

Application Layer

This chapter explores the different application programs, or services, available at the topmost layer, layer five, of the Internet model.

The application layer allows people to use the Internet. We could say that the other four layers are created so that people can use these application programs. This is the reason that we start with this layer, and, in Chapters 3–7, we move downward through the other four layers to see how they provide services to the application layer.

In this chapter, we introduce the application layer and its relationships to the other layers. We discuss the client-server paradigm, the only one used today in the Internet. We also discuss addressing, a feature inseparable from the application layer. We then explore the most common application programs used in the Internet including electronic mail, file transfer, remote logging, World Wide Web, list servers, videoconferencing, and chatting.

The list of application programs does not end here. We introduce some less obvious applications throughout the book. For example, we show how network management is handled by an application program in Chapter 16.

OBJECTIVES

After reading this chapter, the reader should be able to understand:

- The client-server paradigm
- Electronic mail
- File transfer
- Remote login
- The World Wide Web
- Other applications such as videoconferencing and chatting

2.1 INTRODUCTION

Figure 2.1 shows the position of the application layer in the Internet model. For the typical end user, this is the only layer of interest. This user only needs to know how to use the application programs available in this layer. They need to know how to send and receive email; they need to know how to use a browser to surf the World Wide Web. Some may need to use file transfer programs to send and receive bulky files.

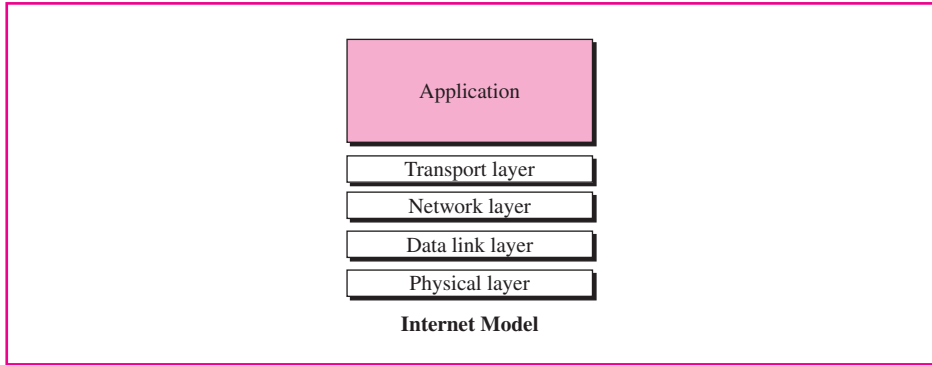


Figure 2.1 Position of application layer in the Internet model

A business manager, however, needs to be aware of the technologies available in each layer and how they can best be used. This knowledge enables a manager to make wise decisions concerning data communications and networking. This chapter, and Chapters 3–6, gives a manager insight into these layers.

2.2 CLIENT-SERVER PARADIGM

The application layer programs are based on the concept of clients and servers. The purpose of a network, and in particular, the global Internet, is to provide a service to a user. A user at a local site wants to receive a service from a computer at a remote site. For example, perhaps a user wants to retrieve a file from a remote computer. How can this be done? Both computers must run programs. The local computer runs a program that requests a service from another program on the remote computer. This means that both computers must run a program, one to request a service and one to provide a service.

At first glance, communication between two application programs—one running at the local site, the other running at the remote site—seems like a simple task. But many questions arise upon the actual implementation of the approach. Some questions that we may ask are:

- ❑ Should the application program be able to request as well as provide services or should the application program just do one or the other? One solution is to have an application program, called the **client**, running on the local machine, request a service from another application program, called the **server**, running on the remote machine. In other words, the tasks of requesting a service and providing a service are separate from each other. An application program is either a requester (a client) or a provider (a server). They come in pairs, client and server, both having the same name. Figure 2.2 illustrates this **client-server model**.

Application-layer programs in the Internet model communicate using the client-server paradigm.

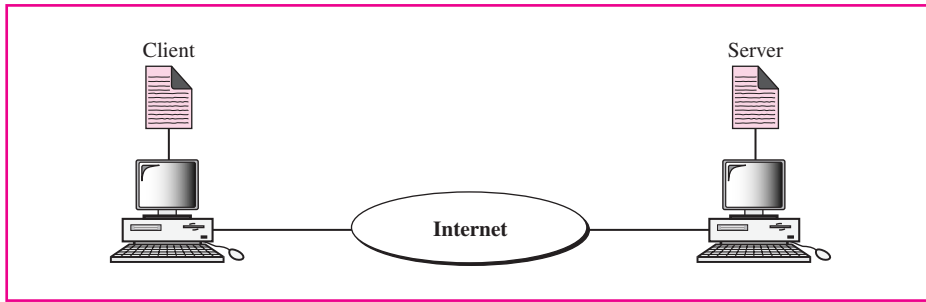


Figure 2.2 Client-server model

- ❑ Should an application program provide service for only one specific program installed elsewhere or should it provide service for any application program that requests this service? The most common solution is a server providing a service for any client, not just one particular client. In other words, the client-server relationship is many-to-one. Many clients can use the services of one server (see Figure 2.3).

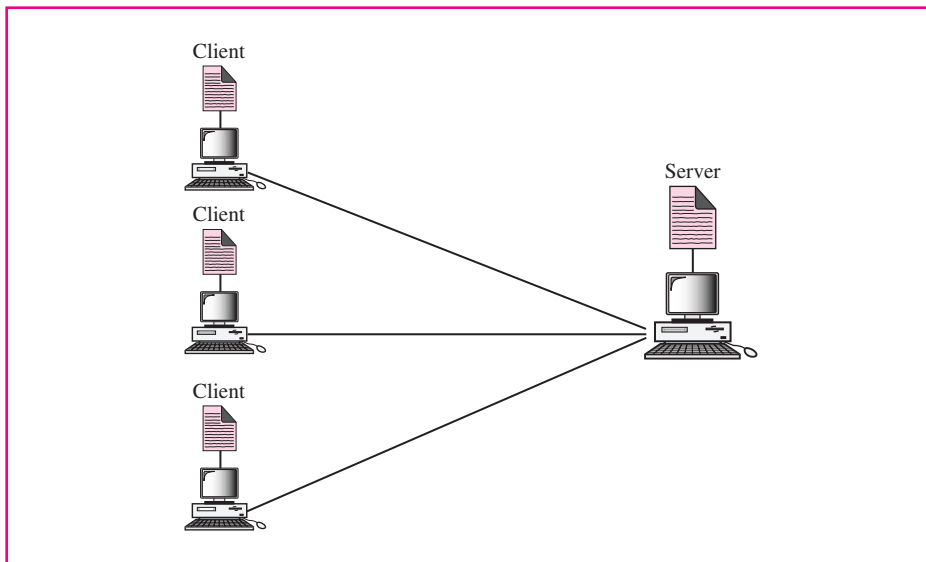


Figure 2.3 Client-server relationship

- ❑ When should an application program be running? All of the time or just when there is a need for the service? Generally, a client program, which requests a service, runs only when it is needed. The server program, which provides a service, runs all of the time because it does not know when its service will be needed.
- ❑ Should there be just one universal application program that can provide any type of service a user wants? Or should there be one application program for each type of service? Generally, a service needed frequently and by many users has a specific client-server application program. For example, one client-server application

program allows users to access files; a different client-server application program allows users to send email, and so on. For services that are more customized, we have one generic application program that allows users to access the services available on a remote computer. For example, there is a client-server application program that allows the user to log onto a remote computer and then use the services provided by that computer.

In the client-server paradigm, many clients can be served by one server, either one at a time or concurrently.

In the client-server paradigm, a client runs for a specific period of time; the server runs continuously.

In the client-server paradigm, there are specific servers and normally one general-purpose server.

CLIENT

A **client** is a program running on the local machine requesting service from a server. A client program is started by the user when the service is needed and terminates when the service is complete. A client opens the communication channel, sends its request, receives the response, and closes the channel. Multiple requests and responses are allowed in one session.

SERVER

A **server** is a program running on the remote machine providing service to the clients. When it starts, it opens the door for incoming requests from clients, but it never initiates a service until it is requested to do so. A server program normally runs infinitely unless a problem arises. It waits for incoming requests from clients. When a request arrives, it responds to the request.

2.3 ADDRESSING

A client and a server communicate with each other through addresses. When a client requests a service from a server, it must include the address of the server as the destination address, as well as its own address as the source address. The source address is required so that the server knows where to send the response. When the server responds to the request, it reverses the addresses; it uses its own address as the source and the address of the client as the destination.

Each application has its own address format. For example, an email address may look like *forouzan@fhda.edu*, while an address to access a web page may look like *http://www.fhda.edu*.

The Internet uses an address to identify an entity. Each computer connected to the Internet needs to be uniquely identified. To prevent duplication, a **hierarchical** or **tree name space** was proposed in which each name has several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this way, assignment and control of the name space can be decentralized.

Each node in the tree has a label. A full **domain name** is a sequence of labels separated by dots. For example, *challenger.atc.fhda.edu* is a domain name that defines a computer at De Anza College. Each label in the domain may define an entity in the organization; the level of detail increases from right to left.

The domain names are divided into two categories: generic domain names and country domain names. In the generic domain name, the rightmost label defines the type of activity. Traditionally, the Internet has defined seven labels: **com** (commerce), **edu** (education), **gov** (government), **int** (international), **net** (Internet), and **org** (non-profit organization). Recently, seven more labels were added to the list: **aero** (airlines and aerospace), **biz** (business and firms), **coop** (cooperative organizations), **info** (information service providers), **museum** (museums and other similar organizations), **name** (personal names), and **pro** (professional organizations).

2.4 ELECTRONIC MAIL

The earliest and the most common application of the Internet is **electronic mail** or email. Email is a store-and-forward application. This means a message can be sent to someone not currently connected to the Internet. The message can remain in the system until the recipient retrieves it.

Email is a store-and-forward application on the Internet.

The email application allows a user to send messages over a private network or the global Internet. Email supports:

- ❑ Sending a single message to one or more recipients.
- ❑ Sending messages that include text, voice, video, or graphics.
- ❑ Organization of message-based criteria such as priority.

EMAIL AND SNAIL MAIL

Email can be compared and contrasted with regular mail, which is often referred to as *snail mail*.

Advantages

Email has several advantages over snail mail:

- ❑ It is faster. In normal traffic hours, an email message can only take a few seconds or minutes to reach the destination if the recipient is in the same country or region;

it may take a few hours to reach an overseas recipient. Snail mail usually takes a couple of days, if not a week or two, to reach its destination.

- ❑ It is easier to distribute to a group of recipients. The sender creates a *distribution list* that contains the email addresses of all recipients. The list is given a group name. The email is sent using the group name as the virtual recipient. The email is delivered to all recipients on the list. In other words, a group name can be used instead of a single recipient. For example, if your clients are categorized according to geographical location, you might have group names such as `clients_east`, `clients_west` and so on. With snail mail separate envelopes must be sent to each individual in the group.
- ❑ It is less expensive. Sending an email is essentially free today if we ignore the cost of being connected to the Internet, which can be justified for other purposes. Sending letters via post office or other couriers is actually more costly than just the price of postage. To find the true cost of sending a letter, we need to add the cost of office supplies (paper, etc.), the cost of typing the letter, and the cost of delivering the letter to the post office.
- ❑ It can be less time-consuming. The first emails were friendly, informal messages. This tradition has found its way into business today. Email exchanged between organizations tend to be less formal than letters. This means less time and effort on both sides to accomplish the same task.

Disadvantages

We must not ignore some problems associated with the sending of email.

- ❑ An email cannot be certified. Although the sender can check to see if the receiver has received the mail, it cannot be used as legal proof. If we need a signature from the recipient, we still need to use the services of snail mail or some other courier service.
- ❑ The privacy of email is still an open question. Although several software packages are on the market to make email confidential, it cannot be guaranteed unless everyone uses one of these packages.
- ❑ Email messaging is subject to abuse. Unwanted and unsolicited email is a nuisance and can fill mailboxes much like junk mail. In addition, there is the threat of viruses and other potentially damaging code attached to email.

COMPONENTS

The sending of electronic mail in the Internet requires these components: user agents (UAs), mail transfer agents (MTAs), and the protocol that controls mail delivery—**Simple Mail Transfer Protocol (SMTP)**.

User Agent (UA)

The **user agent** is software installed on the user computer that reads, replies, forwards, saves, and composes messages. In the beginning of the electronic mail era, the popular user agents such as Berkeley Mail, Pine, and Elm were text-based. Today, user agents such as Eudora, Microsoft's Outlook, and Netscape's Messenger use a GUI (graphic user interface) to provide the users with a menu or window environment and allow the sending of text and multimedia.

A user agent controls the composing, reading, forwarding, replying, and saving of email messages. The user agent is not responsible for sending or receiving email.

Mail Transfer Agent (MTA)

The actual mail transfer requires a **mail transfer agent** (MTA). To send mail, a system must have a client MTA, and to receive mail, a system must have a server MTA. The client MTA is installed on the user's computer. The client and the server MTA are installed on a computer that is used as the mail server.

The task of sending and receiving email is done by a mail transfer agent (MTA).

Simple Mail Transfer Protocol

Simple mail transfer protocol (SMTP) is the protocol that defines the relationship between the UAs and MTAs, defines the format of the message to be transferred, and defines the type of the characters to be used.

The formal protocol for electronic mail in the Internet is Simple Mail Transfer Protocol or SMTP.

MAIL DELIVERY

Now that we know the components of electronic mail, let us show how mail is delivered from the sender to the receiver. The delivery of email from the sender to the receiver takes place in three stages (see Figure 2.4).

First Stage

In the first stage, the email goes from the user agent to the local server. The mail is stored here until the remote server is available. The user agent uses SMTP client software and the local server uses SMTP server software. The message is stored in a queue to be sent when the remote mail server is ready for receiving.

Second Stage

In the second stage, the email is relayed by the local server, which now acts as the SMTP client, to the remote server, which is now the SMTP server. The email is delivered to the remote server, not to the remote user agent. The reason is that SMTP messages must be received by a server that is always running since mail can arrive at any time. However, people often turn off their computers at the end of the day, and those with laptops or mobile computers do not normally have them on all the time. So usually an organization (or an ISP) assigns a computer to be the email server and runs the SMTP server program. The email is received by this mail server and stored in the mailbox of the user for later retrieval.

Third Stage

In the third stage, the remote user agent uses a mail access protocol such as POP3 or IMAP4 (both discussed in the next section) to access the mailbox and retrieve the mail.

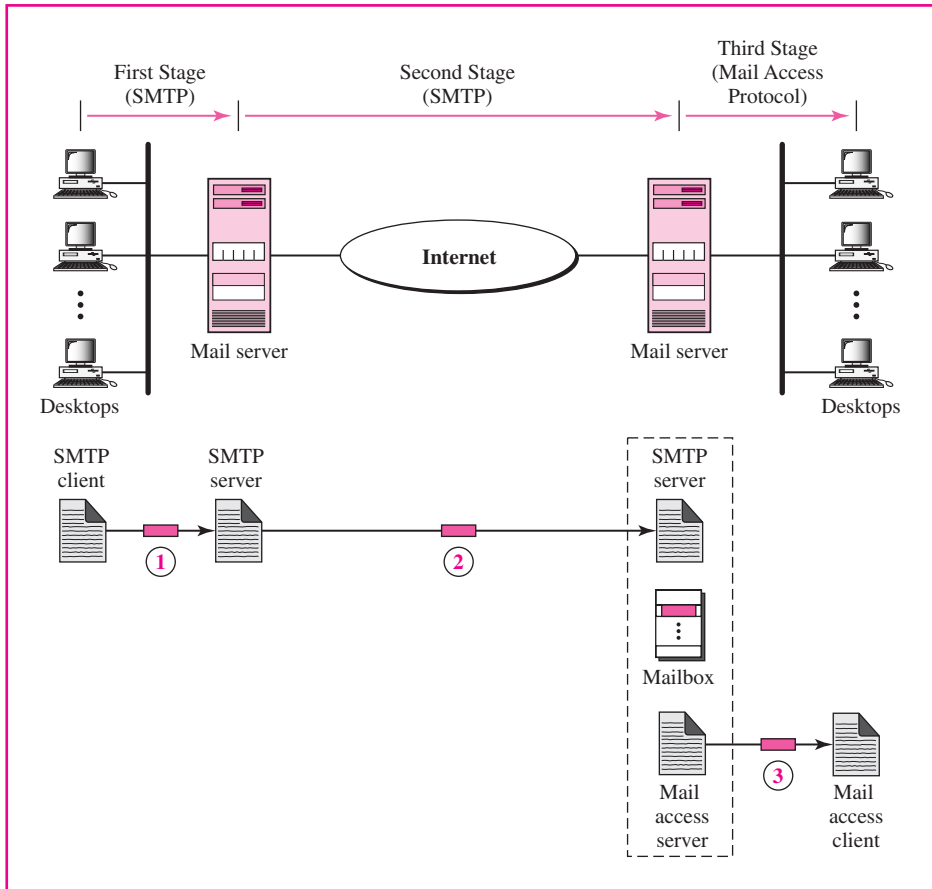


Figure 2.4 Email delivery

MAIL ACCESS PROTOCOLS

The stored mail remains in the mail server until it is retrieved by the recipient through an access protocol. Currently two mail access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol (IMAP).

POP

Post Office Protocol (POP) is simple but limited in functionality. The client POP software is installed on the recipient computer; the server POP software is installed on the mail server.

Mail access starts with the client when the user needs to download the received email from the mailbox on the mail server. The client (user agent) sends the user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one. Figure 2.5 shows an example of downloading using POP.

IMAP

POP has several deficiencies. It does not allow the user to organize mail on the server. The user cannot have different folders on the server. (Of course, the user can create

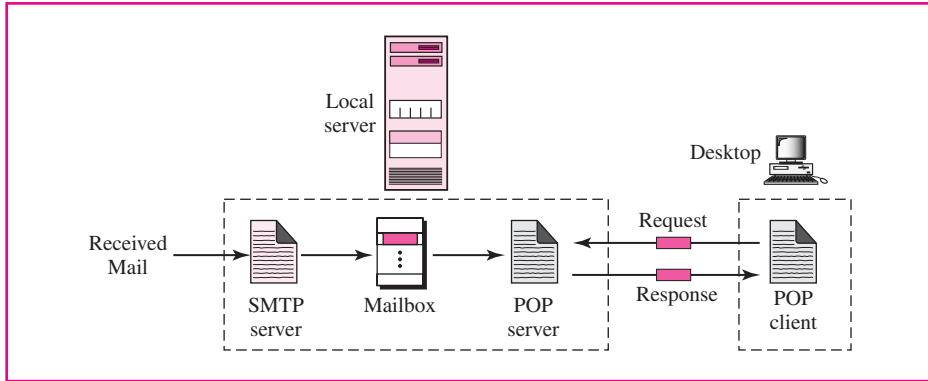


Figure 2.5 POP

folders on her own computer.) In addition, POP does not allow the user to check the contents of the mail before downloading.

A second popular mail access protocol, Internet Mail Access Protocol (IMAP), can handle these deficiencies. IMAP, though similar to POP, has more features and is more powerful and complex.

IMAP provides the following extra functions:

- ❑ A user can check the email header prior to downloading.
- ❑ A user can search the contents of the email prior to downloading.
- ❑ A user can partially download email. This is especially useful if the user has limited capacity and the email is long.
- ❑ A user can create, delete, or rename mailboxes on the mail server.
- ❑ A user can create a hierarchy of mailboxes in a folder for email storage.

The version of IMAP in use today is version 4.

ADDRESSES

A mail handling system must have a unique addressing system to deliver mail. The addressing system used by SMTP consists of two parts: a *local part* and a *domain name*, separated by an @ sign (see Figure 2.6).

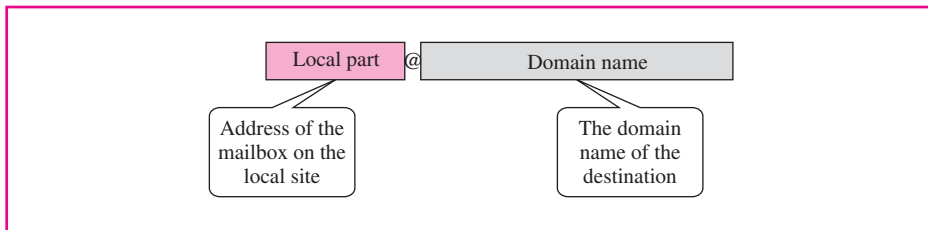


Figure 2.6 Email address

Local Part

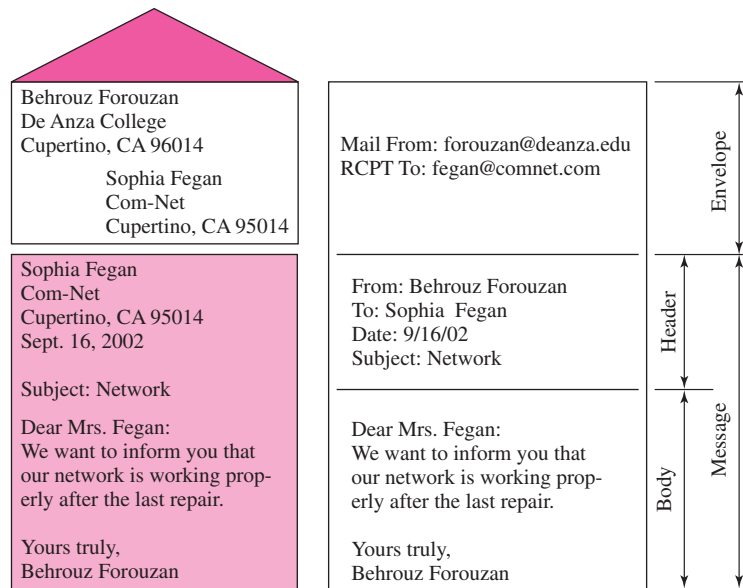
The local part defines the name of a special file, called the user mailbox, where all of the mail received for a user is stored for retrieval by the user agent.

Domain Name

The second part of the address is the name of a computer. An organization usually selects one or more hosts to receive and send email. The computer name assigned to each mail server, the **domain name**, comes from a universal naming system called the domain name system (DNS).

TECHNICAL FOCUS: INSIDE AN EMAIL MESSAGE

An email is made of an envelope and a letter. The letter itself is made of a header and a body as shown in the following figure:



MULTIPURPOSE INTERNET MAIL EXTENSION (MIME)

SMTP is a simple mail transfer protocol. Its simplicity, however, comes with a price. SMTP can send messages only in NVT 7-bit ASCII format. It cannot be used for languages that are not supported by 7-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send binary files or video or audio data.

Multipurpose Internet Mail Extension (MIME) is a supplementary protocol that allows non-ASCII data to be sent through SMTP. MIME is not a mail protocol and cannot replace SMTP; it is only an extension to SMTP.

MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client SMTP to be sent through the Internet. The server SMTP at the receiving side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original data.

We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa (see Figure 2.7).

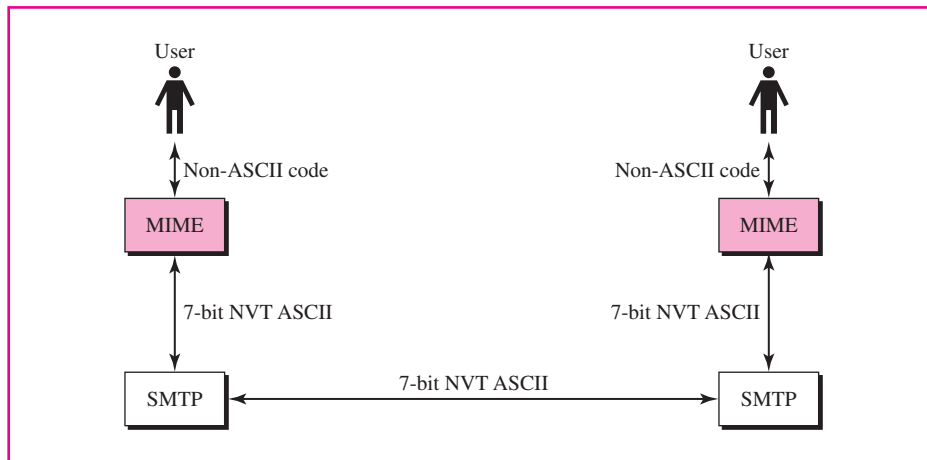


Figure 2.7 MIME

TECHNICAL FOCUS: HOW MIME WORKS

MIME has five header types that can be added to the original SMTP header section to define the transformation parameters:

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

Each header defines one feature of the transformation.

2.5 FILE TRANSFER

Another common user application is file transfer. We often need to electronically transfer files from one computer to another. Although email can be used for this purpose (as an attached document), there are some drawbacks. One problem is that the user who wants the remote file must first contact someone at the remote site to ask that the file be sent. This creates one extra communication and is inconvenient. The second problem is that the bandwidth assigned for email does not allow for quick transferral of a large file. Bandwidth is discussed in the future chapters; for the moment, think of bandwidth as the number of characters allowed on a connection. Transferring a large file needs a lot of bandwidth if we need it sent quickly.

Two applications are currently available for transferring files over the Internet: File Transfer Protocol (FTP) and Trivial File Transfer Protocol (TFTP).

FILE TRANSFER PROTOCOL (FTP)

File Transfer Protocol (FTP) is the standard mechanism for one of the most common tasks on the Internet, copying a file from one computer to another.

Although file transfer from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach.

FTP differs from other client-server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Figure 2.8 shows the basic model of FTP. The client has three components: the user interface, the client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes.

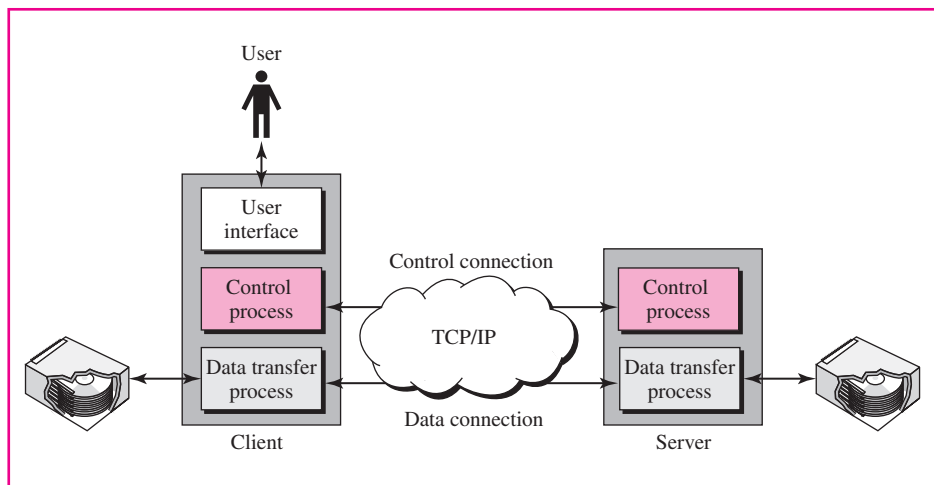


Figure 2.8 FTP

The control connection remains open during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes after the file is transferred. The two FTP connections, control and data, use different strategies and different port numbers.

Closed and Open FTP

The sites that run the FTP server to allow the user to access files on the site use one of two strategies. A site is either closed or open to the public. A closed FTP site allows

only specific users to access files. The access is controlled by the user account and password. The public is not allowed to access files at these sites.

An open FTP site allows the public to access files. These sites offer what is called **anonymous FTP**. The user can use *anonymous* as the user name and *guest* as the password. User participation in this system is very limited. Some sites allow anonymous users only a subset of commands. For example, most sites allow the user to copy some files, but do not allow navigation through the file system. We show an example of anonymous FTP. We connect to *internic.net* and we assume there are public data available for transfer.

```
% ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
bin
...
ftp > close
221 Goodbye
ftp > quit
```

TRIVIAL FILE TRANSFER PROTOCOL (TFTP)

There are occasions when we need to simply copy a file without the need for all of the functionalities of FTP. **Trivial File Transfer Protocol (TFTP)** is designed for these types of file transfer. It makes one single connection and transfers a small file quickly. However, this application is not universally available.

2.6 GENERAL-PURPOSE APPLICATION: TELNET

The main task of the application layer is to provide services for users. For example, users want to be able to run different application programs at a remote site and create results that can be transferred to their local site. One way to satisfy these demands is to create different client-server application programs for each desired service. Programs such as file transfer programs (FTP and TFTP), email (SMTP), and so on are already available. But it would be impossible to write a specific client-server program for each demand.

TELNET is a general-purpose client-server program that lets a user access any application program on a remote computer; in other words, it allows the user to log onto a remote computer. After logging on, a user can use the services available on the remote computer and transfer the results back to the local computer.

TELNET is an abbreviation for terminal network. TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

TELNET is a general-purpose client-server application program.

Local Login When a user logs onto a local time-sharing system, it is called **local login**. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility (see Figure 2.9).

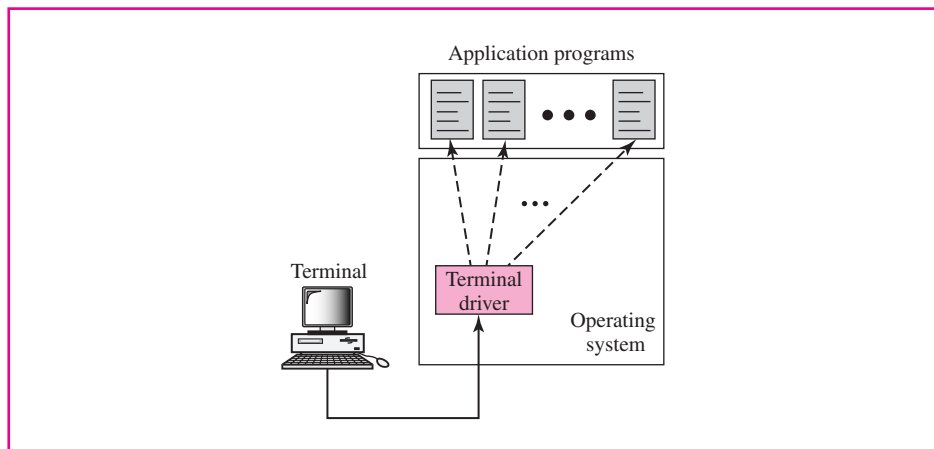


Figure 2.9 Local login

The mechanism, however, is not as simple as it seems because the operating system may assign special meanings to special characters. For example, in UNIX some combinations of characters have special meanings, such as the combination of the control character with the character *z* means suspend; the combination of the control character with the character *c* means abort; and so on. Whereas these special situations do not create any problem in local login because the terminal emulator and the terminal driver know the exact meaning of each character or combination of characters, they may create problems in remote login. Which process should interpret special characters? The client or the server? We will clarify this situation later in this section.

Remote Login When a user wants to access an application program or utility located on a remote machine, he or she performs remote login. Here the TELNET client and server programs come into use. The user sends the keystrokes to the terminal driver where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called *network virtual terminal characters* and delivers them to the local TCP/IP stack (see Figure 2.10).

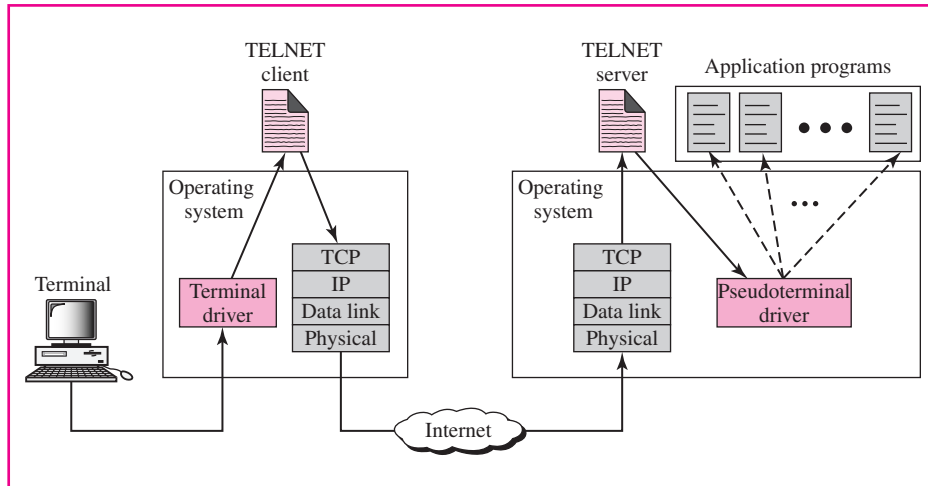


Figure 2.10 Remote login

2.7 WORLD WIDE WEB (WWW)

The World Wide Web (WWW), or the Web, is a repository of information spread all over the world and linked together. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

© Corbis/Vol. 147



The world wide web has revolutionized the way we access information, shop, conduct business.

The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *websites* (see Figure 2.11).

HYPertext AND HYPERMEDIA

The WWW uses the concept of hypertext and hypermedia. In a hypertext environment, information is stored in a set of documents that are linked together using the concept of pointers. An item can be associated with another document using a pointer. The reader who

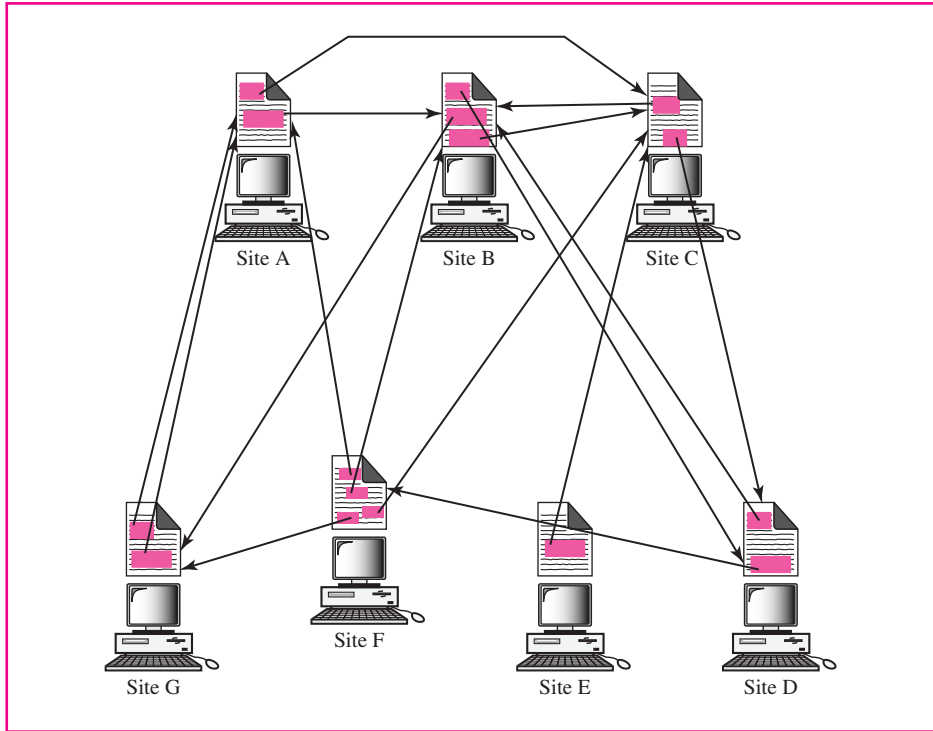


Figure 2.11 Distributed services

is browsing through the document can move to other documents by choosing (clicking) the items that are linked to other documents. Figure 2.12 shows the concept of hypertext.

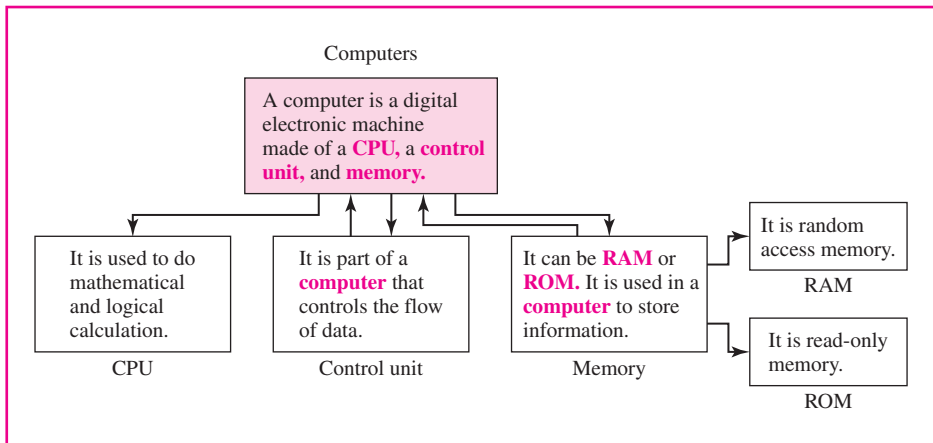


Figure 2.12 Hypertext

Whereas hypertext documents contain only text, hypermedia documents can contain pictures, graphics, and sound.

A unit of hypertext or hypermedia available on the Web is called a *page*. The main page for an organization or an individual is known as a *homepage*.

To use WWW, we need three components: a browser, a web server, and a protocol called the Hypertext Transfer Protocol (HTTP).

Browser

A variety of vendors offer commercial browsers that interpret and display a web document, and all of them use nearly the same architecture. Each browser usually consists of three parts: a controller, client programs, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The client program can be one of the protocols described previously such as FTP, or TELNET, but it is usually HTTP. The interpreter is a language used today on the Internet such as HTML or Java (see Figure 2.13).

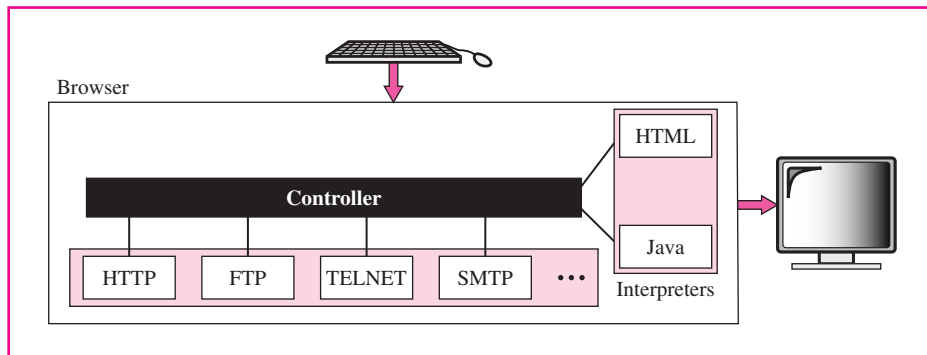


Figure 2.13 Browser architecture

Server

The server stores all pages belonging to the site.

HYPertext TRANSFER PROTOCOL (HTTP)

The **Hypertext Transfer Protocol** (HTTP) is a protocol used mainly to access data on the World Wide Web. The protocol transfers data in the form of plain text, hypertext, audio, video, and so on. It is called the Hypertext Transfer Protocol because its efficiency allows its use in a hypertext environment where there are rapid jumps from one document to another.

The idea of HTTP is very simple. A client sends a request, which looks like mail, to the server. The server sends the response, which looks like a mail reply, to the client. The request and response messages carry data in the form of a letter with MIME-like format.

The commands from the client to the server are embedded in a letterlike request message. The contents of the requested file or other information are embedded in a letterlike response message.

HTTP Transaction

Figure 2.14 illustrates an HTTP transaction between the client and server. The client initializes the transaction by sending a request message. The server replies by sending a response.

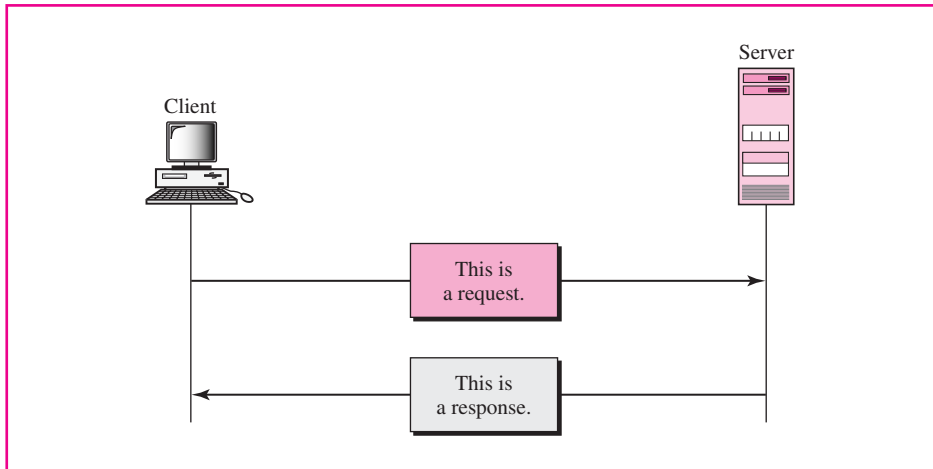


Figure 2.14 HTTP transaction

Messages

There are two general types of HTTP messages, shown in Figure 2.15: a request and a response. Both message types follow almost the same format.

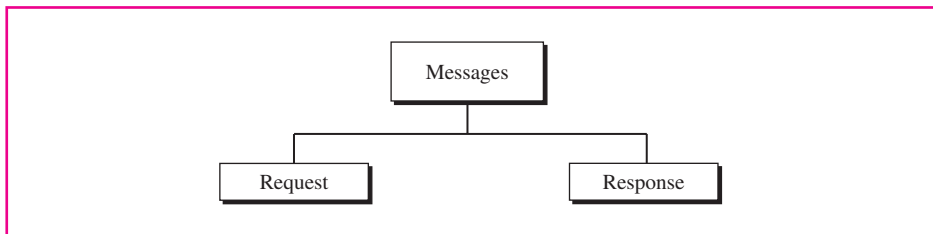


Figure 2.15 Message categories

Request Message A request message consists of a request line, headers, and sometimes a body (see Figure 2.16).

Response Message A response message consists of a status line, headers, and sometimes a body (see Figure 2.16).

ADDRESSES

A client that wants to access a document needs an address. To facilitate the access of documents distributed throughout the world, HTTP uses the concept of locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: method, host computer, port, and path (see Figure 2.17).

The *method* is the protocol used to retrieve the document, for example, HTTP. The *host* is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters “www.” This is not mandatory, however, as the host can be any name given to the computer that hosts the web page.

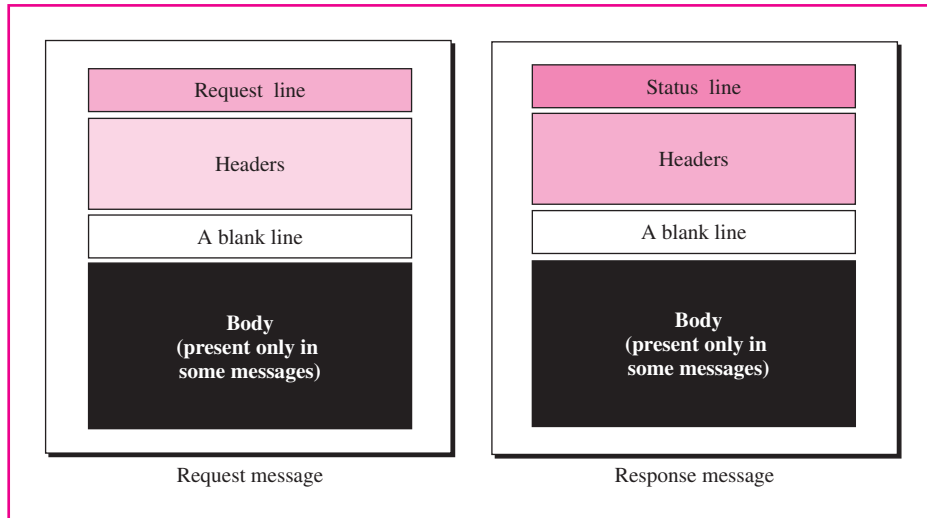


Figure 2.16 Request and response messages

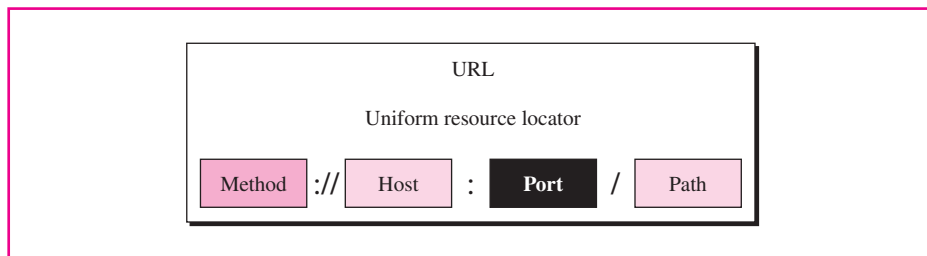


Figure 2.17 URL

The URL optionally can contain the port number of the server. If the *port* is included, it should be inserted between the host and the path, and it should be separated from the host by a colon.

The *path* is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

***BUSINESS FOCUS:* HISTORY OF WWW**

The idea of the World Wide Web started in 1989 at the European Particle Physics Laboratory, CERN, in Geneva, Switzerland. Tim Bernes-Lee needed to create a large database for physics research, which he found impossible to do using one single computer. The obvious solution was to let each piece of information be stored on an appropriate computer and let the computers be linked together through hypertext.

In 1993, the University of Illinois, under the supervision of Anderson, created the first graphical browser called Mosaic.

In 1994, Anderson and some colleagues started Netscape, one of the most popular browsers today. Another widely used browser is Microsoft Explorer.

TECHNICAL FOCUS: HTML

Hypertext Markup Language (HTML) is a language for creating web pages. The term *markup language* comes from the book publishing industry. Before a book is typeset and printed, a copy editor reads the manuscript and corrects it with marks. These marks tell the designer how to format the text. For example, if the copy editor wants part of a line to be printed in boldface, a wavy line is drawn under that part. In the same way, data for a web page are formatted for interpretation by a browser.

Let us explain the idea with an example. To make part of a text displayed in boldface with HTML, we must include the beginning and ending boldface tags (marks) in the text, as shown below:

```
<B> This is the text to be bold </B>
```

The two tags `` and `` are instructions for the browser. When the browser sees these two marks, it knows that the text must be boldfaced.

HTML lets us use ASCII characters (see Appendix A) for both the main text and formatting instructions. In this way, every computer can receive the whole document as an ASCII document. The main text is the data, and the formatting instructions can be used by the browser to format the data.

A tag is enclosed in two brackets (`<` and `>`) and usually comes in pairs. The beginning tag starts with the name of the tag, and the ending tag starts with a slash followed by the name of the tag.

2.8 VIDEOCONFERENCING

Another popular application is **videoconferencing**. Videoconferencing can eliminate the cost of traveling, and save time and energy. Videoconferencing can provide communication between two or more groups of participants or a set of individual participants (sometimes called desktop videoconferencing).

© Corbis/Vol. 147



Videoconferencing can eliminate the cost of traveling and save time and energy.

Each participant or group of participants uses a videoconferencing client program. The video and audio data are sent from the clients to the videoconferencing server program, which then distributes data to all clients as shown in Figure 2.18.

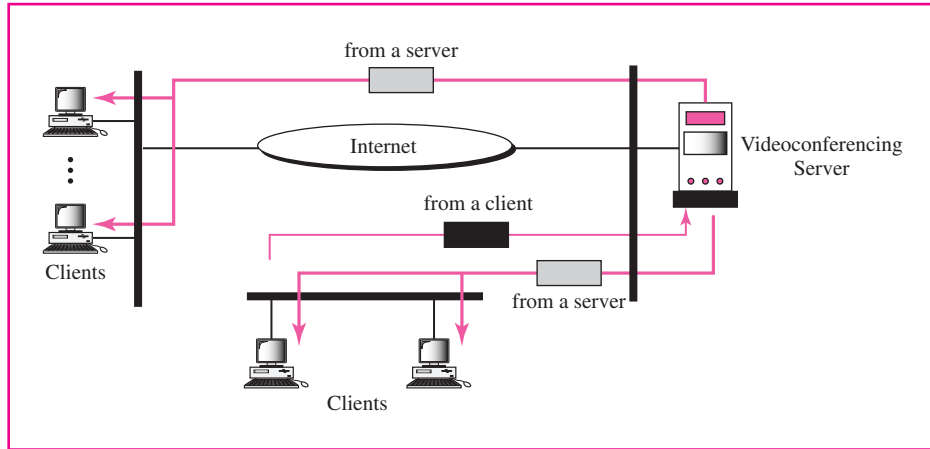


Figure 2.18 Videoconferencing

COMPRESSION

Video and audio communication requires the exchange of a large volume of data in a short period of time. As we know from the motion picture industry, a video is made of tens of picture frames sent each second to simulate a moving picture. This means that we send tens of picture frames per second with each picture frame containing thousands of picture elements (dots). Each picture element is encoded to represent the color of the dot. Video and audio data require the user to send a large amount of data to the network. In data communications parlance, this means that the user must be allocated a large bandwidth. The data road from the network to the user must be a freeway, not a one-lane street. To alleviate the situation, data can be compressed.

2.9 GROUP DISCUSSION: LISTSERV

Another popular application is **listserv**, which allows a group of users to discuss a common topic of interest. For example, there could be a group for those interested in seventeenth century Flemish tapestries or a group for San Francisco Seals fans. One popular use of listserv is the creation of discussion groups for students at colleges and universities, especially for those enrolled in distance learning classes. This is an excellent way for students unable to attend a traditional class to communicate with each other.

This application, like the others we have discussed, is also a client-server application. However, there are two server programs running on the server: the subscriber server and the mailer server. Figure 2.19 shows the situation.

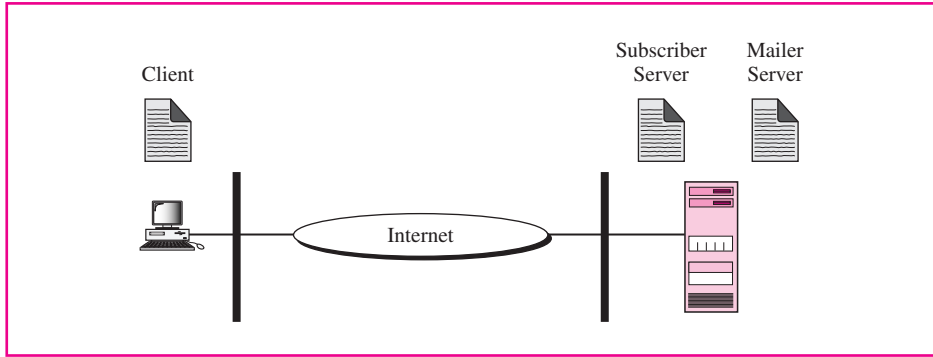


Figure 2.19 Listserv

SUBSCRIBING

The subscriber server accepts membership in a group. A user who wants to be a member of the group sends an email addressed to this server. The email contains the SUBSCRIBE command that is scanned by the subscriber server. The user is then registered if the user has permission to be part of the group. The subscriber server informs the user of the registration by a return email. A typical subscribe command is shown below:

```
SUBSCRIBE mailer-server-email-address user-email-address
```

To subscribe, the user sends the request to the subscriber server, not the mailer server.

SENDING EMAIL

After subscribing, a user can send email to the group. This time, the user sends an email addressed to the mailer server. The mailer server automatically relays the email to every member of the group if the email is from a registered user.

To send an email to every member of the group, the user sends it to the mailer server, which then automatically relays it to every subscriber.

UNSUBSCRIBING

To cancel a subscription, a user sends an UNSUBSCRIBE command to the subscriber server. A typical unsubscribe command is shown below:

```
UNSUBSCRIBE mailer-server-email-address user-email-address
```

The use of listserv in business is gaining in popularity. First, an organization can set up a group for each department for interdepartmental communication. Second, a group can be created between different organizations interested in a particular topic.

2.10 CHATTING

Another application in this layer is **chatting**. This is a real-time application like videoconferencing, in which two parties are involved in a real-time exchange of text, audio, and video. The two parties can send text to each other, talk to each other (the same way they talk on the phone), and even see each other.

Chatting works the same way as videoconferencing but on a smaller scale. The two parties each use a client program to send text, audio, or video information to a server. The server receives the information and relays the data, with a small delay.

Chatting has not found a place in the business world as yet. Currently, it is mostly used for personal purposes, but vendors hope that this will change.



CASE STUDY

NEW AGE HOME PHARMACY

NAHP uses standard application-layer programs: SMTP, POP, FTP, and HTTP. The most widely-used application program is electronic mail (email). An SMTP server program is installed on the main server computer in the basement. All incoming email are received by this server program and stored in mailboxes. Each employee has a mailbox (a file) in this computer. A copy of the SMTP client is installed on each desktop or laptop. The SMTP clients are responsible for sending mail. A copy of the POP client program is also installed on each user computer. The POP client periodically contacts the POP server, installed on the server in the basement, to retrieve any new email in the corresponding mailbox. The user interface, which is also installed on each user computer, is Eudora. This program creates, edits, formats, saves, and deletes email. It calls the POP client to get the received email and calls the SMTP client to send outgoing email. Because NAHP uses non-ASCII characters in email communication, the management has agreed to use MIME.

The addresses adopted for email communication have the domain name *nahp.com*, which is appended to the mailbox name. The mailbox name is the last name of the employee followed by the first name. For example, the email address of Madeline Lindsey is *lindseymadeline@nahp.com*.

Two years ago management hired a web page programmer to create a website for NAHP. The site, called *http://www.nahp.com*, advertises the services offered by NAHP to its target customer, the medical doctor. The website is maintained on the general server in the basement. The protocol used to retrieve web pages is HTTP.

NAHP also uses FTP. However, at present, this protocol is strictly for internal use. As we will see in the Chapter 14 case study, a firewall prevents FTP packets from leaving or entering the organization. FTP is used internally to transfer files (such as customer files) inside the organization.

2.11 KEY TERMS

anonymous FTP	network virtual terminal (NVT)
browser	Post Office Protocol, version 3 (POP3)
chatting	remote login
client	server
client-server model	Simple Mail Transfer Protocol (SMTP)
electronic mail (email)	Simple Network Management Protocol (SNMP)
File Transfer Protocol (FTP)	terminal network (TELNET)
Hypertext Markup Language (HTML)	Trivial File Transfer Protocol (TFTP)
Hypertext Transfer Protocol (HTTP)	uniform resource locator (URL)
Internet mail access protocol (IMAP)	user agent (UA)
listserv	videoconferencing
local login	World Wide Web (WWW)
mail transfer agent (MTA)	
multipurpose Internet mail extension (MIME)	

2.12 SUMMARY

- ❑ In the client-server model, the client runs a program to request a service and the server runs a program to provide the service. These two programs communicate with each other.
- ❑ One server program can provide services for many client programs.
- ❑ The server program is on at all times while the client program is run only when needed.
- ❑ The TCP/IP protocol that supports email on the Internet is called Simple Mail Transfer Protocol (SMTP).
- ❑ Both SMTP client and server require a user agent (UA) and a mail transfer agent (MTA).
- ❑ Post Office Protocol (POP) is a protocol used by a mail server in conjunction with SMTP to receive and hold mail for hosts.
- ❑ Multipurpose Internet mail extension (MIME) is an extension of SMTP that allows the transfer of multimedia messages.
- ❑ TELNET is a client-server application that allows a user to logon to a remote machine, giving the user access to the remote system.
- ❑ The World Wide Web (WWW) is a repository of information spread all over the world and linked together.
- ❑ Hypertext and hypermedia are documents linked to one another through the concept of pointers.
- ❑ The Hypertext Transfer Protocol (HTTP) is the main protocol used to access data on the World Wide Web (WWW).
- ❑ File Transfer Protocol (FTP) is a TCP/IP client-server application for copying files from one host to another.
- ❑ Trivial File Transfer Protocol (TFTP) is a simple file transfer protocol without the complexities and sophistication of FTP.
- ❑ Hypertext Markup Language (HTML) is a language used to create static web pages.

- ❑ Videoconferencing allows people at different sites to participate in a conference.
- ❑ Listserv is an application that allows a group of users to discuss a common topic of interest.
- ❑ Chatting is a real-time application in which two parties are involved in a real-time exchange of text, audio, and video.



PRACTICE SET

Review Questions

1. In the client-server model, what is the role of the client program? What is the role of the server program?
2. What application program allows connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system?
3. How is TFTP different from FTP?
4. What is the function of SMTP?
5. What is the difference between a user agent (UA) and a mail transfer agent (MTA)?
6. How does MIME enhance SMTP?
7. Why is an application such as POP needed for electronic messaging?
8. How are HTTP and the WWW related to the Internet?
9. What is the purpose of HTML?
10. Why should the server always be running whereas the client program should not?
11. Name three advantages of email over snail mail.
12. What is the purpose of an email server?
13. How is IMAP superior to POP?
14. What is the purpose of anonymous FTP?
15. Why is TELNET needed?
16. What are the components of the World Wide Web?
17. How is listserv different from chatting?

Multiple-Choice Questions

18. _____ can request a service.
 - a. A socket interface
 - b. A port
 - c. A client
 - d. A server
19. The UA is responsible for _____.
 - a. message preparation
 - b. message forwarding
 - c. sending messages across the Internet
 - d. a and b
20. The purpose of the MTA is _____.
 - a. message preparation
 - b. envelope creation
 - c. sending messages across the Internet
 - d. a and b

21. When a message is sent using SMTP, the local mail server has SMTP _____ software.
 - a. client
 - b. server
 - c. a and b
 - d. none of the above
22. MIME allows _____ data to be sent through SMTP.
 - a. audio
 - b. non-ASCII data
 - c. image
 - d. all of the above
23. Which of the following is a browser client program?
 - a. HTTP
 - b. FTP
 - c. TELNET
 - d. all of the above
24. Hypertext documents are linked through _____.
 - a. DNS
 - b. TELNET
 - c. pointers
 - d. homepages
25. What are the components of a browser?
 - a. retrieval method, host computer, pathname
 - b. controller, client program, interpreter
 - c. hypertext, hypermedia, HTML
 - d. all of the above
26. A server provides a _____ to a client.
 - a. service
 - b. paradigm
 - c. browser
 - d. none of the above
27. One server can service _____.
 - a. one client
 - b. two clients
 - c. many clients
 - d. one paradigm
28. A client runs _____; a server runs _____.
 - a. continuously; continuously
 - b. for a limited amount of time; for a limited amount of time
 - c. continuously; for a limited amount of time
 - d. for a limited amount of time; continuously
29. Email goes from the user agent to the _____.
 - a. local server
 - b. remote server
 - c. remote user agent
 - d. mailbox

30. _____ is a popular mail access protocol.
 - a. SMTP
 - b. POP
 - c. MIME
 - d. HTTP
31. IMAP has more sophisticated features for mail retrieval than _____.
 - a. SMTP
 - b. POP
 - c. MIME
 - d. HTTP
32. To transfer large files from one computer to another, _____ is the better choice.
 - a. HTTP
 - b. FTP
 - c. TFTP
 - d. SMTP
33. An open _____ site allows the public to access a limited set of files.
 - a. OSI
 - b. TELNET
 - c. HTTP
 - d. FTP
34. _____ is a general-purpose client-server program that lets a user access any application program on a remote computer.
 - a. OSI
 - b. TELNET
 - c. HTTP
 - d. FTP
35. A _____ interprets and displays a web document.
 - a. controller
 - b. port
 - c. MIME
 - d. browser
36. What does the URL need to access a document?
 - a. pathname
 - b. host computer
 - c. retrieval method
 - d. all of the above

