Chapter 11

Networked Applications

LEARNING OBJECTIVES

By the end of this chapter, you should be able to:

- Describe the concept and importance of networked applications.
- Describe how taking over an application can give an attacker the ability to control the computer.
- Describe electronic mail standards and security.
- Describe voice over IP (VoIP) operation and standards.
- Describe the World Wide Web in terms of standards and explain how a webpage with text, graphics, and other elements is downloaded.
- Explain peer-to-peer (P2P) computing including BitTorrent, Skype, SETI@home, and the Tor network.

GhostNet

After the Dali Lama fled Tibet in 1959, he set up a government in exile. Today it is located in Dharamsala, India. It is called the Office of His Holiness the Dalai Lama (OHHDL). In 2008, the staff suspected it was under cyberattacks when it sent an invitation to a certain person. Immediately afterward, this person reported that he received a call from a Chinese government official who discouraged him from accepting the invitation. In addition, a woman working for a group making Internet contacts between Tibetan exiles and Chinese citizens was stopped by Chinese intelligence officers on her way back to Tibet. She was shown transcripts of her online conversations and warned

to stop her political activities. These incidents led to an investigation that began to dissect a sophisticated cyberattack. This investigation was conducted by researchers at Cambridge University.[1]

The investigation found that the attack began with a spear phishing campaign that sent a legitimate-sounding message designed to entice staff members to open an attachment. Figure 11-1 shows one of these messages. It appears to be completely legitimate. However, when the monk opened the attachment, a Trojan horse was dropped on the monk's computer. This was a remote access Trojan (RAT) that allowed the attackers to remotely control the computer. (One monk reported seeing Outlook open by itself and send a message.) The Trojan could also do video and audio surveillance from the client computer, using the computer's camera and audio capability. Analysis showed that the malware searched computers for sensitive files and did keystroke logging. Using a modified version of HTTP, the attackers exfiltrated these files to Sichuan, China, which is where the Chinese intelligence group had been tasked with monitoring the OHHDL.

Using stolen credentials, the attackers then installed malware on the hosted OHHDL mail server in California. The malware frequently replaced attachments in legitimate e-mail with a malicious attachment. This allowed more and more client computers to be compromised with Trojan horses. Analysis of the mail server logs indicated successful logins from ISPs in China. No logins from China could have been legitimate.

```
Subject: Kalon Tripa Succession
From:    "Pema Rinzin" <prinzintibet@yahoo.com>
Date:    Thu, September 18, 2008 8:14 am
To:      choejor@dalailama.com
--------------------------------------------------------------------------

Dear Sir,

Attached please find the final Tibetan translation of my English
announcement for the Kalon Tripa succession initiative. Response to my
press release on September 2nd has been very positive and I have been
receiving lots of email and phone messages from Tibetans everywhere.

I am trying to get someone to translate the Kalon Tripa Hochoe into
English, but if you already have it translated, please send it to me.

Any advice from you in this initiative of mine would be greatly appreciated.

Yours sincerely,

Pema Rinzin
President
TAC




Official Photographer/webmaster
Office of His Holiness the Dalai Lama
Thekchen Choeling
P/O Mcleod ganj 176219
Dharamsala (H.P.)
India
```

FIGURE 11-1 Spear Phishing E-Mail with an Attachment

[1] Shishir Nagaraja, Ross Anderson, *The Snooping Dragon: Social-Malware Surveillance of the Tibetan Movement*, Technical Report Number 746, University of Cambridge Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom, March 2009.

While this investigation by Cambridge University researchers was underway, researchers from the University of Toronto[2] were finding that the attack on OHHDL was not an isolated incident. The techniques used against the Dali Lama, they discovered, were the tools of a large network of attack servers and other sites that the researchers named GhostNet. They were able to identify 1,295 infected computers in 103 countries, with a concentration in Southeast Asia. Almost a third of the victim organizations were high-value diplomatic, political, economic, or military targets.

Attackers are increasingly working through application programs and social engineering to implement their cyberattacks. Operating system exploits have been increasingly difficult to use with success, so attackers have largely turned to using application programs.

Test Your Understanding

1. a) How did the attackers gain their initial foothold on client PCs? b) What could the Trojan RAT do? c) Once they extended their control to the e-mail server, what did they do to get users to install additional RATs? d) What protocol did the attackers use to exfiltrate the files they discovered? e) What types of organizations were the most frequent victims of GhostNet?

INTRODUCTION

Networked Applications

Applications that require networks to operate are called **networked applications**. The World Wide Web and e-mail are networked applications. So is the Salesforce franchise management application used by Papa Murphy's.

Application Architectures In this chapter, we will focus on **application architectures**—that is, how application layer functions are spread among computers to deliver service to users.

- Early PCs used stand-alone operation in which all processing was done on the PC.
- Today, we have seen that client/server processing is dominant.
- We are now seeing the emergence of peer-to-peer (P2P) processing, in which user devices communicate directly, with little or no use of servers.

An application architecture describes how functions are spread among computers to deliver service to users.

Important Networked Applications In addition to looking broadly at application architectures, we will look at some of the most important of today's networked applications, including e-mail, voice over IP, the World Wide Web, cloud computing, peer-to-peer (P2P) computing, and mobile applications.

[2] Ronald J. Deibert and Rafal A. Rohozinski, "Tracking 'GhostNet': Investigating a Cyber Espionage Network," Report JR02-2009, Munk Center for International Studies, University of Toronto, Toronto, Canada, March 2009.

Networked Applications

- Applications that require networks to operate
- World Wide Web, e-mail, etc.

Application Architectures

- How application layer functions are spread among computers to deliver service to users
- Stand-alone operation
- Client/server operation
- Peer-to-peer (P2P) operation

Important Networked Applications

- E-mail, voice over IP, the World Wide Web, cloud computing, peer-to-peer (P2P) computing, and mobile applications

Importance of the application layer to users

- The only layer whose functionality users see directly
- What happens at lower layers should simply happen

FIGURE 11-2 Basic Networked Application Concepts (Study Figure)

Importance of the Application Layer to Users In this chapter, we will focus on the application layer. This is the only layer whose functionality users see directly. When users want e-mail, it is irrelevant what is happening below the application layer, unless there is a failure or performance problem at lower layers.

Test Your Understanding

2. a) What is a networked application? b) What is an application architecture? c) Why do users focus on the application layer?

The Evolution of Client Devices and Networking

Some years ago, the president of the Stanley Works told his Board of Directors, "Last year, we sold 4 million drill bits that nobody wanted." After pausing to let that provocative statement sink in, he went on to explain by saying, "What they wanted was holes." Drill bits and drills are expensive. They are merely tolerated, despite their expense and difficulty of use, because the customer needs holes. The message he was trying to emphasize is that drills are not the only ways to make holes. The company needed to focus on customer needs, not technology. Of course, although he was fundamentally correct, quite a few customers do like drills a lot. So technology can, to some extent, drive demand.

With networked applications, the same dynamic occurs. Obviously, applications must serve customer needs. However, as client devices and networks have evolved, each step in their evolution has brought new "killer apps" that were impossible before new devices and new networks appeared. Figure 11-3 shows important steps in the evolution of client devices.

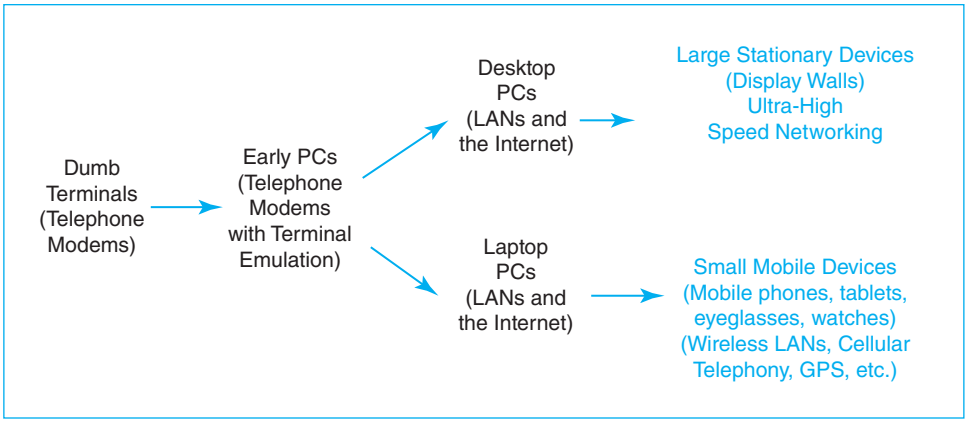

FIGURE 11-3 The Evolution of Client Devices and Networking

Dumb Terminals When computers were first built, the only way to interact with them was to throw binary switches, rewire boards, and place programs on punch cards into readers. Then, in the 1960s, remote terminals appeared. People could interact with computers via keyboards and printers (later keyboards and displays). This was before the days of microprocessors, so these devices were dumb terminals. They could merely send keystrokes to the large central computer and send characters from the central computer to the printer or display. Transmission used the telephone system. Dumb terminals were plugged into telephone modems, which sent their data over the telephone network. This limited speeds to around 10 kbps initially. At such low speed, the user interface on displays was very limited, consisted of monochrome text (plain text in a single color against a contrasting color background), which only required a few bytes to send per screen. In terms of application architecture, all processing had to be done on the large host computer.

Early PCs In the mid-1970s, the first personal computers appeared. Thanks to microprocessors, they were complete computers. These could be used by average employees in average companies and even homes. There was an explosion in new types of computer applications, including word processing, spreadsheet programs, graphics, and games. A few began to communicate to large computers by emulating (acting like) dumb terminals, but they were primarily stand-alone devices. They again used telephone modems, which again limited the user interface.

Desktop and Laptop PCs In the 1980s, a fundamental divide began between desktop PCs and laptop PCs. For desktop use, the key was processing power, massive amounts of storage, larger screens, and other performance-enhancing technologies. For laptop PCs, the keys were decreasing size and weight while gradually adding processing power. Both categories of PCs benefitted from the growth of local area networks in the 1990s.

In addition, while stand-alone applications were very popular, LANs created a new type of application architecture, client/server processing, which we have

seen since the beginning of this book. With PCs growing in performance, client/server processing split the processing work between the client and the server. Before PCs, this was not possible. In the 1990s, the Internet caused client/server processing to grow explosively. All computers soon came with browsers, which were initially universal clients for webserver programs. Browsers soon grew beyond this single application to be clients for the file transfer protocol, e-mail, and many other applications. With tremendous economies of scale, the cost of using the Internet fell rapidly. In addition, telephone modem access quickly gave way to broadband access services.

Small Mobile Devices The 21st century expanded the divide between desktop and smaller devices. New technology has allowed vendors to pack a substantial amount of processing power and battery life into small handheld products. These include tablets, smartphones, and new devices such as smart glasses and smart watches. Even laptops are becoming smaller, and they are getting touch screens as well.

Small device technology and wireless networking (including 802.11 WLANs, cellular telephony, and GPS) have combined to create a blizzard of new types of applications. Ever-growing transmission speeds have created rich applications impossible even on large desktop computers only a few years before.

Large Stationary Devices Small mobile devices are dominating the news on client evolution. However, working on a small screen can be difficult. Many office workers already have desktop PCs with multiple displays, offering a great deal of "real estate" for laying out windows for multiple applications. In some cases, these are already touch screens.

In the future, tables, desks, and walls will themselves be displays and will all be linked together by ultra-high-speed networks. Presumably, these devices too will create a revolution in applications. If history has taught us anything, it is that when new computer and networking technologies are in-vented, we always see a flood of startling new applications.

Test Your Understanding

3. Create a table. The first column should list the client device (dumb terminal, PC, etc.). The second column should list the technical advance embodied in the client. The third should be networking advances associated with the client. The fourth and last should be new networked applications made possible by the client and networking.
4. What advance made the client/server application architecture possible?

Application Security

In the past, hackers focused primarily on vulnerabilities in the operating system in order to break into computers. Today, however, hackers primarily attack individual applications running on the computer.

The reason for this is shown in Figure 11-4. If a hacker can take over an application, then he or she receives all of the permissions that the operating system gave to the

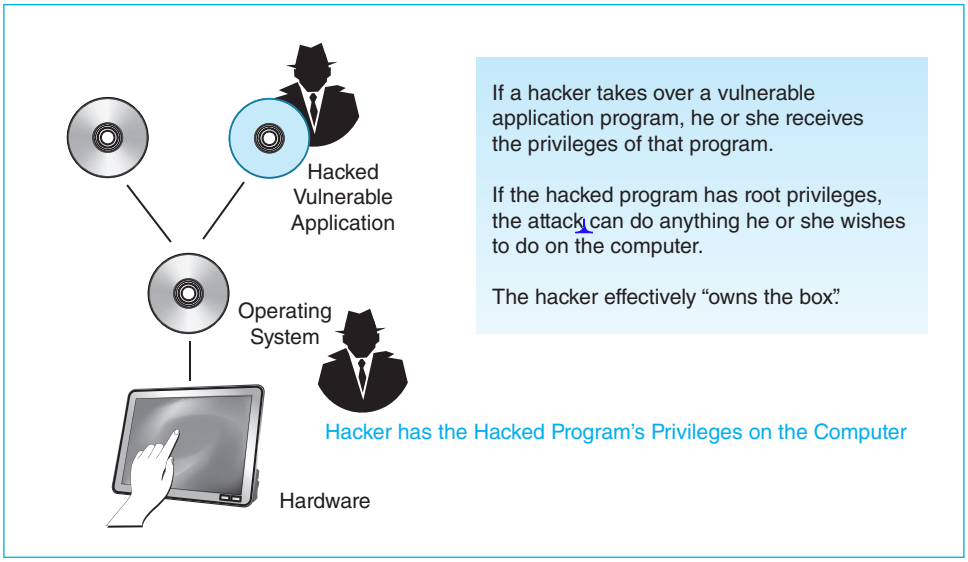

FIGURE 11-4 Application Hacking

application. Many applications run with **root privileges**, which means that they can do anything on the computer. Taking over such an application gives the hacker total control over the computer.

If a hacker can take over an application, then he or she receives all of the permissions that the operating system gave to the application.

Finding a vulnerability in the operating system is increasingly difficult. However, with the many applications running on most computers, and with inconsistent security quality across applications, the probability of finding a vulnerable application on a computer is high. Security vulnerabilities in specific applications are listed in many hacker forums that are readily available to attackers.

We are now seeing an explosion in apps created for mobile devices. In addition, we are seeing diversity in mobile operating systems. The newness of mobile operating systems and mobile applications has led many inexperienced developers to create applications with severe vulnerabilities. Coupled with a lack of corporate control over mobile devices, this lack of experience has created a flood of application (and operating system) vulnerabilities.

Test Your Understanding

5. a) Why are hackers now focusing on taking over applications? b) What can hackers do if they take over an application with root privileges? c) Why is the explosion of applications and small mobile devices a particular concern?

Cross-Site Scripting (XSS)

There are many ways to hack application programs. One popular attack vector is the **cross-site scripting (XSS)** attack. In these attacks, the user is asked for an input variable such as their name. The user may enter the name "Pat." The website then creates a webpage that contains something like "Hello Pat." This is called **reflection**. It is dangerous because the webpage will contain whatever the user chooses to give.

An attacker may be able to misuse sites that reflect user input. Figure 11-5 shows that an attacker has begun by sending the CEO of a corporation an e-mail message that purports to be from a subordinate. The message contains an apparently safe link to devour.com. Presumably, the company uses devour.com extensively, so the CEO sees the site as "safe."

In HTTP, the text that appears for a link may not be the true link. In Figure 11-5, the actual link is *http://www.devour.com/Default.aspx?name=<script>alert('Hacked!')</script>*. The link does take the victim to default.com. However, the problem is that it does more than that.

Most importantly, it will pass information to a particular program on Devour.com, Default.aspx. Default.aspx expects an input string for its name variable. Not shown in the figure, Default.aspx will reflect this name on a webpage. Probably, it will include something like "Hello *name*" on the webpage.

Given the e-mail message's crafted URL, however, the webpage being visited will reflect the script *<script>alert(Hacked!')</script>* on the webpage. When scripts are placed on a webpage, the user does not see them. However, the script executes when the page is rendered. This particular script is not too damaging. The user will see a pop-up alert box that contains the message "Hacked!"

Most XSS attacks have far more damaging scripts. For example, the script may steal the user's login cookie and send it to the attacker. This may give the attacker the

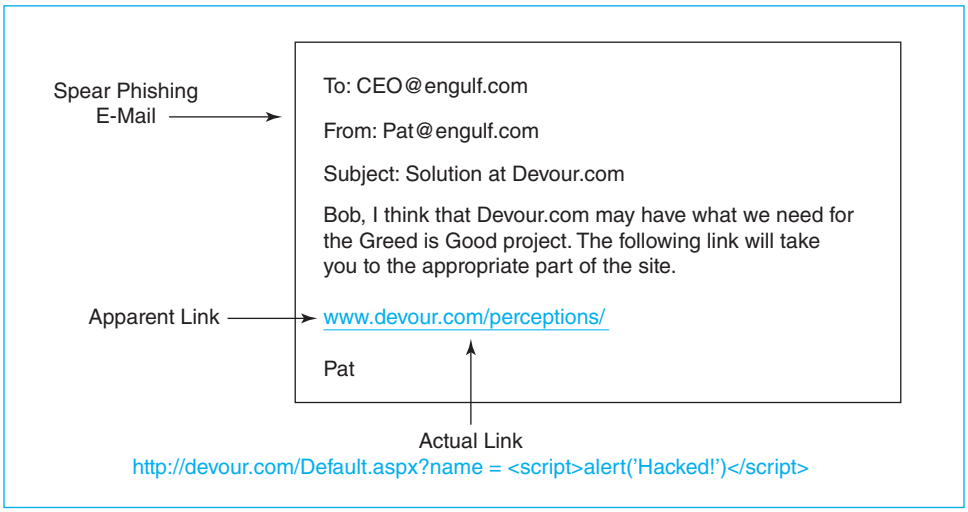

FIGURE 11-5 Cross-Site Scripting (XSS) E-Mail Message

victim's username and password. XSS attacks can also redirect the victim to another webpage, install malware, or even rewriting the contents of a webpage.

Cross-site scripting attacks do not always use e-mail or websites with deceptive links. For example, suppose a legitimate site allows user comments on webpages. Typically, the user enters text in a dialog box. The website then writes the comments onto the bottom of the webpage. If the comment contains a script, the script will execute every time someone visits the webpage afterward. This is a persistent XSS attack.

How can website designers thwart XSS attacks? At the broadest level, programmers should never trust user input. If information is to be reflected onto a webpage, the programmer must test the user input. It may seem simple to identify <script>and</script> tags, but scripts can be obfuscated (made less obvious). Also, there are many cross-site scripting attacks that do not use scripts. Thwarting cross-site scripting attacks is a difficult skill. This may explain why XSS vulnerabilities are pandemic on websites.

Programmers should never trust user input.

Test Your Understanding

6. a) Why is reflecting a user's input dangerous? b) What attitude should programmers have about user input?

SQL Injection Attacks

In many cases, user input becomes the basis for an SQL database query. If a malicious user knows this, he or she can enter specially crafted text to cause unanticipated damage. For example, suppose that a user enters a shipping destination in a dialog box. The website will compute the shipping cost. To do this, it will have to look up the shipping cost in a database using an SQL query.

To compute the shipping cost, the program may contain the following three lines of code. The first defines a new variable, destination. This is the destination city. The second gives the destination variable the destination name (inputdestination) the user has input into the form. The third creates an SQL query string. When this text string is entered into an SQL program, the program will find the destination city in the ShippingTable. It will return the value of the shippingcost column in that row. For instance, if the destination city is Tulsa, the program might tell the user that shipping the good will cost $20.

```
var destination;
destination = input.form ("inputdestination");
var sql = "select shippingcost from ShippingTable where destination = '" + inputdestination + "'";
```

However, if a user is malicious, he or she might enter the following information in the inputdestination field of the dialog box:

```
Tulsa'; drop table ShippingTable --
```

This string will be placed into the third line in the code. The "–" indicates that the rest of the line is a comment. This essentially "eats up" the rest of the line. The effect will be the following:

```
select shippingcost from ShippingTable where destination = 'Tulsa'; drop table ShippingTable
```

This will cause SQL to execute *two* statements instead of just one. First, it will tell the user the shipping cost to Tulsa. Second, it will delete the ShippingTable table. When the next person tries to look up the shipping cost for a particular city, they will only receive an error message.

Test Your Understanding

7. In an SQL injection attack, what does the user input instead of the expected input?

ELECTRONIC MAIL (E-MAIL)

Having discussed some factors that are driving new networked applications, we can now turn to specific applications. We will begin with electronic mail (e-mail) which was one of the earliest applications on wide area networks and that is still growing rapidly today.

E-Mail Standards

A major driving force behind the wide acceptance of Internet e-mail is standardization. Figure 11-6 shows that e-mail uses multiple standards for different aspects of its operation.

Message Body Standards

Obviously, message bodies have to be standardized, or we would not be able to read arriving messages. In physical mail, message body standards include the language the partners will use (English, etc.), the formality of language, and other matters.

RFC 2822 (Originally RFC 822) Bodies The initial standard for e-mail bodies (and headers) was **RFC 822**, which has been updated as **RFC 2822**. This is a standard for plain text messages—multiple lines of typewriter-like characters with no boldface, graphics, or other amenities. The extreme simplicity of this approach made it easy to create early client e-mail programs.

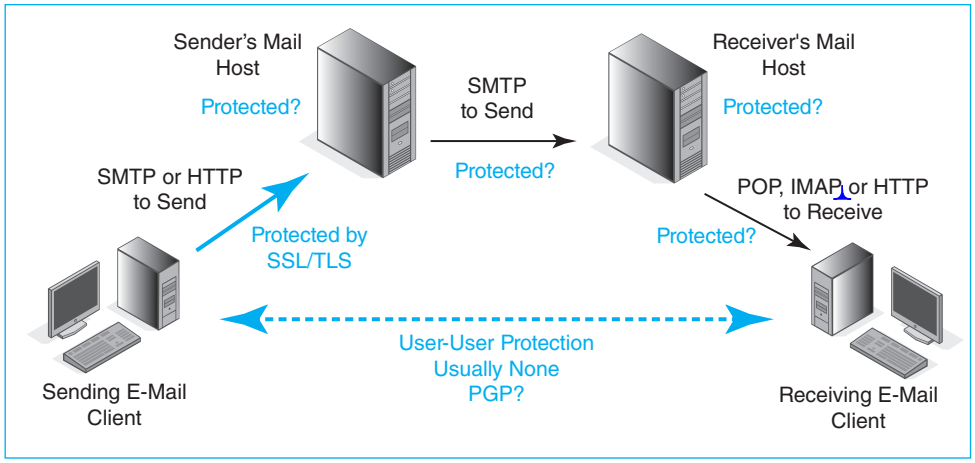

FIGURE 11-6 Classic E-Mail Standards

HTML Bodies Later, as HTML became widespread on the World Wide Web, most mail venders adopted **HTML bodies** with richly formatted text and even graphics.

UNICODE Bodies RFC 822 specified the use of the ASCII code to represent printable characters. Unfortunately, ASCII was developed for English, and even European languages need extra characters. The **UNICODE** standard allows characters of all languages to be represented, although most mail readers cannot display all UNICODE characters well yet.

Simple Mail Transfer Protocol (SMTP)

We also need standards for delivering RFC 2822, HTML, and UNICODE messages. In the postal world, we must have envelopes that present certain information in certain ways, and there are specific ways to post mail for delivery, including putting letters in post office drop boxes and taking them to the post office.

Figure 11-6 shows how e-mail is posted (sent). The e-mail program on the user's PC sends the message to its outgoing mail host, using the **Simple Mail Transfer Protocol (SMTP)**. Figure 11-7 shows the complex series of interactions that SMTP requires between the sender and the receiver before and after mail delivery.

Receiving Mail (POP and IMAP)

Figure 11-6 also shows two standards that are used to *receive* e-mail. These are the **Post Office Protocol (POP)** and the **Internet Message Access Protocol (IMAP)**. These standards allow the e-mail user to download new messages whenever they find convenient.

SMTP Process	Command	Explanation
Receiving	220 mail.panko.com Ready	When the sending host establishes a TCP session, the receiver signals that it is ready.
Sending	HELO voyager.shilder.hawaii.edu	Sender asks to begin sending a message. Identifies itself. (Yes, HELO, not HELLO)
Receiving	250 mail.panko.com	Receiver signals it is ready to receive a message.
Sending	MAIL FROM: david@voyager.hawaii.edu	Sender identifies the message author.
Receiving	250 OK	Receiver accepts the message author.
Sending	RCTP TO: ray@panko.com	Sender identifies the first recipient.
Receiving	250 OK	Receiver accepts the first recipient.
Sending	RCTP TO: lee@panko.com	Sender identifies the second recipient.
Receiving	550 No such user here	Receiver rejects the second recipient but will deliver the message to the first recipient.
Sending	DATA	Message will follow.
Receiving	354 Start mail input; end with <CRLF><CRLF>	Gives permission to begin sending the message.
Sending	*When in the course…*	Sender sends the message. Multiple lines of text. Ends with two carriage return/ line feeds, which gives a blank line.
Receiving	250 OK	Accepts the message.
Sending	QUIT	Sender requests termination of the SMTP session.
Receiving	221 Mail.Panko.COM Service closing transmission channel.	Receiver terminates session.

FIGURE 11-7 Simple Mail Transfer Protocol (SMTP)

Web-Enabled E-Mail

All client PCs have browsers. Many mail hosts are now Web-enabled, meaning that users only need browsers to interact with them in order to send, receive, and manage their e-mail. As Figure 11-8 shows, all interactions take place via HTTP. These systems use HTML to render pages on-screen.

SMTP for Transmission between Mail Hosts

So far, we have been looking at interactions between client hosts and their mail hosts. Figure 11-6 shows that mail hosts use SMTP when they transmit to each other. This is even true if client hosts communicate with their mail hosts using HTTP.

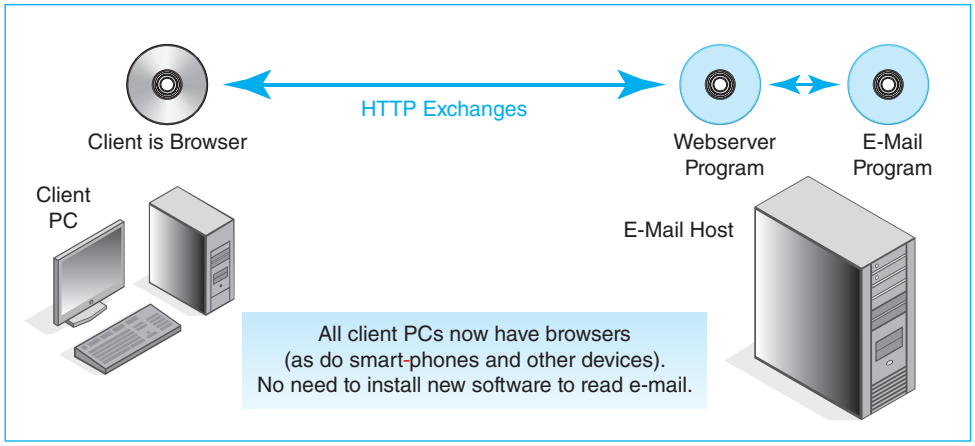

FIGURE 11-8 Web-Enabled E-Mail Operation

Test Your Understanding

8. a) Distinguish among the major standards for e-mail bodies. b) In traditional e-mail, when a station sends a message to its mail server, what standard does it use? c) When the sender's mail server sends the message to the receiver's mail server, what standard does it use? d) In traditional e-mail, when the receiver's e-mail client downloads new mail from its mail server, what standards is it likely to use? e) What is Web-enabled e-mail? f) What do you think are the advantages of a Web-enabled e-mail system? (The answer is not explicitly in the text.)

Malware Filtering in E-Mail

Although e-mail is tremendously important to corporations, it is a source of intense security headaches. As we learned in Chapter 3, the most widespread security compromises are attacks by malware. Malware enters an organization primarily, although by no means exclusively, through e-mail attachments and (sometimes) through scripts in e-mail bodies. E-mail attachments can also be used to install worms and Trojan horse programs on victim PCs.

The obvious countermeasure to e-mail-borne viruses is antivirus software, which scans incoming messages and attachments for viruses, worms, and Trojan horses. One problem is that most companies attempt to confront security threats by installing virus scanning on the user PCs. Unfortunately, too many users either turn off their antivirus programs if they seem to be interfering with other programs (or appear to slow things down too much) or keep their programs active but fail to update them regularly. In the latter case, newer viruses will not be recognized by the antivirus program.

Consequently, many companies are beginning to do central scanning for e-mail-borne viruses and Trojan horses. As Figure 11-9 shows, there are several places that this scanning can be done.

One popular place to do this is the corporate mail server. Users cannot turn off antivirus filtering on the mail server, and the e-mail staff (hopefully) updates virus definitions on these servers frequently.

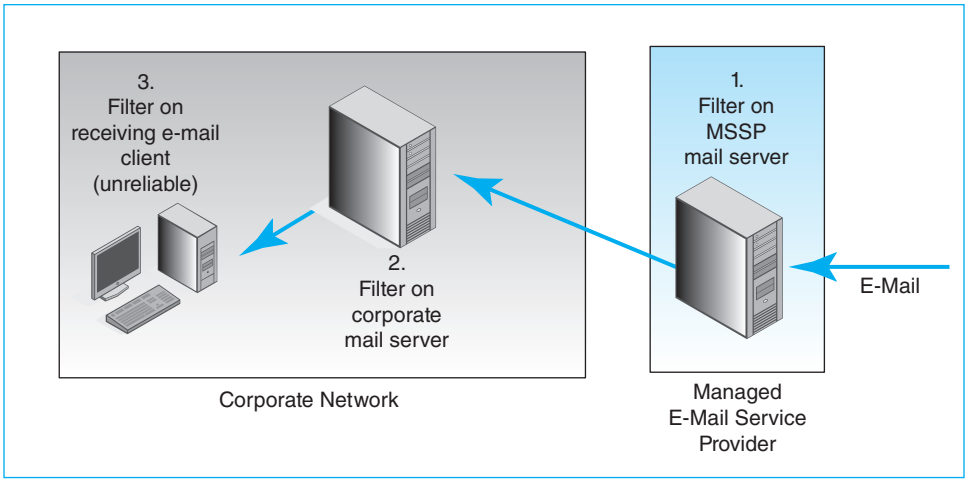

FIGURE 11-9 Possible Scanning Locations for E-Mail Malware

Some companies are even outsourcing antivirus/anti-Trojan horse scanning to outside security firms called managed e-mail service providers. To do this, they must redirect all incoming e-mail to a managed e-mail service provider. Fortunately, this is easy to do. Each second level domain's DNS server has an MX record that gives the IP address of the company's main mail server. The company simply changes the MX record to give the IP address of the managed e-mail service provider. The outsourcing firm has specialized expertise in searching for malware in e-mail. In addition, it handles the mail traffic of many firms. This gives it a large volume of e-mail, which allows it to analyze traffic for subtle malware trends.

Test Your Understanding

9. a) What is the main tool of firms in fighting viruses and Trojan horses in e-mail attachments? b) Why does filtering on the user's PC often not work? c) What options do firms have for where antivirus filtering may be done? d) According to the principle of defense in depth, how should firms do antivirus filtering?

Encryption for Confidentiality in E-Mail Transmission

Given the importance of e-mail and the sensitive information it often carries, you might expect that corporations would rigorously protect all e-mail through encryption for confidentiality. Actually, this is not often the case.

Link-by-Link Security Figure 11-10 shows a typical situation in e-mail security. The sending client is transmitting a message to its mail host. This transmission is protected by SSL/TLS. The user knows that the message is encrypted for confidentiality. This may cause the user to believe that there is end-to-end encryption for confidentiality with the other user.

However, SSL/TLS only protects communication between the sending client and the sender's mail host. SSL/TLS gives **link protection**, not an end-to-end protection. The link in this case is the transmission between the sending client and the sender's mail host.

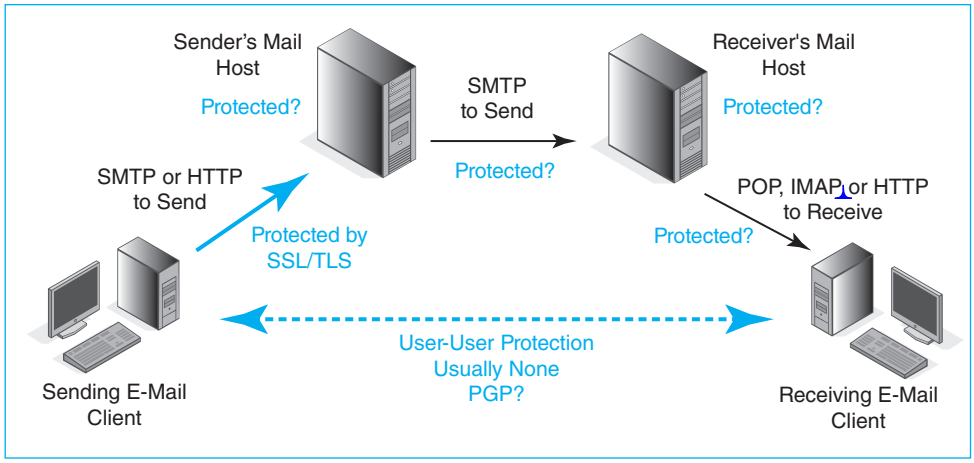

FIGURE 11-10 Encryption for Confidentiality in E-Mail

SSL/TLS provides link protection, not an end-to-end protection.

Is there any protection during the rest of the message's trip over the Internet? The answer is, "Perhaps." There may be encryption over the SMTP link between the sender's mail host and the receiver's mail host, but there is no guarantee. The receiving client may encrypt between itself and its mail host, but there is no guarantee of that either.

Even if encryption for confidentiality is implemented in all three links, the message is still not entirely safe. The sender's mail host must decrypt incoming messages, then re-encrypt the message to send it on to the receiver's mail host. For a brief period of time, the message will be in plaintext. A hacker who has planted spyware on the sender's or receiver's mail host will be able to capture the unencrypted message.

End-to-End Security The obvious remedy to the limits of link encryption is **end-to-end encryption** between the sending client and the receiving client. However, this is rarely done in practice. There is no widely accepted standard for end-to-end client encryption. The two clients must agree upon an encryption method if they wish to use one. Even if there is an end-to-end encryption method available, furthermore, users rarely use it.

There may even be legal issues with end-to-end encryption for confidentiality in a corporate environment. A number of laws require corporations to retain certain types of e-mail messages for later reading. For example, if an employee is fired, all electronic correspondence involved in the firing must be retained for a certain period of time. With end-to-end encryption, how would this be possible?

Test Your Understanding

10. a) If a message sender uses SSL/TLS when it sends a message, how is protection likely to be limited? b) Distinguish between link encryption and end-to-end encryption for confidentiality. c) Why is link-by-link encryption for confidentiality not fully secure even if there is encryption for confidentiality in all links along the

way? d) What is the remedy for the limitations of link-by-link encryption? e) Why is end-to-end encryption uncommon? f) Why may there be legal problems with end-to-end encryption?

VOICE OVER IP (~~VOIP~~)

Another example of the client/server architecture is **voice over IP (VoIP)**, which provides telephone conversations over IP networks and internets, instead of over the traditional telephone system. Like e-mail, VoIP is a client/server application in which both the sender and the receiver have their own servers. A major difference between these applications, however, is that~~,~~ after setting up a connection, the servers in VoIP get out of the way almost completely, and the two clients communicate by sending packets directly to each other until the end of the call.

Voice over IP (VoIP) provides telephone conversations over IP networks and internets, instead of over the traditional telephone system.

Basics

VoIP offers the promise of reducing telephone costs by moving from traditional telephone transmission, which reserves capacity for a call even when neither side was transmitting, to more efficient packet switching, which only charges for information actually sent. This can substantially reduce cost.

Clients Figure 11-11 illustrates VoIP operation. The figure shows two clients. One is a client PC with multimedia hardware (a microphone and speakers) and VoIP software. The other is a **VoIP telephone**, which has a **codec** (the electronics to encode voice for digital transmission) and the ability to send and receive packets over a TCP/IP internet. With VoIP, these two client users can talk with each other.

Media Gateway The figure also shows a media gateway. The **media gateway** connects a VoIP system to the ordinary public switched telephone network. Without a media gateway, VoIP users can talk only to one another, but they could not call people on landlines. The media gateway translates both signaling and transport communication across IP networks and traditional telephone networks.

The media gateway translates both signaling and transport transmissions.

Test Your Understanding

11. a) What is VoIP? b) What is the promise of VoIP? c) What devices can be used by VoIP callers? d) What is a codec? e) What is the purpose of a media gateway? f) Why is having a media gateway in a VoIP system important? g) Does the media gateway translate signaling transmissions or transport transmissions?

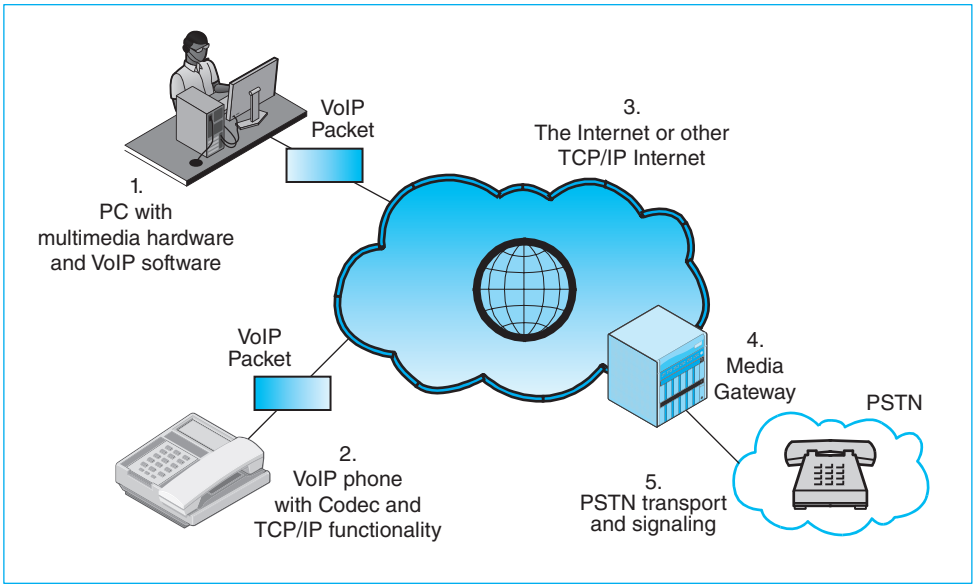

FIGURE 11-11 Voice over IP (VoIP) Operation

VoIP Signaling

In telecommunications, there is a fundamental distinction between signaling and transport. Signaling consists of the communication needed to set up circuits, tear down circuits, handle billing information, and do other supervisory chores. Transport is the actual carriage of voice.

In telecommunications, there is a fundamental distinction between signaling and transport. Signaling consists of the communication needed to set up circuits, tear down circuits, handle billing information, and do other supervisory chores. Transport is the actual carriage of voice.

There are two major VoIP signaling protocols. The first was the ISO **H.323** standard, which was effective but very complex. More recently, the IETF created the **Session Initiation Protocol (SIP)** standard. Most older VoIP systems use H.323 to control signaling. However, the use of SIP is growing rapidly, and most VoIP systems today use SIP for signaling.

Figure 11-12 illustrates the SIP protocol. Each subscriber has an SIP proxy server. The calling VoIP telephone sends a SIP INVITE message to its SIP proxy server. This message gives the IP address of the receiver. The caller's SIP proxy server then sends the SIP INVITE message to the called party's SIP proxy server. The called party's proxy server sends the SIP INVITE message to the called party's VoIP telephone or multimedia PC.

Test Your Understanding

12. a) What are the two major protocols for VoIP signaling? b) Which of these protocols is growing rapidly? c) Describe how SIP initiates a communication session.

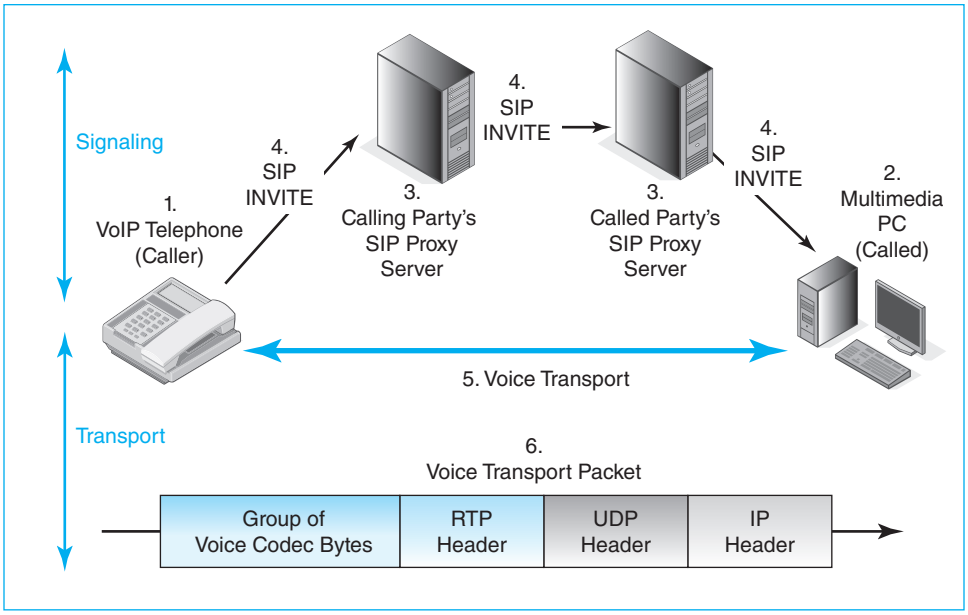

FIGURE 11-12 Voice over IP (VoIP) Signaling and Transmission Standards

VoIP Transport

After SIP or H.323 creates a connection, the two VoIP clients begin communicating directly. This is the beginning of transport, which is the transmission of voice between callers. VoIP, as its name suggests, operates over routed IP networks. Therefore, digitized voice has to be carried from the sender to the receiver in packets.

Codecs VoIP telephones and multimedia PCs need codecs to convert analog voice signals into digital voice data streams. VoIP systems can use many different codecs. Figure 11-13 shows that some codecs convert voice streams into bit streams as small as 5.3 kbps. However, the codecs that do the most compression also lose the most voice quality. Selecting codec in a VoIP network means making a trade-off between voice quality and cost reduction.

VoIP Transport Packets As noted in Chapter 1, long application messages have to be fragmented into smaller pieces that can be carried in individual packets. Each packet carries a small part of the application message.

Figure 11-12 shows a VoIP transport packet. The application message is a stream of voice codec bytes. Each packet carries a few bytes of the conversation.

TCP allows reliable application message delivery. However, the retransmission of lost or damaged TCP segments can take a second or two—far too long for voice conversations. Voice needs to be transmitted in real time. Consequently, VoIP transport uses UDP at the transport layer. UDP reduces the processing load on the VoIP telephones, and it also limits the high network traffic that VoIP generates. If packets are lost, the

Codec Standard	Bits Transmitted per Second
G.711	64 kbps
G.722	48, 56, or 64 kbps
G.721	32 kbps
G.722.1	24, 32 kbps
G.726	16, 24, 32, 40 kbps
G.728	16 kbps
G.729AB	8 kbps
G.723	5.33, 6.4 kbps
G.7231A	5.3, 6.3 kbps

FIGURE 11-13 Codec Standards

receiver creates fake noise for the lost codec bytes. It does this by extrapolating between the content of the preceding and following packets.

Between the UDP header and the application message, VoIP adds an additional header, a **Real Time Protocol (RTP)** header, to make up for two deficiencies of UDP:

- First, UDP does not guarantee that packets will be delivered in order. RTP adds a sequence number so that the application layer can put packets in the proper sequence.
- Second, VoIP is highly sensitive to jitter, which is variable latency in packet delivery. Jitter literally makes the voice sound jittery. RTP contains a time stamp for when its package of octets should be played relative to the octets in the previous packet. This allows the receiver to provide smooth playback.

Test Your Understanding

13. a) What is the purpose of a VoIP codec? b) Some codecs compress voice more. What do they give up in doing so? c) In a VoIP transport packet, what is the application message? d) Does a VoIP transport packet use UDP or TCP? Explain why. e) What two problems with UDP does RTP fix? f) List the headers and messages in a VoIP transport packet, beginning with the first packet header to arrive at the receiver. (Hint: See Figure 11-12.)

THE WORLD WIDE WEB

HTTP and HTML Standards

We have discussed the World Wide Web throughout this book. Figure 11-14 shows that the Web is based on two primary standards.

- First, webpages themselves are created using the Hypertext Markup Language (HTML).
- Second, the transfer of requests and responses uses the Hypertext Transfer Protocol (HTTP).

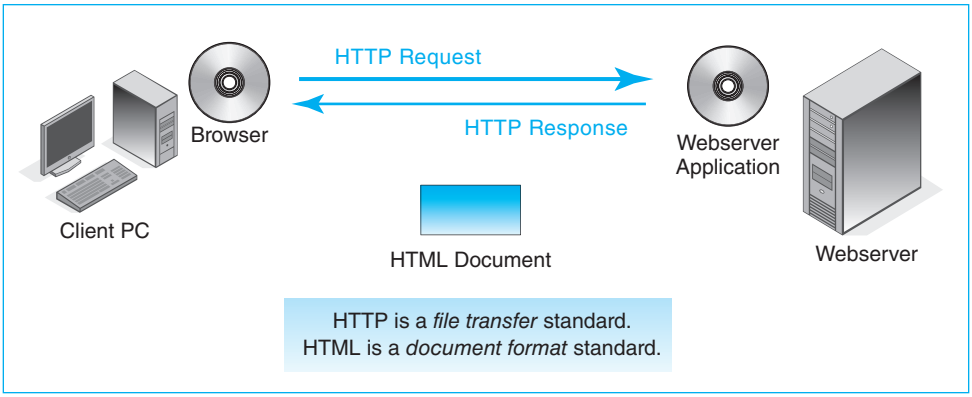

FIGURE 11-14 World Wide Web (WWW) Standards

To give an analogy, an e-mail message may be created using RFC 2822, but it will be delivered using SMTP. Many application standards consist of a document format standard and a file transfer standard.

Many application standards consist of a document format standard and a file transfer standard.

Complex Webpages

Actually, most "webpages" really consist of several files—a master text-only HTML file plus graphics files, audio files, and other types of files. Figure 11-15 illustrates the downloading of a webpage with two graphics files.

The HTML file consists merely of the page's text, plus **tags** to show where the browser should render graphics files, when it should play audio files, and so forth.[3] The HTML file is downloaded first because the browser needs the tags to know what other files should be downloaded.

Consequently, several HTTP request–response cycles may be needed to download a single webpage. Three request–response cycles are needed in the example shown in the figure.

Test Your Understanding

14. a) Distinguish between HTTP and HTML. b) You are downloading a webpage that has six graphics and two sound clips. How many request–response cycles will be needed?

[3] For graphics files, the IMG tag is used. The keyword *IMG* indicates that an image file is to be downloaded. The SRC parameter in this tag gives the target file's directory and file name on the webserver.

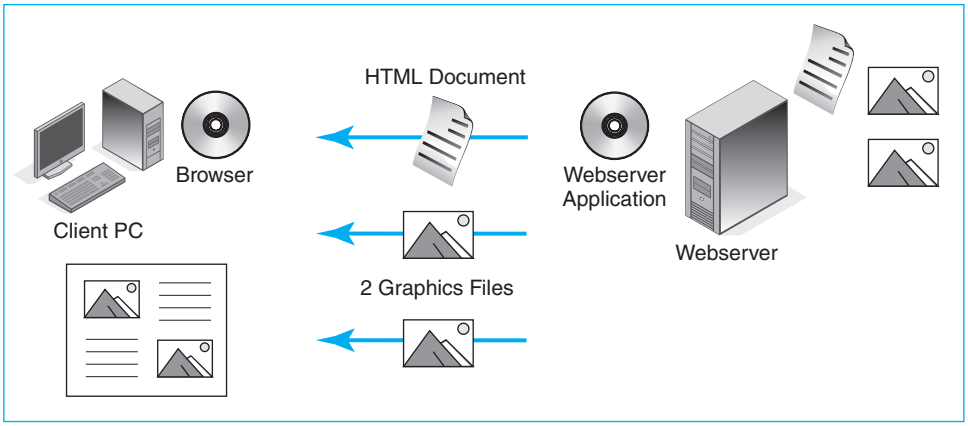

FIGURE 11-15 Downloading a Webpage with Two Graphics Files

PEER-TO-PEER (P2P) APPLICATION ARCHITECTURES

So far, we have examined two different application architectures: traditional terminal–host processing and client/server processing. Another application architecture is the **peer-to-peer (P2P) architecture**, in which most or all of the work is done by cooperating user computers, such as desktop PCs. If servers are present at all, they play only facilitating roles and do not control the processing.

In a peer-to-peer (P2P) architecture, most or all of the work is done by cooperating user computers, such as desktop PCs.

Traditional Client/Server Applications

Figure 11-16 shows a traditional client/server application. In this application, all of the clients communicate with the central server for their work.

Advantage: Central Control One advantage of this *server-centric* approach is central control. All communication goes through the central server, so there can be good security and policy-based control over communication.

Disadvantages Although the use of central service is good in several ways, it does give rise to two problems.

- One disadvantage is that client/server computing often uses expensive server capacity while leaving clients underused. Clients normally are modern PCs with considerable processing power, not dumb terminals or early low-powered PCs. Thus, power, storage, and bandwidth are all wasted in this model.

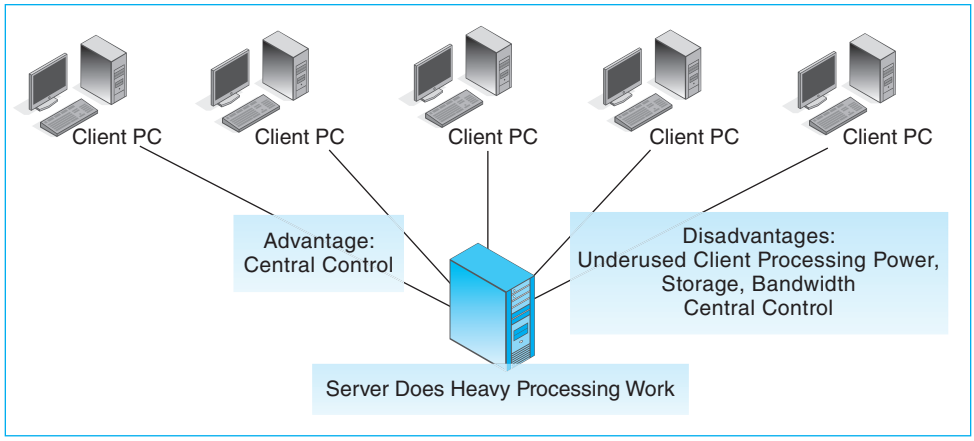

FIGURE 11-16 Traditional Client/Server Operation

- From the end users' point of view, central control can be a problem rather than an advantage. Central control limits what end users can do. Just as PCs freed end users from the red tape involved in using mainframe computers, peer-to-peer computing frees end users from the red tape involved in using a server. There is a fundamental clash of interests between central control and end user freedom.

P2P Applications

Figure 11-17 shows that, in a P2P application, user PCs work directly with one another, at least for part of their work. In this figure, all of the work involves P2P interactions. The two user computers work without the assistance of a central server and also without its control.

Advantages The benefits of P2P computing are the opposite of those of client/server computing. Client users are freed from central control, for better or worse, and less user computer capacity is wasted.

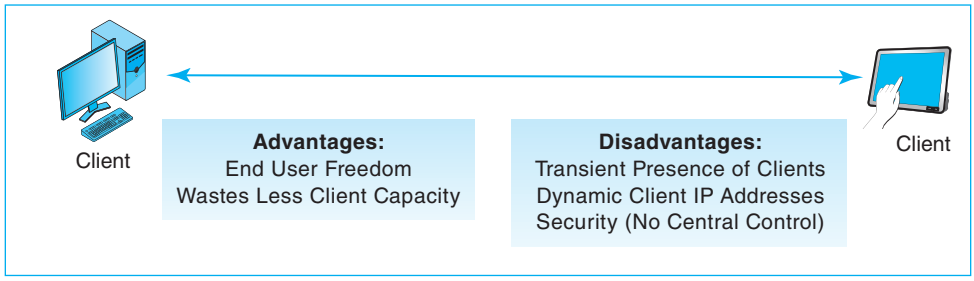

FIGURE 11-17 Simple Peer-to-Peer (P2P) Operations

Disadvantages P2P architectures have a number of unique disadvantages.

- *Transient Presence.* Most obviously, user PCs have transient presence on the Internet. They are frequently turned off, and even when they are on, users may be away from their machines. There is nothing in P2P like always-present servers.
- *Transient IP Addresses.* Another problem is that each time a user PC connects to the Internet, its DHCP server is likely to assign it a different IP address. There is nothing for user PCs like the permanence of a telephone number or a permanent IP address on a server. Dynamic IP addresses make finding PCs that provide service difficult.
- *Security.* Even if user freedom is a strong goal, there needs to be some kind of security. P2P computing is a great way to spread viruses and other illicit content. Without centralized filtering on servers, security will have to be implemented on all user PCs, or chaos will result.

Test Your Understanding

15. a) What are peer-to-peer (P2P) applications? b) What are the advantages of P2P applications compared to traditional server-centric client/server applications? c) What are the disadvantages?

P2P File-Sharing Applications: BitTorrent

One particularly popular type of P2P application is file sharing. In P2P file sharing, one client PC downloads a file that it needs from one or more other clients. Figure 11-18 shows **BitTorrent**, which is a popular peer-to-peer file sharing program.

Figure 11-18 shows the main steps in BitTorrent operation. First, your computer uses a **BitTorrent client program**. The BitTorrent client program searches for the file it wants, typically by going to an **index website**, which contains **.torrent** files giving information about specific files and where they are stored (Step 1). Next, the BitTorrent client program contacts a **tracker**, a server that coordinates the actual file transfer (Step 2). Trackers are usually run by independent clients, rather than being directly managed by BitTorrent.

To coordinate the file transfer, the tracker program examines all of the computers currently connected to its network to find out which have all or part of the file (Step 3). These computers are called the **swarm** for that particular file. The tracker passes this information to the BitTorrent client program, which begins to download different parts of the same file from multiple computers in the swarm (Step 4). These downloads occur simultaneously. The individual pieces are reassembled on the receiving computer to form the complete file (Step 5).

In order to encourage its users to share files, the BitTorrent system gives faster download speeds to users who opt to make their own files available for others to download. As more users share the same file, the download speeds for that file will become even higher, since a client program can take even smaller pieces of each file from each computer in the swarm.

Corporate Security Concerns Corporations that decide to use BitTorrent should consider several security concerns. First, BitTorrent uses specific port numbers, usually the TCP ports 6881 through 6889. Since firewalls commonly block these ports

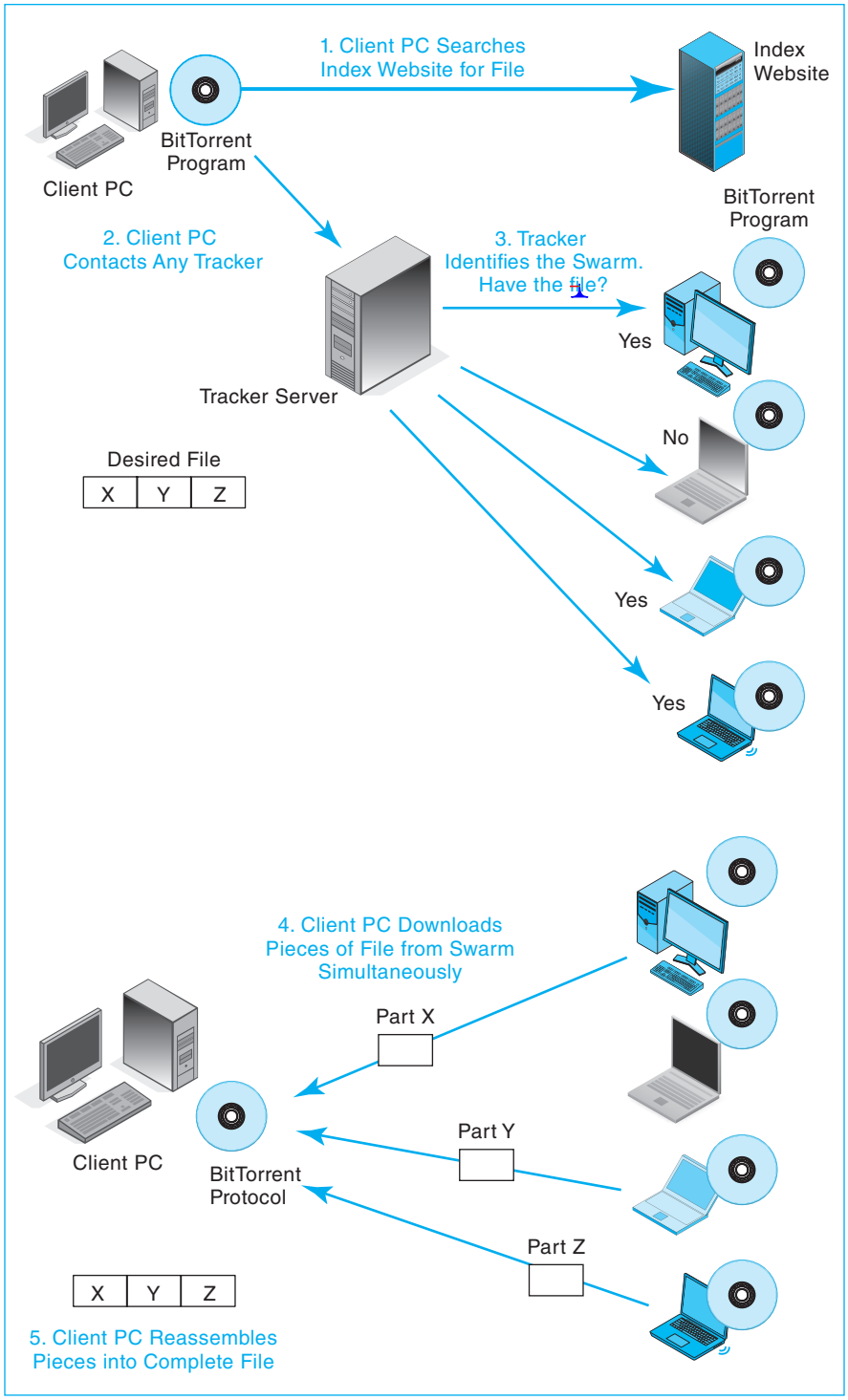

FIGURE 11-18 BitTorrent P2P File Retrieval

by default, using BitTorrent requires reconfiguring the firewall, possibly putting the firm at risk for attacks that exploit these ports.

Another security concern is that employees might use BitTorrent to download an infected file, which could then compromise the corporation's computers. Also, while using BitTorrent to share files is not itself illegal, one problem is that people have used the technology to share copyrighted material. A corporation must consider whether it will be held responsible if an employee uses BitTorrent to distribute illegal content.

Corporate Benefits Despite these potential problems, BitTorrent has started to see corporate use. The main advantage of using BitTorrent is that it allows the corporation to use clients (whose capacity is often underused) rather than expensive server processing power. This results in cost savings.

BitTorrent's efficient method of sharing files has been used by broadcasters like Canada's CBC and Norway's NPK to distribute their television programs. Video game developer and publisher Blizzard Entertainment has used the BitTorrent protocol to deliver updates and patches for its World of Warcraft game. The BitTorrent company has also released BitTorrent DNA, a content delivery product designed to aid corporations that want to use BitTorrent to handle large downloads and streaming video.

Test Your Understanding

16. a) Distinguish between client/server file retrieval and P2P file sharing. b) In BitTorrent, what is an index website? c) What are .torrent files? d) In BitTorrent, what is a tracker? e) In BitTorrent, what is a swarm? f) What security concerns must firms address if they plan to use BitTorrent? g) What is the main advantage of BitTorrent file sharing?

P2P Communication Applications: Skype

Another popular P2P application is Skype. While BitTorrent is used for file sharing, Skype is used for communication between people. Early in this chapter, we saw how voice over IP (VoIP) worked in the traditional client/server architecture. With Skype, we will see how VoIP works as a P2P application.

Skype is a P2P VoIP service that currently offers free calling among Skype customers over the Internet and reduced-cost calling to and from Public Switched Telephone Network customers. Skype offers a range of features, from phone calls to instant messaging and video calling. At the time of this writing, Skype is the most popular P2P VoIP service. Skype's free calls from computer to computer have greatly contributed to this popularity. Figure 11-19 illustrates how Skype operates.

There are three main elements in the Skype network: the Skype login server, host nodes, and super nodes.

- The **Skype login server** is a central server managed directly by Skype. It is the only centralized component in the Skype network.
- A **host node** is a Skype application that runs on a user's computer.
- A **super node** is a host node that takes on the work of signaling. Any regular host node may be made a super node if it has enough memory, network bandwidth, and CPU.

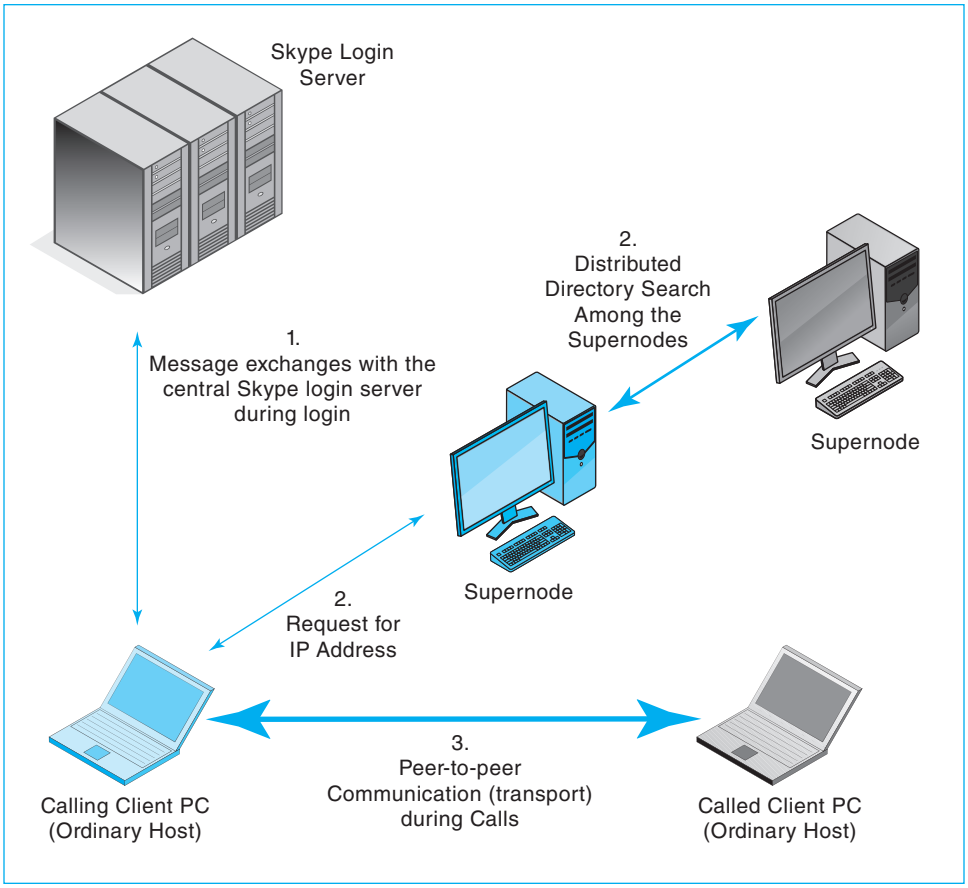

FIGURE 11-19 Skype P2P VoIP Operation

These elements are involved in the three steps that must occur for a user to place a call with Skype.

- *Step 1 Login.* First, a user must log in to the Skype login server. In this step, the username and password are authenticated. The Skype server also notes the user's IP address, which will be needed later, in the directory search process. Login is the only step that involves a central server; the rest of the call process is done peer-to-peer, using host nodes and super nodes. This step is similar to the login process in traditional voice over IP, where each client must log in to its own proxy server.
- *Step 2 Signaling/Directory Search.* After login, the user can place calls. His or her host will begin the signaling process. One of the main aspects of Skype signaling is the *directory search*, the process where a Skype application looks up the username and IP address of the party it wants to contact. A Skype directory search is a completely P2P process that is done using the super nodes. This is a major difference from traditional voice over IP, where signaling uses servers (proxy servers).

	Traditional VoIP	Skype	Comparison
Login	Server: user logs into his or her proxy server	Server: User logs into the Skype login server	Similar
Signaling	Server: proxy server manages signaling	Peer-to-Peer: Super nodes manage signaling, using P2P searching.	Major difference
Transport	P2P between the two hosts	P2P between the two hosts	Similar

FIGURE 11-20 Traditional VoIP and Skype

- *Step 3 Transport.* Figure 11-20 compares Skype with traditional VoIP. While Skype's super nodes handle signaling, transport is done entirely by the two host nodes involved in the call. In transport, the voice packets are routed completely P2P, from caller to called party and vice versa. This is similar to traditional voice over IP transport, where the two clients also communicate directly.

Because the signaling and transport are done by peers rather than going through a central server, Skype only carries the burden of managing a login server. This greatly reduces Skype's operational costs, resulting in its low-cost calls.

Test Your Understanding

17. a) What is Skype? b) Do you have to pay a fee to make calls using Skype? Explain. c) What is the most popular P2P VoIP service?

18. a) List and define Skype's three main elements. b) Explain how login works in Skype. c) What is a directory search in Skype? d) Which element of the Skype network is in charge of signaling? e) Which element of the Skype network is in charge of transport? f) Which of Skype's three steps is done P2P? g) Compare Skype and traditional voice over IP in terms of whether login, signaling, and transport are P2P or whether they use servers.

P2P Processing Applications: SETI@Home

As noted earlier, most PC processors sit idle most of the time. This is even true much of the time when a person is working at his or her keyboard. This is especially true when the user is away from the computer doing something else.

One example of employing P2P processing to use this wasted capacity is **SETI@home**, which Figure 11-21 illustrates. SETI is the Search for Extraterrestrial Intelligence project. Many volunteers download SETI@home screen savers that really are programs. When the computer is idle, the screen saver awakens, asks the SETI@home server for work to do, and then does the work of processing data. Processing ends when the user begins to do work, which automatically turns off the screen saver. This approach allows SETI to harness the processing power of millions of PCs to do its work. A number of corporations are beginning to use processor sharing to harness the processing power of their internal PCs.

Test Your Understanding

19. How does SETI@home make use of idle capacity on home PCs?

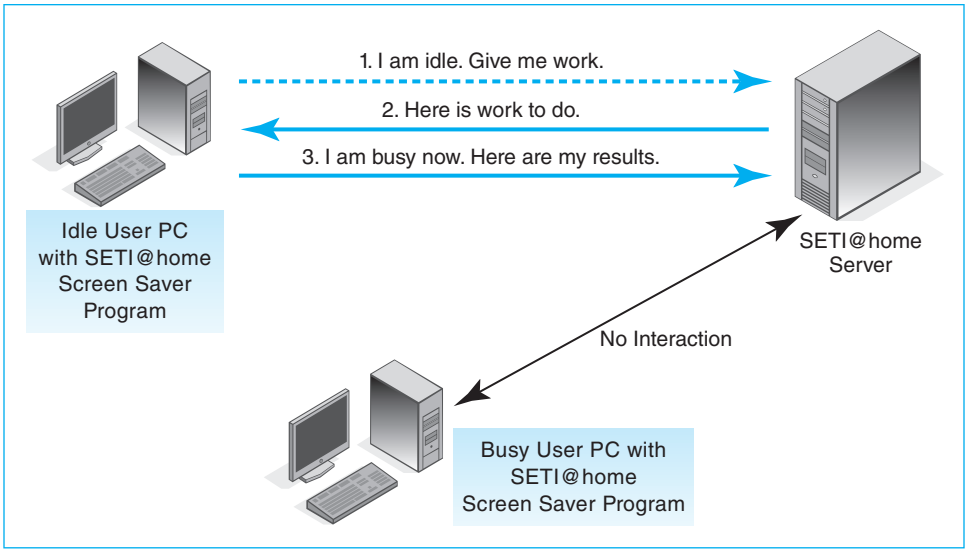

FIGURE 11-21 SETI@home P2P Application Processing

Privacy Protection: Tor

Even before we learned of the National Security Agency's eavesdropping on American Internet traffic, some people wished to use the Internet anonymously to preserve their privacy. This is difficult because IP packets have source IP addresses that identify the sender.

The **Tor** project was created to allow anonymous Internet use. In Figure 11-22, Client A wishes to connect anonymously with Server B. Client A knows that many P2P devices acting as Tor routers (relays) offer anonymous Internet transmission. These Tor routers are numbered in the figure. Collectively, they constitute the **Tor network**.

Client A negotiates a specific path through the Tor network using the **Tor protocol**. Client A then creates a message for Server B. The Tor path will go through three Tor routers. Client A will encrypt the message three times. Each Tor router decrypts the outermost layer of the cryptographic "onion" and passes the semi-decrypted message to the next Tor router on the path.

The final Tor router is an *exit node*. When it finishes its decryption, it has the original plaintext message from Client A to Server B. It passes this decrypted message to the server. The exit node is a concern because it can read the message. However, it does not know the IP address of Client A. It only knows the IP address of the previous Tor router. This provides a good degree of anonymity to Client A. However, if the exit node is malicious, it may be able to analyze the messages for an indication of Client A's identity.

Test Your Understanding

20. a) What is the goal of the Tor network? b) How does Tor use P2P devices? c) Does Tor provide confidentiality? (This is not a simple question.)

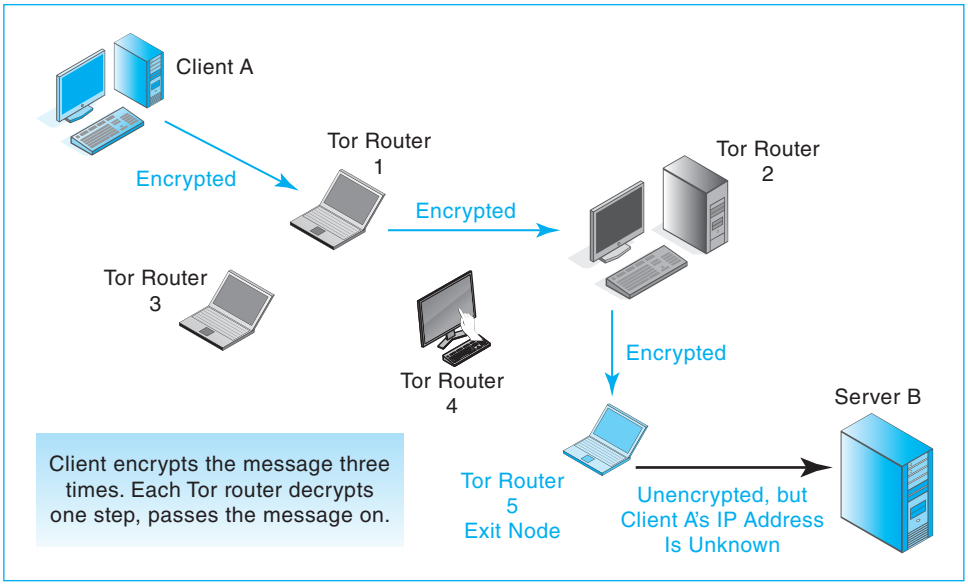

FIGURE 11-22 Tor Anonymity

Facilitating Servers and P2P Applications

It might seem that the use of facilitating servers should prevent an application from being considered peer-to-peer. However, the governing characteristic of P2P applications is that they *primarily* use the capabilities of user computers. Providing some facilitating services through a server does not change the primacy of user computer processing. For example, Skype is still considered a P2P application despite its use of a login server. Similarly, SETI is considered P2P even though it uses a server that downloads to and accepts data from the SETI@home users.

Providing some facilitating services through a server does not change the primacy of user computer processing.

Test Your Understanding

21. Explain how P2P applications may use facilitating servers yet still be called P2P applications.

CONCLUSION

Synopsis

Networked applications are applications that require a network to function. Although IT professionals must understand all layers in networking, users are only concerned with applications. Applications are also critical in terms of security. If an attacker takes

over an application, he or she receives all of the application's permissions. In some cases, taking over a single application can allow a hacker to control the computer.

E-mail is extremely important for corporate communication. Thanks to attachments, e-mail also is a general file delivery system. In operation, both the sender and the receiver have mail servers. Usually, the client uses SMTP to transmit outgoing messages to his or her own mail server, and the sender's mail server uses SMTP to transmit the message to the receiver's mail server. The receiver usually downloads mail to his or her client PC by using POP or IMAP. With Web-enabled mail service, however, senders and receivers use HTTP to communicate with a webserver interface to their mail servers. Transmissions between mail servers still use SMTP.

Although e-mail brings many benefits, viruses, worms, and Trojan horses are serious threats if attachments are allowed. Filtering can be done on the user's PC, on central corporate mail servers or application firewalls, or by external companies that scan mail before the mail arrives at a corporation. The problem with filtering on user PCs is that users often turn off their filtering software or at least fail to update these programs with sufficient frequency. Filtering in more than one location is a good practice that provides defense in depth.

Voice over IP (VoIP) is a client/server application in which telephone signals are transmitted over IP packet-switched networks (including the Internet) instead of over circuit-switched networks. There are two major VoIP signaling protocols: the ISO H.323 standard and the Session Initiation Protocol (SIP), which is growing rapidly. While servers are involved in signaling, transport is done directly between the two VoIP clients. Transport packets have an IP header, a UDP header, an RTP header, and a segment of application data.

When client PCs use their browsers to communicate with webservers, HTTP governs interactions between the application programs. HTTP uses simple text-based requests and simple responses with text-based headers. HTTP can download many types of files. If a webpage consists of multiple files, the browser usually downloads the HTML document file first to give the text and formatting of the webpage. It then downloads graphics and other aspects of the webpage. MIME fields are used to describe the format of a downloaded file.

Most applications today are client/server applications. In peer-to-peer applications, in contrast, user PCs do most or all of the work. In a few P2P applications, no servers are used. However, it often makes sense to use servers to facilitate limited aspects of P2P applications. For instance, Skype is still considered a P2P application despite its use of a login server. Similarly, SETI@home is considered P2P even though it uses a server that downloads to and accepts data from the SETI@home user PCs. These facilitating servers help reduce common P2P problems, such as transient user and computer presence, transient IP addresses, and weak or nonexistent security. However, the application uses peer processing for *most* of its work—just not all of it.

There are several categories of P2P applications. So far, P2P has been dominated by file-sharing applications (such as BitTorrent), communication applications (such as Skype), and processor-sharing applications (such as SETI@home). New types and categories are appearing, including tools to anonymize Internet use (such as the Tor network).

END-OF-CHAPTER QUESTIONS

Thought Questions

11-1. Do you think that pure P2P architectures will be popular in the future? Why or why not?

11-2. Come up with a list of roles that facilitating servers can play in P2P applications. This will require you to read through the section on P2P applications carefully. You should also try to think of an example not in the text.

Troubleshooting Question

11-3. You perform a BitTorrent search and get no responses. Troubleshoot this problem.

Perspective Questions

11-4. What was the most surprising thing for you in the chapter?

11-5. What was the most difficult material for you in the chapter? Why was it difficult?