229-231
protocols, 230-231
multimedia files, 27
helper applications, 153
plug-ins, 153
viewing, 25
see also audio
Multipurpose Internet Mail Extensions (MIME), 79

N

name servers, 33
NAP (Network Access Point), 9, 11
National Television Standards Committee (NTSC), 63-64
NCs (Network Computers), 51
ATMs (Automated Teller Machines), 68-69
files, saving, 67, 69
PINs (personal identification numbers), 67-68
RAM (random-access memory) and ROM (read-only memory), 69
servers, 67
Nelson, Ted, 165
NetCams (digital cameras), 221, 226
JPEG (Joint Photographic Experts Group), 227
netcasting (Netscape), 189
Channel Properties dialog box (Netscape Netcaster), 192
channels, crawling, 193
Netscape Communicator, 154-155
Network Access Point (NAP), 9, 11
Network Computers, *see* NCs
Network Virtual Terminal (NVT), 125
networks
backbones, 5, 11
circuit-switched, 13
DSL (Digital Subscriber Line), 10
email, sending, 87
Ethernet, 10
firewalls, 87
intranets, 247-249
workgroup applications, 251-253
ISDN (integrated services digital network), 11

LANs (local area networks), 9
linking, 10-11
local and regional, leased lines, 5
NAP (Network Access Point), 11
network cards, 13
packet-switched, 13
PPP (Point-to-Point Protocol), 13
regional, 6
satellite links, 11
SLIP (Serial Line Internet Protocol), 13
sockets, 13
TCP/IP (Transmission Control Protocol/ Internet Protocol)
mechanics, 14-15
stacks, 13
token-ring, 10
WANs (wide area networks), 9
Winsock, 13
see also security
news agents, 206-207
newsgroups, 98
binary files, 97
FAQs (Frequently Asked Questions), 97
hierarchy, 100-101
identifying, 101
messages
archived, 97
moderated and unmoderated, 97-98
reader software, 99
servers, 99
subscribing and cancelling, 97
threads, 97
uuencode, 97, 99
nodes (cable television), 63
NTSC (National Television Standards Committee), 63-64
NVT (Network Virtual Terminal), 125

O

<OBJECT> tag (HTML), 200-201
object-oriented languages, JavaScript, 197
<BODY> tag (HTML), 202
commands, adding parameters, 203
objects, methods, 203
offline or online, viewing files, 25
online
services
firewalls, 54-55
Internet, connecting to, 51, 53-55
shopping, 255-257
electronic cash, 258-261
OPS (Open Profiling Standard), 281, 284, 286
organizational structure, Web sites, 143
out-of-band signaling, 59

P

P3P (Platform for Privacy Preferences), 281, 284
Packet Internet Groper (ping), smurf attacks with, 273-275
packets
checksum, 14-15
corrupted data, 15
email, delivering, 82
filtering, 267, 269
headers, 14
Internet, telephone calls, 109, 111

IP (Internet Protocol) envelopes, 14-15
networks, packet-switched, 13
routers, 9, 37, 39
TCP (Transmission Control Protocol), 37, 39
TCP/IP (Transmission Control Protocol/Internet Protocol), 13
pages, *see* Web pages
parental control software, 297-299
parsing Web pages, 161
passports, 281, 284-285
pattern matching, blocking Web sites, 298
payment gateway, online shopping, 256-257
PDAs (personal digital assistants), 21
Perl (Practical Extraction and Reporting Language), 209-210
personal digital assistants (PDAs), 21
personal identification numbers (PINs), 67-68
phones, *see* telephones
PICS (Platform for Internet Content Selection), 297-298
ping (Packet Internet Groper), smurf attacks with, 273-275
PINs (personal identification numbers), 67-68
PKZIP, .ZIP extension, 27
Platform for Internet Content Selection (PICS), 297-298
Platform for Privacy Preferences (P3P), 281, 284
plug-ins, 153
Shockwave, 237
animations, 240-214
VR (virtual reality), 235
PointCast
channels, 194-195
URLs (uniform resource locators), 195
Point-to-Point Protocol, *see* PPP
ports
input, 38
numbers, 179
routers, 37
posting Web pages, 149
PostScript, files and printers, 25
POTS (plain old telephone service), 22
ADSL (Asymmetric Digital Subscriber Line), 73
PPP (Point-to-Point Protocol), 13, 59
Internet, connecting to, 49, 51
Practical Extraction and Reporting Language (Perl), 209-210
printers, PostScript, 25
printing files, 25
privacy issues, 281
cookies, 282-283
cryptography, 289-291
online shopping, 255, 258-261
parental control software, 297-299
passports, 284-285
Web tracking software, 286-287
programming
languages, Perl (Practical Extraction and Reporting Language), 209-210
tools
ActiveX, 197, 200-201
Java, 197-199
JavaScript, 197, 202-203

proprietary interfaces, 53
protocols
  EGP (Exterior Gateway Protocol), 38
  FTP (File Transfer Protocol), 119, 127
  HTTP (Hypertext Transfer Protocol), 41, 127
  IP (Internet Protocol), 13, 18-19
    telephony, 112-113
  LDAP (Lightweight Directory Access Protocol), 79
  MP (Multilink PPP), 59
  multicast, 230-231
  PPP (Point-to-Point Protocol), 13, 49
  RIP (Routing Information Protocol), 38
  SLIP (Serial Line Internet Protocol), 13
  TCP (Transmission Control Protocol), 13
    handling packets, 37, 39
  TCP/IP (Transmission Control Protocol/Internet Protocol), 5, 7
    Internet Engineering Task Force, 5
    mechanics, 14-15
    packets, 13
  UDP (User Datagram Protocol), 219
proxy servers, 63, 267, 270-271
PSTN (public switched telephone network), 112-113
public-key cryptography, 289
push technology
  channels, 189
    CDF (Channel Definition Format), 191
    Netcasting (Netscape), 192-193
    PointCast, 194-195
    subscribing, 192-193
    Web casting (Microsoft), 190-191
  Netcasting (Netscape), 189, 192-193
  Web casting (Microsoft), 189-190
    CDF (Channel Definition Format), 191

Q-R

query strings, 182
queues, input, length of, 37

RAM (random access memory), 21, 69
random structure (Web sites), 143
read-only memory (ROM), 69
readers, 84
  files, viewing, 25-26
RealAudio
  metafiles, 218-219
  UDP (User Datagram Protocol), 219
Red-Green-Blue (RGB), 64
reflectors, 224-225
Refresh command (client pull animations), 238
regional networks, 6
  leased lines, 5
registering domains, 5
relative hyperlinks, 167
repeaters, 9
requests for comments (RFCs), 5
response headers, 237
RFCs (requests for comments), 5
RGB (Red-Green-Blue), 64
Rich Music Format (RMF), 217
RIP (Routing Information Protocol), 38
RMF (Rich Music Format), 217

robots, *see* agents
ROBOTS.TXT file, 207
ROM (read-only memory), 69
root gophers, 119
routers, 10
  access, 269
  choke, 268
  dynamic routing tables, 38
  EGP (Exterior Gateway Protocol), 38
  email
    blocking Spam, 95
    delivering, 82
  hop counts, 37
  input ports, 37-38
  input queues, 37, 39
  packets, 37, 39
    managing, 9
  RIP (Routing Information Protocol), 38
  routing
    dynamic, 38
    processes, 37
    protocols, 38
    tables, 37-39
  screening routers, 269
  TCP (Transmission Control Protocol), 37, 39
Routing Information Protocol (RIP), 38
RSA (public-key encryption), 289

S

satellite links, 11
saving files, NCs (Network Computers), 67, 69
scanners, anti-virus software, 277, 279
screening routers, 268-269
scripting languages, 210
  client-side, 163
  JavaScript, 197
    <BODY> tag (HTML), 202
    commands, adding parameters, 203
    objects, methods, 203
search engines
  crawlers or spiders, 134
  HTML (Hypertext Markup Language) pages, 135
  indexes, 133
  Internet, 133
  meta-search software, agents, 136-137
  URLs (Uniform Resource Locators), 134-135
secret-key cryptography, 289
Secure Electronic Transaction protocol (SET), 255-256
security
  ActiveX controls, 201
  bastion hosts, 267, 269
  digital certificates, 293-295
  directories, CGI (Common Gateway Interface), 210
  firewalls, 247-248, 267-269
  online shopping, 255-257
    credit card transactions, 258-259
    electronic cash, 260-261
  privacy issues, 281
    cookies, 282-283
    cryptography, 289-291
    parental control software, 297-299
    passports, 284-285
    Web tracking software, 286-287
  proxy servers, 267, 270-271

smurf attacks, 273-275
viruses, 277-279
Web browsers, 201
selectors (Gophers), 120
Serial Line Internet Protocol, *see* SLIP
servers
AIM (Instant Messenger), 106-107
CGI (Common Gateway Interface), 177, 179
client/server architecture, 41-43
connecting to, 41
DNS (Domain Name System), 34-35
FTP (File Transfer Protocol), 127-129
Gophers, 119-121
hosts, 123-125
IP (Internet Protocol) addresses, port numbers, 179
IRC (Internet Relay Chat)
channels, 103, 105
client/server models, 103-104
list servers (listservs), 79
names (domains), 33
NCs (Network Computers), 67
proxy, 63, 267, 270-271
push, 237
animations, 239
software, 177-179
Telnet, 123
Usenet, newsgroups, 99
WWW (World Wide Web), 143
CGI (Common Gateway Interface), 209-211
documents, retrieving with URLs (uniform resource locators), 171
SET (Secure Electronic Transaction protocol), 255-256
shell accounts, 50
Shockwave, 237
animations, 240-241
shopping
agents, 206-207
online, 255-257
electronic cash, 258-261
signature areas (email), 81
sites, Web
accessing, 21
agents, 205-207
blocking with parental control software, 297-299
channels
Netcasting (Netscape), 192-193
PointCast, 194-195
push technology, 189-191
subscribing, 192-193
Web casting (Microsoft), 190-191
databases, linking, 181-183
home pages, 143
hosts, server software, 177-179
organizational structure, 143, 147
privacy issues, 281
cookies, 282-283
passports, 284-285
Web tracking software, 286-287
push technology, 189
CDF (Channel Definition Format), 191
Netcasting (Netscape), 192-193
PointCast, 194-195
Web casting (Microsoft), 190-191

servers, software, 177-179
URLs (uniform resource locators), 170
Web maintenance spiders, 207
Yahoo!, 133, 181
ZDNet Software Library, 128
see also Internet; Web browsers; Web pages; WWW (World Wide Web)
SLIP (Serial Line Internet Protocol), 13
Internet, connecting to, 49
smurf attacks, 273-275
sniffers, Web tracking software, 286-287
sockets, 13
software
clients
downloading, 128-129
Internet, connecting to, 55
downloading, 127
email, 84-85
Gophers, 119-120
meta-search, agents, 136-137
mrouted, 231
readers for newsgroups, 99
servers, 177-179
sockets, 13
suites for Web browsers, 151
TCP/IP (Transmission Control Protocol/Internet Protocol) stacks, 13
telephone directories, 110
tracking, 281
Winsock, 13
sounds, *see* audio
Spam (Spammers)
email
blocking, 91, 94-95
sending, 92-93
filters and blockers, 94-95
Spamouflauge, 93
spiders, *see* agents
static routing tables, 39
Stop button, 239
streaming
audio, 27, 217
RealAudio, metafiles, 218-219
video, 27, 221-222
UDP (User Datagram Protocol), 223
structural organization (Web sites), 147
Sun Microsystems, DNS (Domain Name System), 17
SurfWatch (parental control software), 297-298
switches, digital, ISDN (Integrated Services Digital Network), 58
symmetric cryptography, 289

T

T1 and T3 telephone lines, 10
tables, routing, 37
dynamic, 38
static, 39
tags
HTML
<BODY>, 202
<EMBED>, 241

<IMG>, 239
<META>, 237-238
<OBJECT>, 200-201
markup languages, 159
Web pages, tagging, 160-161
TCP (Transmission Control Protocol), 9, 13
packets, handling, 37, 39
TCP/IP (Transmission Control Protocol/Internet Protocol), 5, 7, 13
email, delivering, 82
Internet, connecting to, 53
Internet Engineering Task Force, 5
mechanics, 14-15
packets, 13
checksum, 14-15
stacks, 13
telephone calls, digitalized voice data, 109, 111
Web documents, retrieving with URLs (uniform resource locators), 171
telephones
calls
compressing voice files, 111
digitalized voice data, 109
directories, 110
IP (Internet Protocol), 110, 112-113
placing on Internet, 109-111
sending and receiving, 110-111
lines
ADSL (Asymmetric Digital Subscriber Line), 22, 73
backbones, 5
dedicated, 10
DSLs (Digital Subscriber Lines), 71
Internet, connecting to, 49
ISDN (integrated services digital network), 22
leased, 5
POTS (plain old telephone service), 22
T1 and T3, 10
WWW (World Wide Web), connecting to, 22-23
PSTN (public switched telephone network), 112-113
television
cable, 62-63
Internet, merging, 61-65
NTSC (National Television Standards Committee), 64
VPN (Virtual Private Network), 65
WebTV, 64-65
Telnet
addresses, typing 123-125
client/server models, 123
data, sending, 125
gophers, 120
hosts, 123-125
Internet, connecting to, 53
IP (Internet Protocol) addresses, 124-125
NVT (Network Virtual Terminal), 125
servers, hosts, 123-125
terminal emulation, VT-100, 123-125
terminals
dumb, 50
emulation, 50
VT-100 emulation, 123-125
text, HTML (Hypertext Markup Language), 159

threads (newsgroups), 97
To line (email), 81
token-ring networks, 10
tokens (file compression), 130
tools
  agents, 205-207
  CGI (Common Gateway Interface), 209-211
  FTP (File Transfer Protocol), 127-129
  gophers, 119-121
  indexes, searching Internet, 133
  meta-search software, searching Internet, 136-137
  programming
    ActiveX, 197, 200-201
    Java, 197-199
    JavaScript, 197, 202-203
  push technology, 189
    Netcasting (Netscape), 192-193
    PointCast, 194-195
    Web casting (Microsoft), 190-191
  search engines, 133-135
  Telnet, 123-125
tracking software, 281
Transmission Control Protocol, *see* TCP
Transmission Control Protocol/Internet Protocol, *see* TCP/IP
transmissions, MBone, 229-231
tree structures (Web sites), 143
Trojan horses (viruses), 277
tunnels, MBone, 231
TV, *see* television

U

UDP (User Datagram Protocol), 219
  videoconferencing, 225
  videos, streaming, 223
Uniform Resource Locators, *see* URLs
unmoderated and moderated newsgroup messages, 97-98
unmoderated mailing lists, 79
URLs (Uniform Resource Locators), 33-35, 134-135
  CGI (Common Gateway Interface), launching programs, 211
  HTTP (Hypertext Transfer Protocol), 169
  imagemaps, 173-175
  interactive forms, 173-175
  parts of, 145
  PointCast, 195
  spiders, 134
  structure, 170
  Web documents, retrieving, 171
Usenet, 155
  Internet, connecting to, 53
  newsgroups, 63, 97-98
    hierarchy, 100-101
    identifying, 101
    reader software, 99
    servers, 99
User Datagram Protocol, *see* UDP
users
  display strings (gophers), 121
  interfaces, 53
uuencode, 79, 97, 99

V

vBNS (very high-speed Backbone Network Services), 6, 9
VeriSign (Certificate Authority), 293
videoconferencing, 221, 249, 251-252
  reflectors, 224-225
videos
  files, codec, 222
  MBone (Multicast Backbone), 221
    multicast IP (Internet Protocol) broadcasts, 230-231
  NetCams (digital cameras), 221, 226
    JPEG (Joint Photographic Experts Group), 227
  streaming, 221-222
    UDP (User Datagram Protocol), 223
  videoconferencing, 221
    reflectors, 224-225
    UDP (User Datagram Protocol), 225
  whiteboard applications, 221
Virtual Private Network (VPN), 65
Virtual Reality Modeling Language (VRML), 233-235
virtual reality, *see* VR
viruses, 277-278
  Java applets, detecting, 199
  markers, 279
v-markers (virus markers), 279
voices
  files, compressing, 111
  gateways, 113
VPN (Virtual Private Network), 65
VR (virtual reality), 233-234
  plug-ins, 235
VRML (Virtual Reality Modeling Language), 233-235
VT-100 emulation, 123-125

W

W3C (World Wide Web Consortium), 5
WANs (wide area networks), 9
Web browsers, 21, 34-35, 42-43, 55, 144-145
  CDF (Channel Definition Format) files, 191
  clients, 41
  Collabra, 155
  documents, displaying (parsing), 161
  files, cannot be displayed, 153
  helper applications, 153
  HTML (Hypertext Markup Language), 152-153
  information, processing, 178-179
  Internet Explorer, Active Desktop, 156-157
  Java applets, 199
  JavaScript, 202-203
  Netscape
    Conference, 155
    Messenger, 155
    Navigator, 154
  plug-ins, 153
  security, 201
  Stop button, 239
  suites of software, 151
  Usenet, 155
  WebTV, 65
  whiteboard applications, 151, 155

windows, opening, 203
see also Internet; Web pages; Web sites; WWW (World Wide Web)
Web pages, 143
<BODY> tag (HTML), 202
client-side scripting languages, 163
crawling, 190
creating, 148-149
CSS (Cascading Style Sheets), 163
DHTML (Dynamic HTML), 159, 162-163
DOM (Document Object Model), 163
forms, interactive, 173-175
home pages, 143
hyperlinks, 146-147
HTML (Hypertext Markup Language), 143, 151, 159
tagging, 160-161
hyperlinks, 165-166
relative and absolute, 167
hypertext, 165
imagemaps, 173-175
JavaScript, 202-203
markup languages, tags, 159
media, 149
meta-information, 237
<OBJECT> tag (HTML), 200-201
organizing, 146-147
parsing, 161
posting, 149
response headers, 237
tagging, 160-161
URLs (uniform resource locators), 169-170
see also Internet; Web browsers; Web sites; WWW (World Wide Web)
Web servers, CGI (Common Gateway Interface), 209-211
Web sites
accessing, 21
agents, 205-207
blocking with parental control software, 297-299
channels
Netcasting (Netscape), 192-193
PointCast, 194-195
push technology, 189-191
subscribing, 192-193
Web casting (Microsoft), 190-191
databases, linking, 181-183
home pages, 143
hosts, server software, 177-179
organizational structure, 143, 147
privacy issues, 281
cookies, 282-283
passports, 284-285
Web tracking software, 286-287
push technology, 189-190
CDF (Channel Definition Format), 191
Netcasting (Netscape), 192-193
PointCast, 194-195
Web casting (Microsoft), 190-191
servers, software, 177-179
URLs (uniform resource locators), 170
Web maintenance spiders, 207
Yahoo!, 133, 181
ZDNet Software Library, 128
see also Internet; Web browsers; Web pages; WWW (World Wide Web)

WebTV, 64
Internet, accessing, 51, 65
ISPs (Internet service providers), 65
POPs (points of presence), 65
VPN (Virtual Private Network), 65
see also Web browsers
whiteboard
applications, 151, 155, 221
software, 251-252
wide area networks (WANs), 9
windows
files, .WAV extension, 27
Web browsers, opening, 203
Winsock, 13
work flow groupware, 251, 253
workgroup applications, 247, 251-253
World Wide Web Consortium (W3C), 5
World Wide Web, *see* WWW
worms, *see* viruses
WWW (World Wide Web)
browsing, 55
casting (Microsoft), 189-190
CDF (Channel Definition Format), 191
client/server model, 143
connecting to, 21
telephone lines, 22-23
documents
DHTML (Dynamic HTML), 159, 162-163
HTML (Hypertext Markup Language), 144-145, 159-161, 171
HTTP (Hypertext Transfer Protocol), 145
IP (Internet Protocol) addresses, connecting to, 23
ISPs (Internet service providers), connecting to, 23
Netscape
Conference, 155
Messenger, 155
Navigator, 154
software
downloading, 127
tracking, 281, 286-287
spiders, maintaining, 207
URLs (uniform resource locators), 169
components, 145
see also Internet; Web browsers; Web pages; Web sites

X-Z

X coordinates (mouse), 173-175
X.509 standard, digital certificates, 293-294
X-html and X-text files, 171

Y coordinates (mouse), 173-175
Yahoo! Web site, 133, 181

ZDNet Software Library Web site, 128

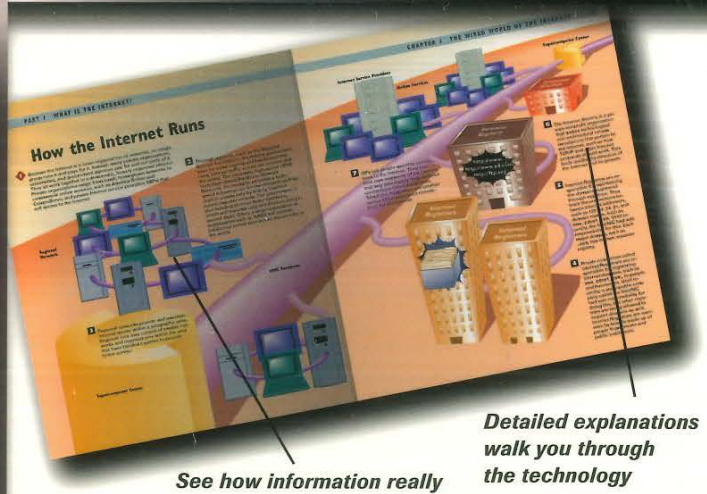
HOW THE INTERNET WORKS

FOURTH EDITION

194
BOOKS
07 44967 194

Gralla

que



See how information really moves on the Internet

Detailed explanations walk you through the technology

The Internet has completely changed the way people work, communicate, socialize, and have fun. All around the world, millions of people log on the cyberspace network that knows no political, racial, ethnic, or religious boundaries to meet, conduct research, send email, and play games. It's a revolutionary medium that is rewriting the script of human interaction.

Get an insider's view of Internet technology.

Preston Gralla is a long-time journalist with Ziff-Davis Publications and is Executive Editor for the ZDNet Software Library. While at *PC/Computing* magazine, he was instrumental in the development of the *How It Works* features on which the series is based.

- **Closely examine new technologies such as Java™, ActiveX™, Agents, and Web Sound**
- **See how Push technologies can broadcast information to your computer**
- **Learn how to use email, surf the Internet using a Web browser, and more**
- **Understand how RealAudio and Virtual Reality work on a Web page**
- **Discover how to make phone calls on the Internet**
- **Explore the growing world of Web commerce and learn how to purchase products securely through the Internet**

0 29236 71726 1

ISBN 0-7897-1726-3

92999

9 780789 717269

$29.99 USA / $42.95 CAN / £27.49 Net UK

que®

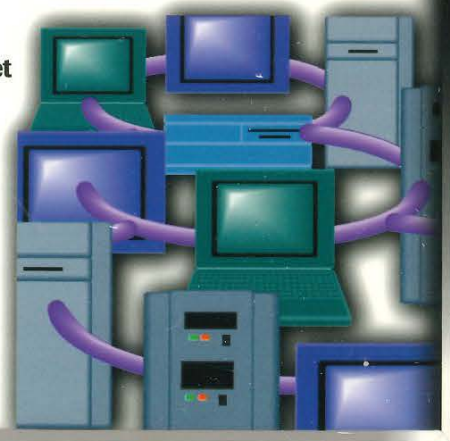
www.quecorp.com

COVERS: The Internet

CATEGORY: Internet Online/Communications

USER LEVEL: Beginning—Intermediate

Covering today's latest technology, How the Internet Works, Fourth Edition explains the Internet and the technologies that make it work.



494

07 45405 494

THRIFTBOOKS