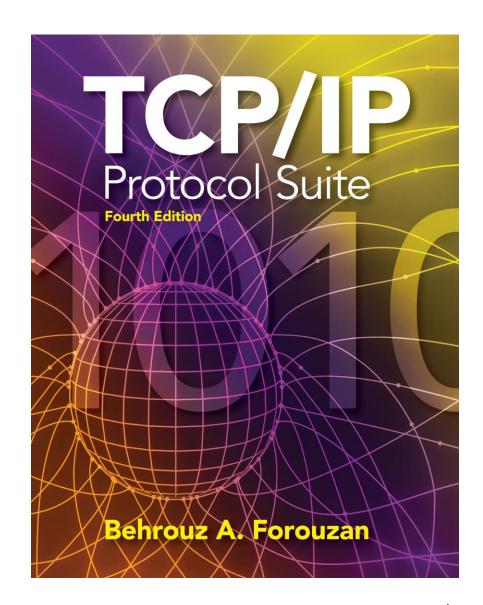
The McGraw·Hill Companies

Chapter 23

Electronic Mail: SMTP, POP IMAP, and MIME



OBJECTIVES:

- ☐ To explain the architecture of electronic mail using four scenarios.
- ☐ To explain the user agent (UA), services provided by it, and two types of user agents.
- ☐ To explain the mechanism of sending and receiving e-mails.
- ☐ To introduce the role of a message transfer agent and Simple Mail
- ☐ Transfer Protocol (SMTP) as the formal protocol that handles MTA.
- ☐ To explain e-mail transfer phases.
- ☐ To discuss two message access agents (MAAs): POP and IMAP.

OBJECTIVES (continued):

- ☐ To discuss MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa.
- ☐ To discuss the idea of Web-based e-mail.
- **☐** To explain the security of the e-mail system.

Chapter **Outline**

23.1 Architecture
23.2 User Agent

23.3 Message Transfer Agent

23.4 Message Access Agent

23.5 MIME

23.6 Web-Based Mail

23.7 Electronic Mail Security

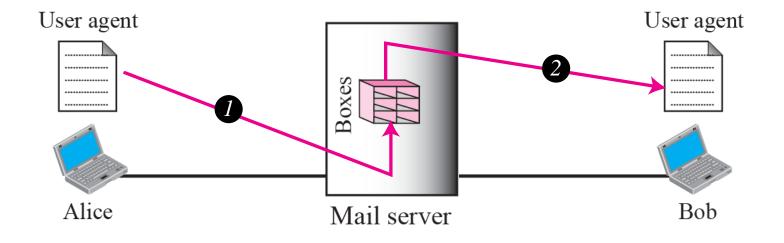
23-1 ARCHITECTURE

To explain the architecture of e-mail, we give four scenarios. We begin with the simplest situation and add complexity as we proceed. The fourth scenario is the most common in the exchange of e-mail.

Topics Discussed in the Section

- **✓** First Scenario
- **✓ Second Scenario**
- **✓ Third Scenario**
- **✓ Fourth Scenario**

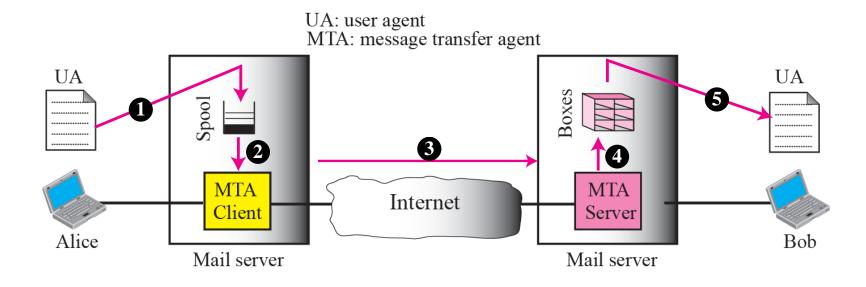
Figure 23.1 First scenario



Note

When the sender and the receiver of an e-mail are on the same mail server, we need only two user agents.

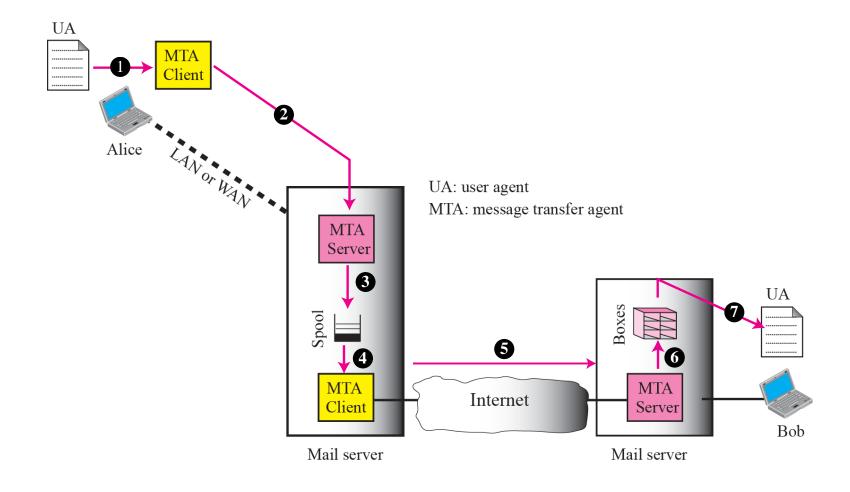
Figure 23.2 Second scenario



Note

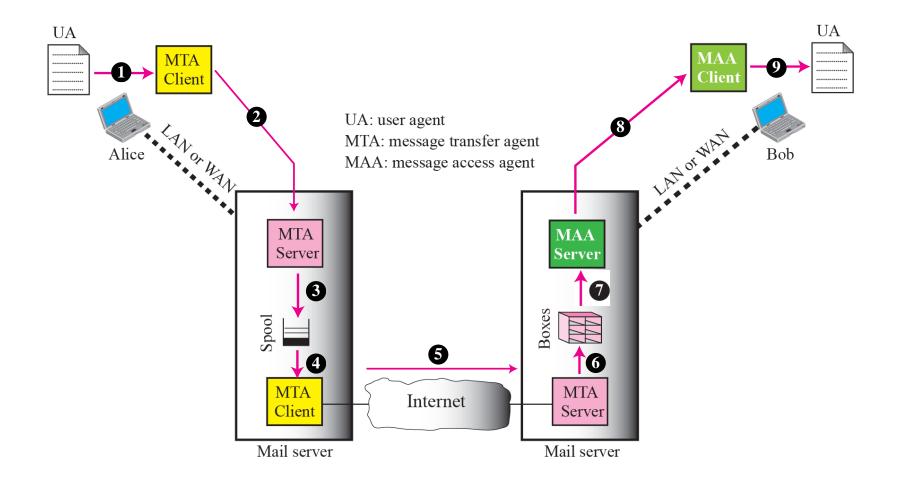
When the sender and the receiver of an e-mail are on different mail servers, we need two UAs and a pair of MTAs (client and server).

Figure 23.3 Third scenario



Note

When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).



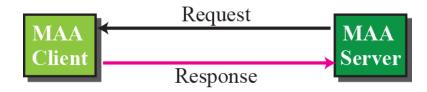
Note

When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

Figure 23.5 Push versus pull



a. Client pushes messages



b. Client pulls messages

23-2 USER AGENT

The first component of an electronic mail system is the user agent (UA). It provides service to the user to make the process of sending and receiving a message easier.

Topics Discussed in the Section

- ✓ Services Provided by a User Agent
- **✓** User Agent Types
- **✓** Sending Mail
- **✓** Receiving Mail
- **✓** Addresses
- **✓** Mailing List or Group List

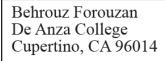
Note

Some examples of command-driven user agents are mail, pine, and elm.

Note

Some examples of GUI-based user agents are Eudora, Outlook, And Netscape.

Figure 23.6 Format of an email



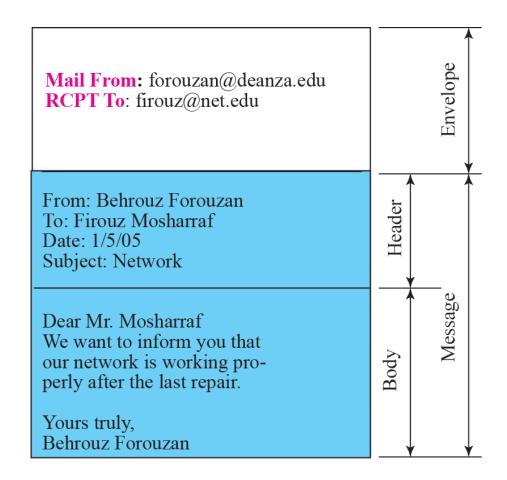
Firouz Mosharraf Com-Net Cupertino, CA 95014

Firouz Mosharraf Com-Net Cupertino, CA 95014 Jan. 5, 2005

Subject: Network

Dear Mr. Mosharraf We want to inform you that our network is working properly after the last repair.

Yours truly, Behrouz Forouzan



Local part

(a)

Domain name

Mailbox address of the recepient

The domain name of the mail server

23-3 MESSAGE TRANSFER AGENT

The actual mail transfer is done through message transfer agents (MTAs). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). As we said before, two pairs of MTA client-server programs are used in the most common situation (fourth scenario). Figure 23.8 shows the range of the SMTP protocol in this scenario.

Topics Discussed in the Section

- **✓ Commands and Responses**
- **✓** Mail Transfer Phases

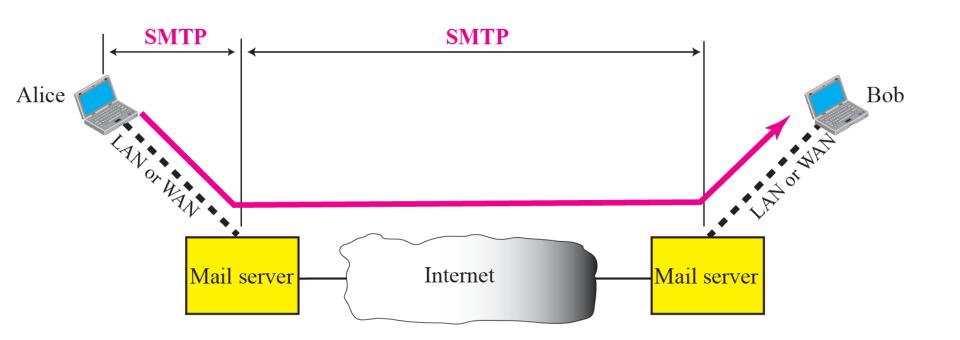


Figure 23.9 Commands and responses





 Table 23.1
 Commands

Keyword	Argument(s)	Keyword	Argument(s)
HELO	Sender's host name	NOOP	
MAIL FROM	Sender of the message	TURN	
RCPT TO	Intended recipient	EXPN	Mailing list
DATA	Body of the mail	HELP	Command name
QUIT		SEND FROM	Intended recipient
RSET		SMOL FROM	Intended recipient
VRFY	Name of recipient	SMAL FROM	Intended recipient



 Table 23.2
 Responses

Code	Description		
Positive Completion Reply			
211	System status or help reply		
214	Help message		
220	Service ready		
221	Service closing transmission channel		
250	Request command completed		
251	User not local; the message will be forwarded		
Positive Intermediate Reply			
354	Start mail input		
Transient Negative Completion Reply			
421	Service not available		
450	Mailbox not available		
451	Command aborted: local error		
452	Command aborted; insufficient storage		
	Permanent Negative Completion Reply		
500	Syntax error; unrecognized command		
501	Syntax error in parameters or arguments		
502	Command not implemented		
503	Bad sequence of commands		
504	Command temporarily not implemented		
550	Command is not executed; mailbox unavailable		
551	User not local		
552	Requested action aborted; exceeded storage location		
553	Requested action not taken; mailbox name not allowed		
554	Transaction failed		

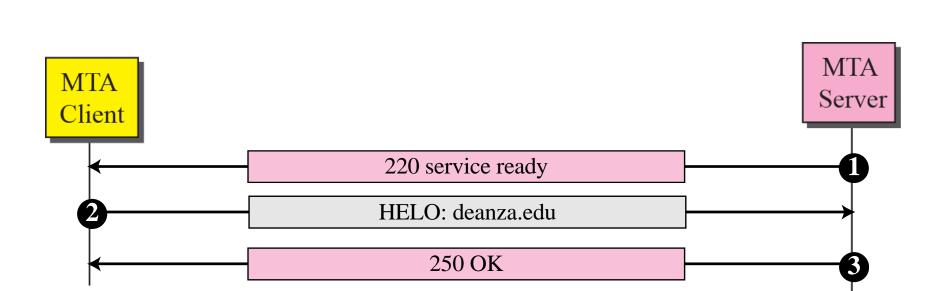


Figure 23.11 Message transfer

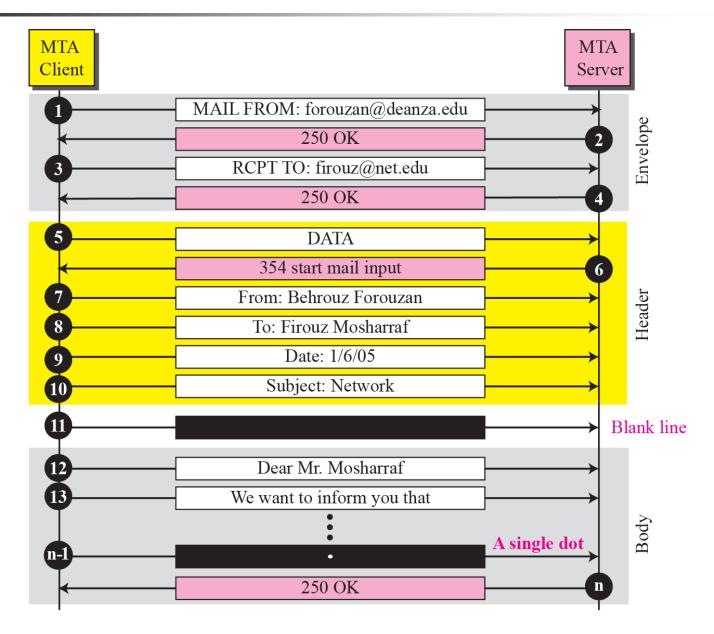
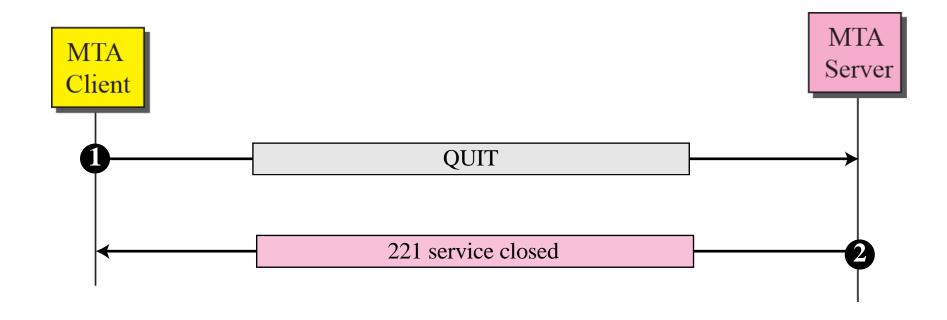


Figure 23.12 Connection termination



Example 23.1

Let us see how we can directly use SMTP to send an e-mail and simulate the commands and responses we described in this section. We use TELNET to log into port 25 (the well-known port for SMTP). We then use the commands directly to send an e-mail. In this example, forouzanb@adelphia.net is sending an e-mail to himself. The first few lines show TELNET trying to connect to the adelphia mail server.

```
$ telnet mail.adelphia.net 25
Trying 68.168.78.100...
Connected to mail.adelphia.net (68.168.78.100).
```

After connection, we can type the SMTP commands and then receive the responses as shown below. We have shown the commands in black and the responses in color. Note that we have added for clarification some comment lines, designated by the "=" sign. These lines are not part of the e-mail procedure.

Example 19.1 Continued

```
=========== Connection Establishment =============
 220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
 250 mta13.adelphia.net
                          Envelope
______
MAIL FROM: forouzanb@adelphia.net
 250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
 250 Recipient <forouzanb@adelphia.net> Ok
                       Header and Body
_____
DATA
 354 Ok Send data ending with <CRLF>.<CRLF>
From: Forougan
TO: Forouzan
This is a test message
to show SMTP in action.
```

```
250 Message received: adelphia.net@mail.adelphia.net
QUIT

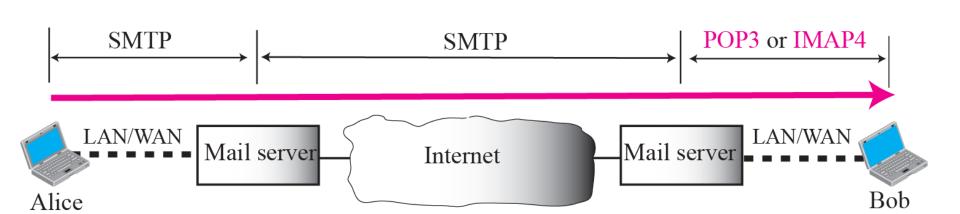
221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

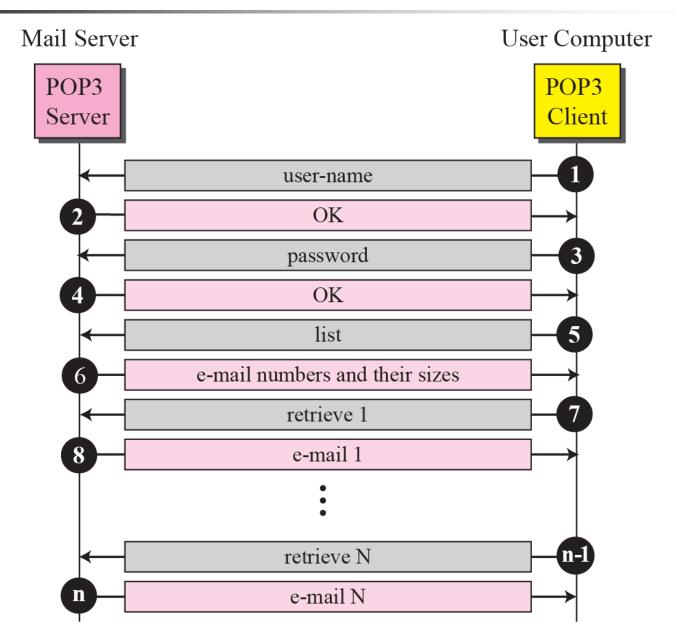
23-4 MESSAGE ACCESS AGENT

The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server. In other words, the direction of the bulk data (messages) is from the client to the server. On the other hand, the third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data are from the server to the client. The third stage uses a message access agent.

Topics Discussed in the Section

- ✓ POP3
- ✓ IMAP4





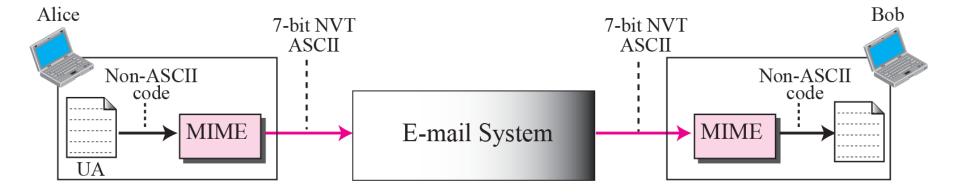
23-5 MIME

Electronic mail has a simple structure. Its simplicity, however, comes with a price. It can send messages only in NVT 7-bit ASCII format. In other words, it has some limitations. Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.

Topics Discussed in the Section

✓ MIME Headers







E-mail header

MIME headers

MIME-Version: 1.1

Content-Type: type/subtype

Content-Transfer-Encoding: encoding type

Content-Id: message id

Content-Description: textual explanation of nontextual contents

E-mail body



Туре	Subtype	Description				
Text	Plain	Unformatted				
	HTML	HTML format (see Appendix E)				
Multipart	Mixed	Body contains ordered parts of different data types				
	Parallel	Same as above, but no order				
	Digest	Similar to Mixed, but the default is message/RFC822				
	Alternative	Parts are different versions of the same message				
Message	RFC822	Body is an encapsulated message				
	Partial	Body is a fragment of a bigger message				
	External-Body	Body is a reference to another message				
Image	JPEG	Image is in JPEG format				
	GIF	Image is in GIF format				
Video	MPEG	Video is in MPEG format				
Audio	Basic	Single channel encoding of voice at 8 KHz				
Application	PostScript	Adobe PostScript				
	Octet-stream	General binary data (eight-bit bytes)				



 Table 23.4
 Content-Transfer-Encoding

Туре	Description
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign plus an ASCII code

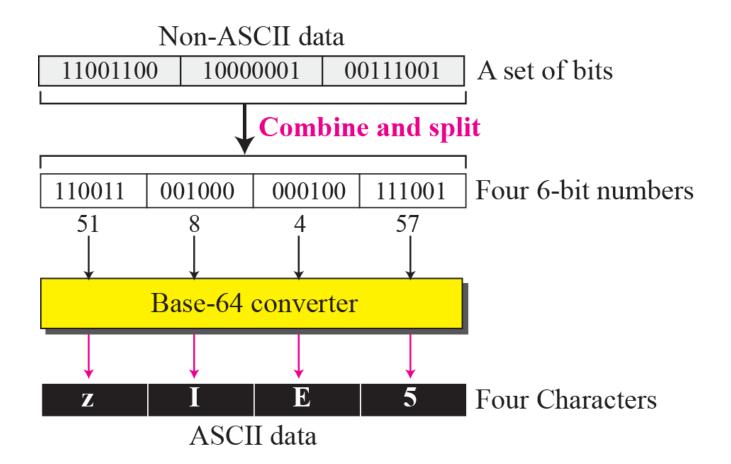
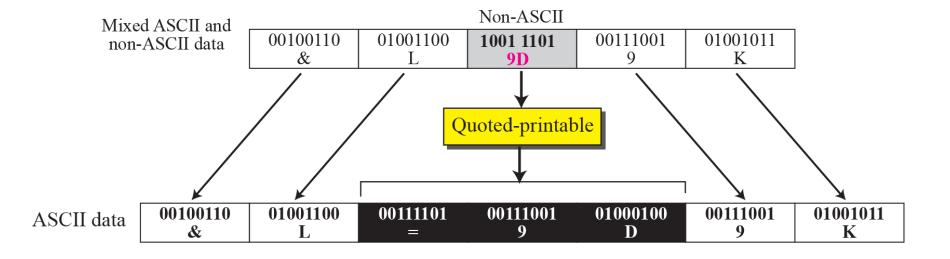


 Table 23.5
 Base-64 Converting Table

Value	Code										
0	A	11	L	22	W	33	h	44	S	55	3
1	В	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	0	25	Z	36	k	47	V	58	6
4	E	15	P	26	a	37	l	48	W	59	7
5	F	16	Q	27	b	38	m	49	X	60	8
6	G	17	R	28	С	39	n	50	y	61	9
7	H	18	S	29	d	40	0	51	Z	62	+
8	I	19	T	30	e	41	р	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

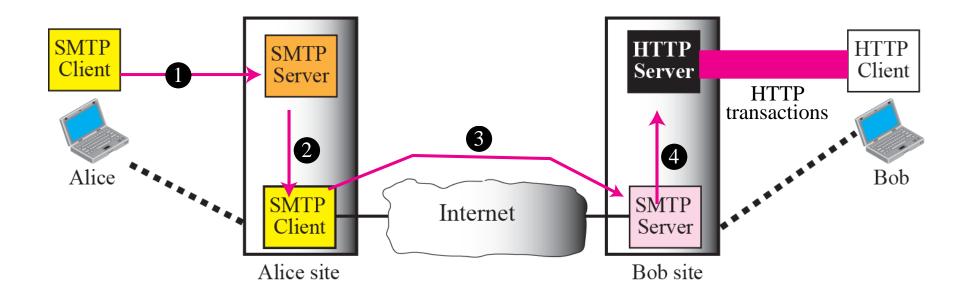


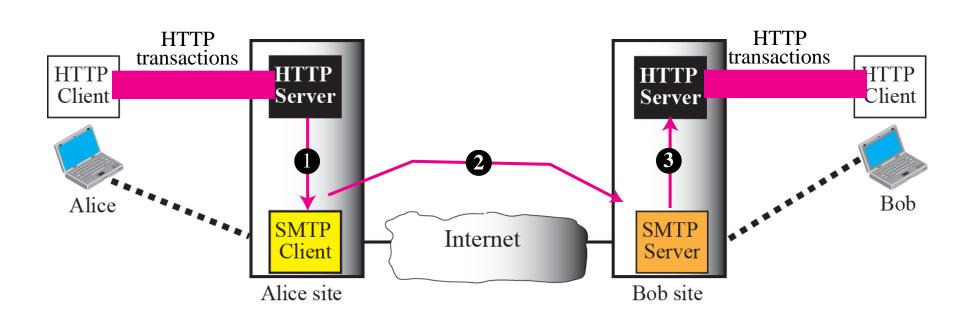
23-6 WEB-BASED MAIL

E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Three common sites are Hotmail, Yahoo, and Google. The idea is very simple. Let us go through two cases:

Topics Discussed in the Section

- ✓ Case I
- **✓ Case II**





23-6 E-MAIL SECURITY

The protocol discussed in this chapter does not provide any security provisions per se. However, e-mail exchanges can be secured using two application-layer securities designed in particular for e-mail systems. Two of these protocols, Pretty Good Privacy (PGP) and Secure MIME (SMIME) are discussed in Chapter 30 after we have discussed the basic network security.