Chapter 5

Application Layer

- INTRODUCTION
- CLIENT-SERVER PARADIGM
- STANDARD APPLICATIONS
  - HTTP
  - FTP
  - **SMTP**
  - Telnet
  - SSH
  - DNS

Electronic Mail

- Architecture
- User Agent
  - Sending Mail
  - Receiving Mail
  - Addresses
  - Mailing List or Group List
- Message Transfer Agent: SMTP
  - Commands and Responses
  - Mail Transfer Phases
- Message Access Agent: POP and IMAP
- MIME
  - MIME Headers
- Web-Based Mail
- E-Mail Security

Electronic Mail

Electronic mail (or e-mail) allows users to exchange messages. The nature of this application, however, is different from other applications discussed so far. In an application such as HTTP or FTP, the server program is running all the time, waiting for a request from a client. When the request arrives, the server provides the service. In the case of electronic mail, the situation is different.

First, e-mail is considered a one-way transaction. When Alice sends an e-mail to Bob, she may expect a response, but this is not a mandate. Bob may or may not respond. If he does respond, it is another one-way transaction. Second, it is neither feasible nor logical for Bob to run a server program and wait until someone sends an e-mail to him. Bob may turn off his computer when he is not using it. This means that the idea of client/ server programming should be implemented in another way: using some intermediate computers (servers).

Architecture

Mail Box:

- The administrator has created one mailbox for each user where the received messages are stored. A mailbox is part of a server hard drive, a special file with permission restrictions.
- The administrator has also created a queue (spool) to store messages waiting to be sent.

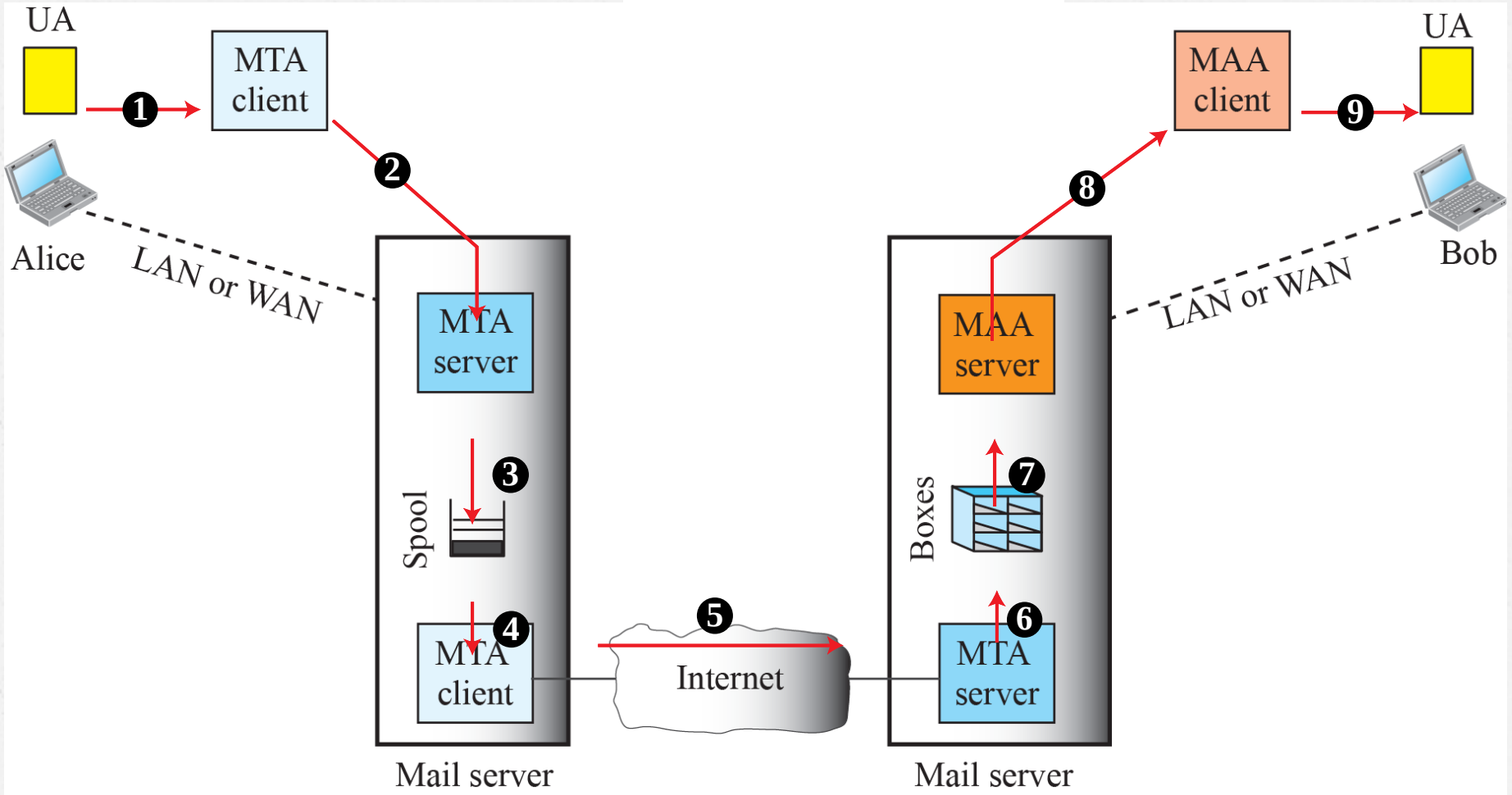
Uses three different agents:

- **a User Agent (UA),**
- **a Mail Transfer Agent (MTA), and**
- **a Message Access Agent (MAA).**

The electronic mail system needs two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server).

Common scenario

UA: user agent
MTA: message transfer agent
MAA: message access agent



User Agent

- It provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers.
- Two types of user agents:

Command-driven:

- belong to the early days of electronic mail. They are still present as the underlying user agents.
- A command-driven user agent normally accepts a one character
- command from the keyboard to perform its task.
- e.g. mail, pine, and elm

GUI-based:

- They contain graphical user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse.
- e.g. Eudora and Outlook

The main functionalities of User agent are:

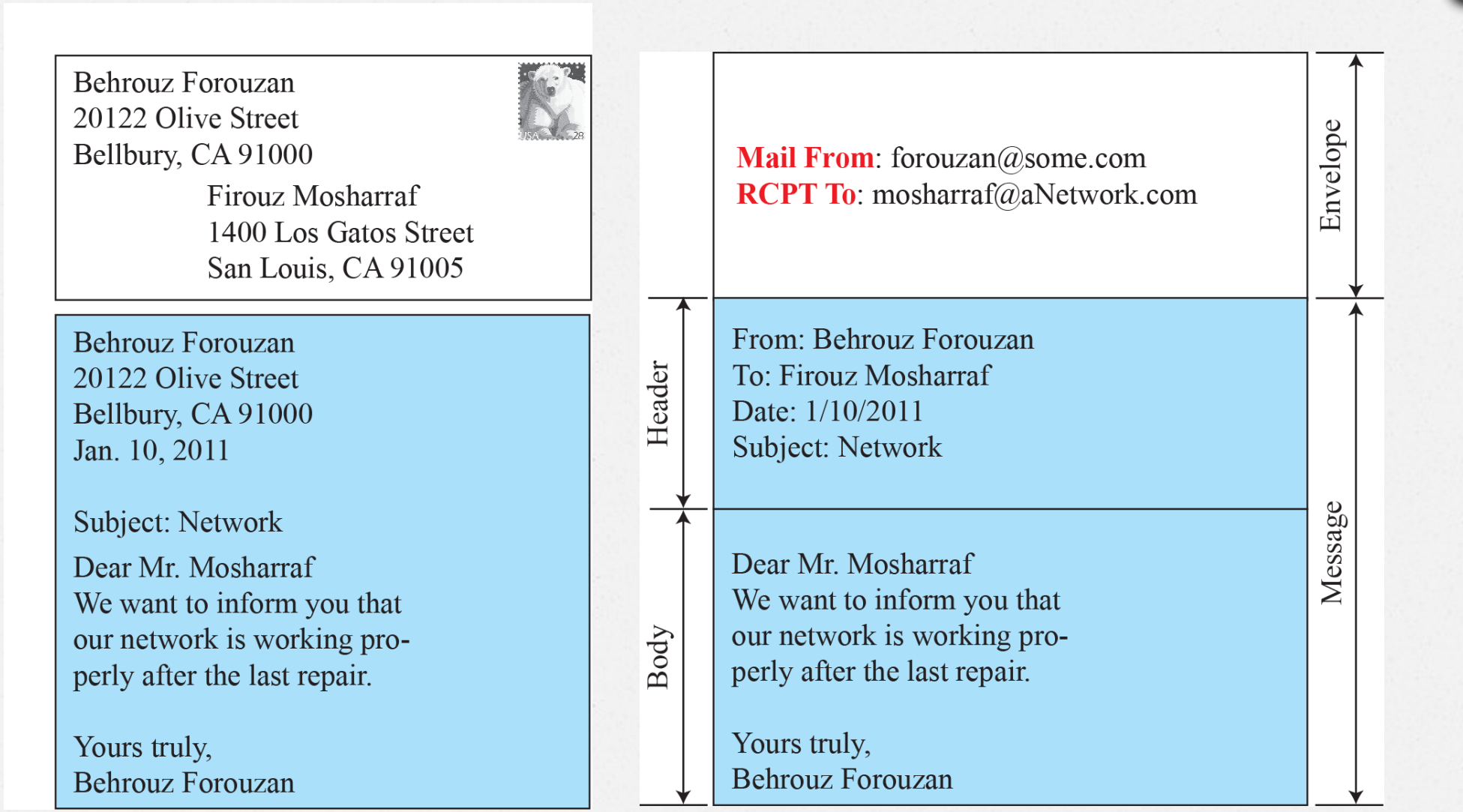
Sending Mail

- To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message.
  - The envelope usually contains the sender address, the receiver address, and other information.
  - The message contains the header and the body.

Receiving Mail

- The user agent is triggered by the user (or a timer). If a user has mail, the UA informs the user with a notice.
- If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox.

Format of an e-mail



E-mail address

To deliver mail, a mail handling system must use an addressing system with unique addresses.

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

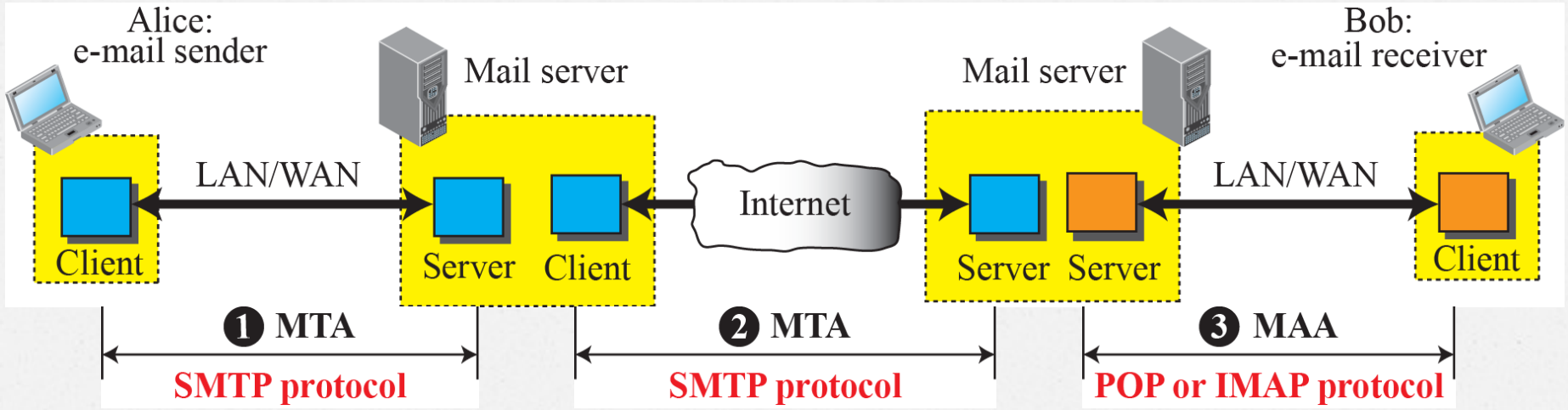
The second part of the address is the **domain name**. An organization usually selects one or more hosts to receive and send e-mail; they are sometimes called mail servers or exchangers. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name

Mailing List or Group List

Electronic mail allows one name, an alias, to represent several different e-mail addresses; this is called a mailing list.

Every time a message is to be sent, the system checks the recipient's name against the alias database; if there is a mailing list for the defined alias, separate messages, one for each entry in the list, must be prepared and handed to the MTA.

Protocols used in electronic mail



Simple Mail Transfer Protocol (SMTP) → MTA

Post Office Protocol(POP3) and Internet Mail Access Protocol(IMAP4) → MAA

SMTP: Well known port 25 POP/IMAP: Well known port 110

Message Transfer Agent: SMTP

- The formal protocol that defines the MTA client and server in the Internet
- SMTP is used two times, between **the sender and the sender's mail server** and **between the two mail servers**.
- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.
  - The command is from an MTA client to an MTA server;
  - the response is from an MTA server to the MTA client.

Each command or reply is terminated by a two character (carriage return and line feed) end-of-line token.

SMTP Commands:

- Commands are sent from the client to the server
- It consists of a keyword followed by zero or more arguments

SMTP Commands

Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list.
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *or* the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *and* the mailbox of the recipient

SMTP Responses:

- Responses are sent from the server to the client.
- A response is a three digit code that may be followed by additional textual information.

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

SMTP responses (continued)

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

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment:
After a client has made a TCP connection to the well known port 25, the SMTP server starts the connection phase. This phase involves the following three steps:

1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).
2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.
3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer

After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps: (3&4 for more recpients)

1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
2. The server responds with code 250 or some other appropriate code.
3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.
4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.
7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. The server responds with code 250 (OK) or some other appropriate code.

Connection Termination

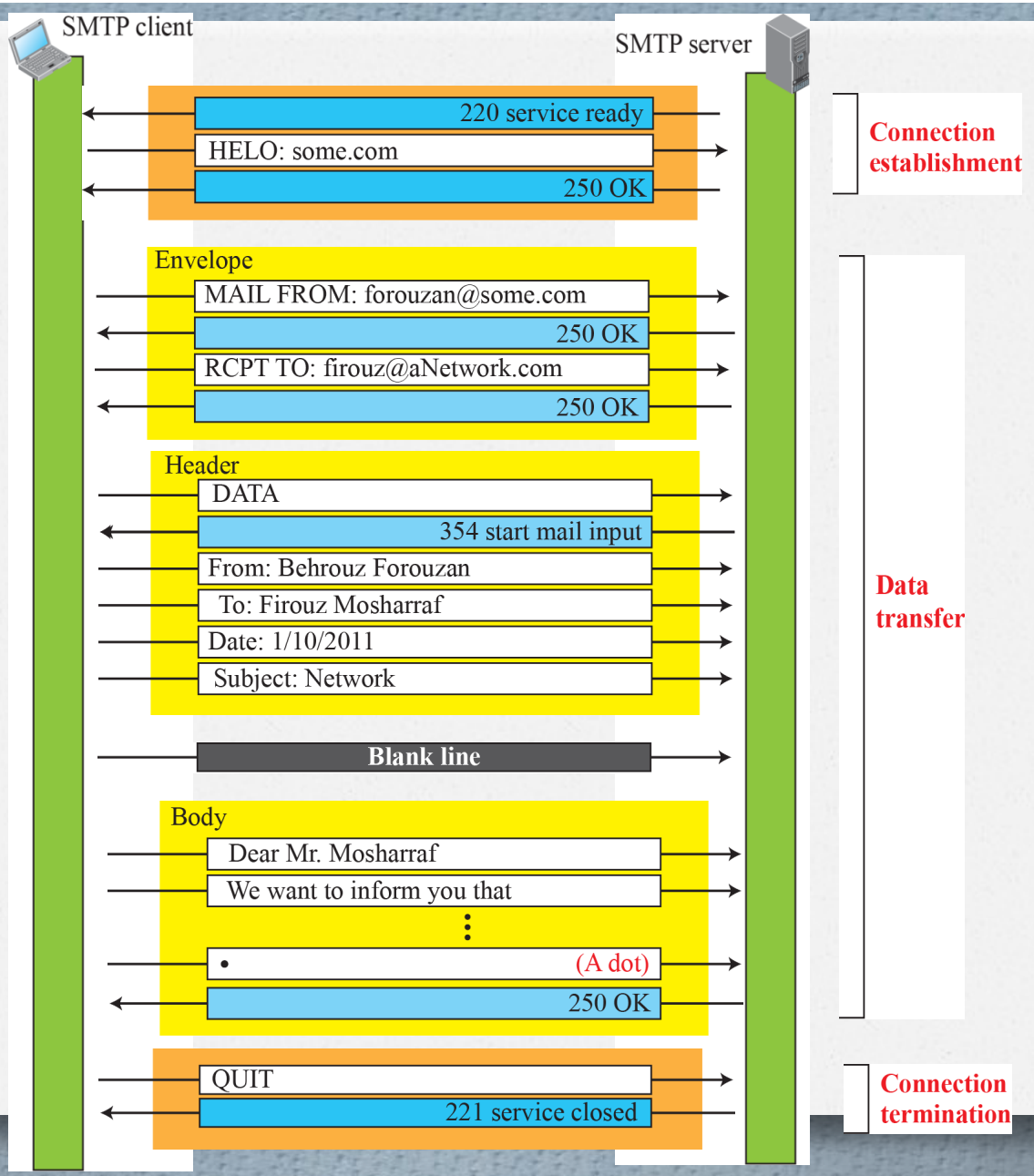
After the message is transferred successfully, the client terminates the connection. This phase involves two steps.

1. The client sends the QUIT command.
2. The server responds with code 221 or some other appropriate code.

Example

To show the three mail transfer phases, we show all of the steps described above using the information depicted in Figure. In the figure, we have separated the messages related to the envelope, header, and body in the data transfer section. Note that the steps in this figure are repeated two times in each e-mail transfer: once from the e-mail sender to the local mail server and once from the local mail server to the remote mail server. The local mail server, after receiving the whole e-mail message, may spool it and send it to the remote mail server at another time.

Example


SMTP client
SMTP server
220 service ready
HELO: some.com
250 OK
Envelope
MAIL FROM: forouzan@some.com
250 OK
RCPT TO: firouz@aNetwork.com
250 OK
Header
DATA
354 start mail input
From: Behrouz Forouzan
To: Firouz Mosharraf
Date: 1/10/2011
Subject: Network
Blank line
Body
Dear Mr. Mosharraf
We want to inform you that
•
(A dot)
250 OK
QUIT
221 service closed
Connection establishment
Data transfer
Connection termination

Message Access Agent: POP and IMAP

- The first and 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.
- the third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent.
- Currently two message access protocols are available:
  - Post Office Protocol, version 3 (POP3) and
  - Internet Mail Access Protocol, version 4 (IMAP4).

Post Office Protocol, Version 3 (POP3)

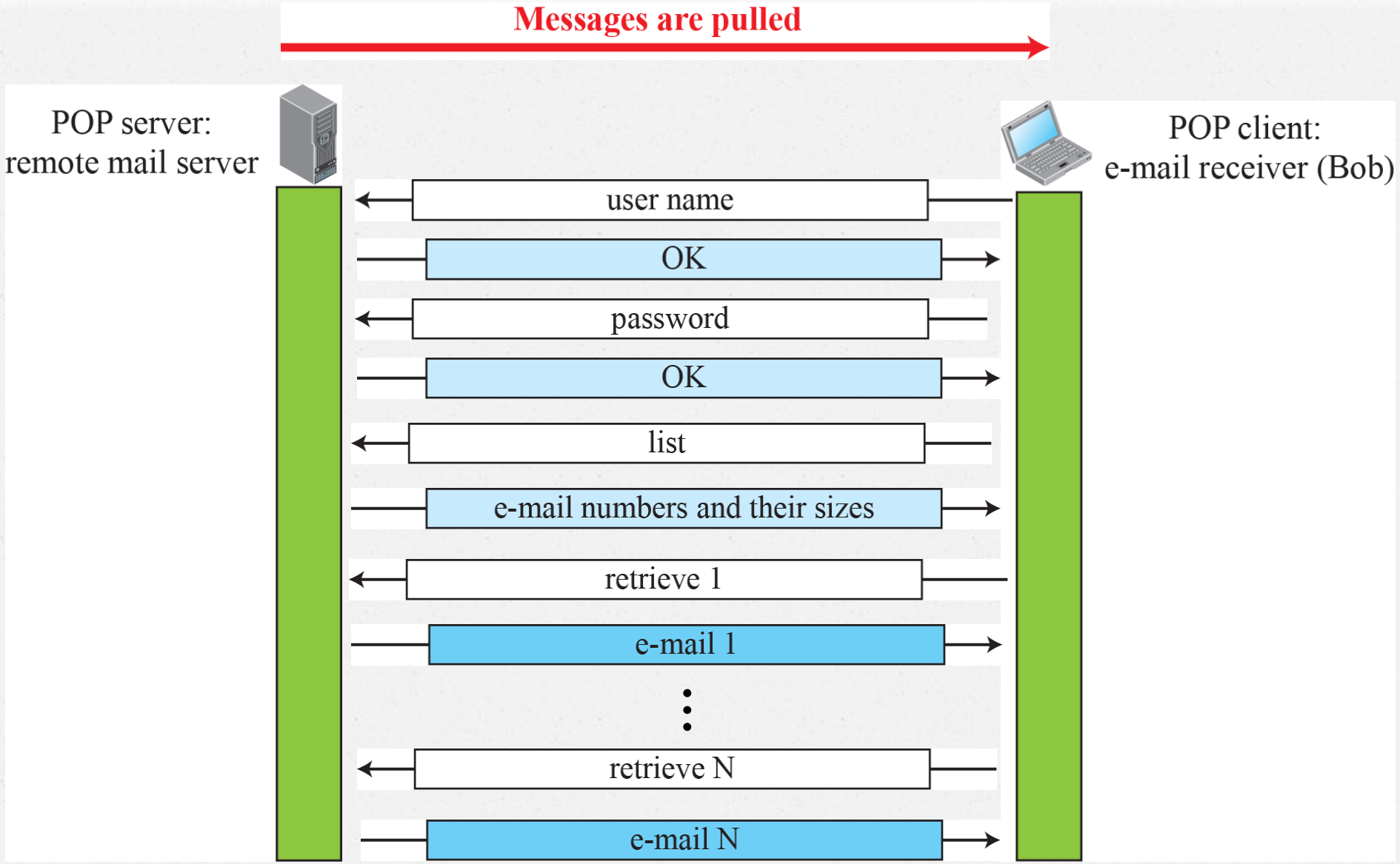
The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

- The client opens a connection to the server on TCP port 110.
- It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

POP3 has two modes: the delete mode and the keep mode.

- In the **delete mode**, the mail is deleted from the mailbox after each retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.
- In the **keep mode**, the mail remains in the mailbox after retrieval. The keep mode is normally used when the user accesses her mail away from her primary computer (for example, from a laptop). The mail is read but kept in the system for later retrieval and organizing.

POP3


Messages are pulled
POP server:
remote mail server
POP client:
e-mail receiver (Bob)
user name
OK
password
OK
list
e-mail numbers and their sizes
retrieve 1
e-mail 1
retrieve N
e-mail N

Internet Mail Access Protocol, Version 4 (IMAP4)

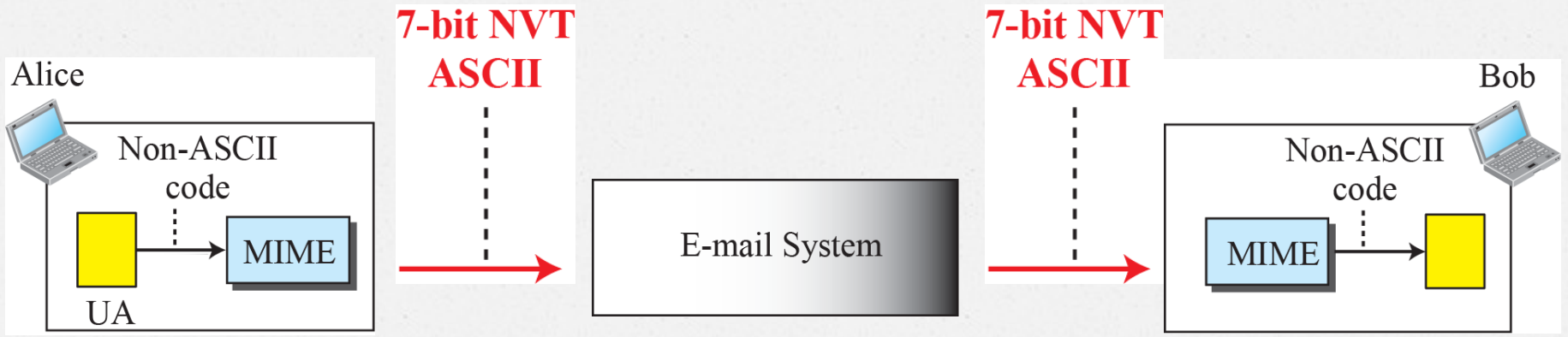
- IMAP4 is more powerful and more complex
- POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.
- IMAP4 provides the following extra functions:
  - A user can check the e-mail header prior to downloading.
  - A user can search the contents of the e-mail for a specific string of characters prior to downloading.
  - A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
  - A user can create, delete, or rename mailboxes on the mail server.
  - A user can create a hierarchy of mailboxes in a folder for e-mail storage.

Multipurpose Internet Mail Extensions (MIME)

- Electronic mail has a simple structure.
  - It can send messages only in NVT 7-bit ASCII format.
  - It cannot be used for languages other than English
  - It cannot be used to send binary files or video or audio data.

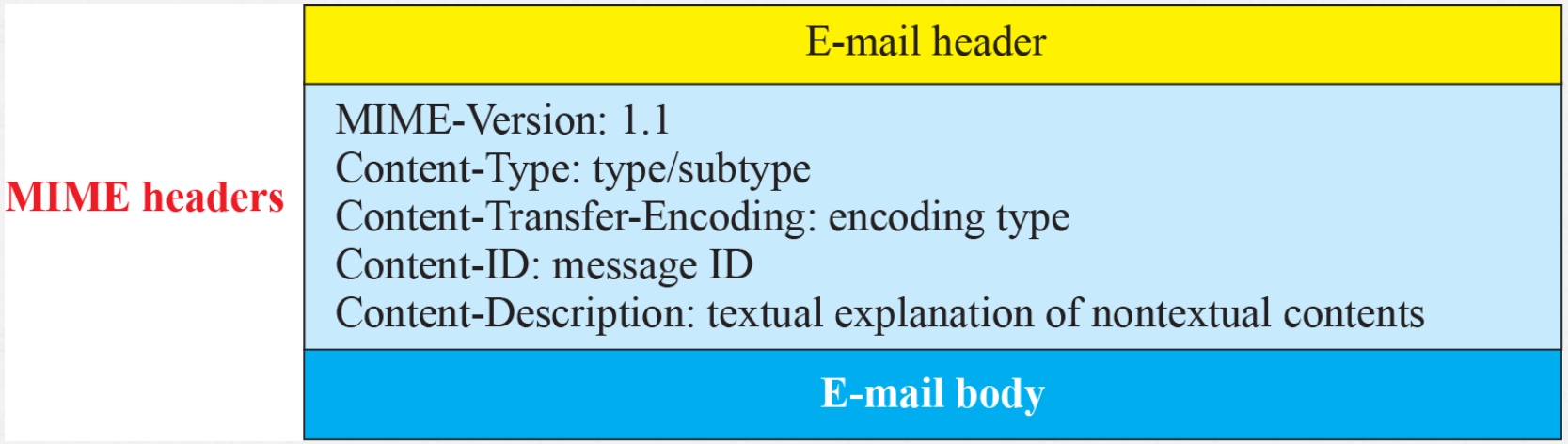
- MIME is a supplementary protocol that allows non-ASCII data to be sent through e-mail.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.

MIME


Alice
Non-ASCII
code
MIME
UA
7-bit NVT
ASCII
E-mail System
7-bit NVT
ASCII
Bob
Non-ASCII
code
MIME

MIME header

MIME defines five headers, which can be added to the original e-mail header section to define the transformation parameters



MIME-Version: This header defines the version of MIME used. The current version is 1.1.

Content-Type: This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash.

Data Types and Subtypes in MIME

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
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)

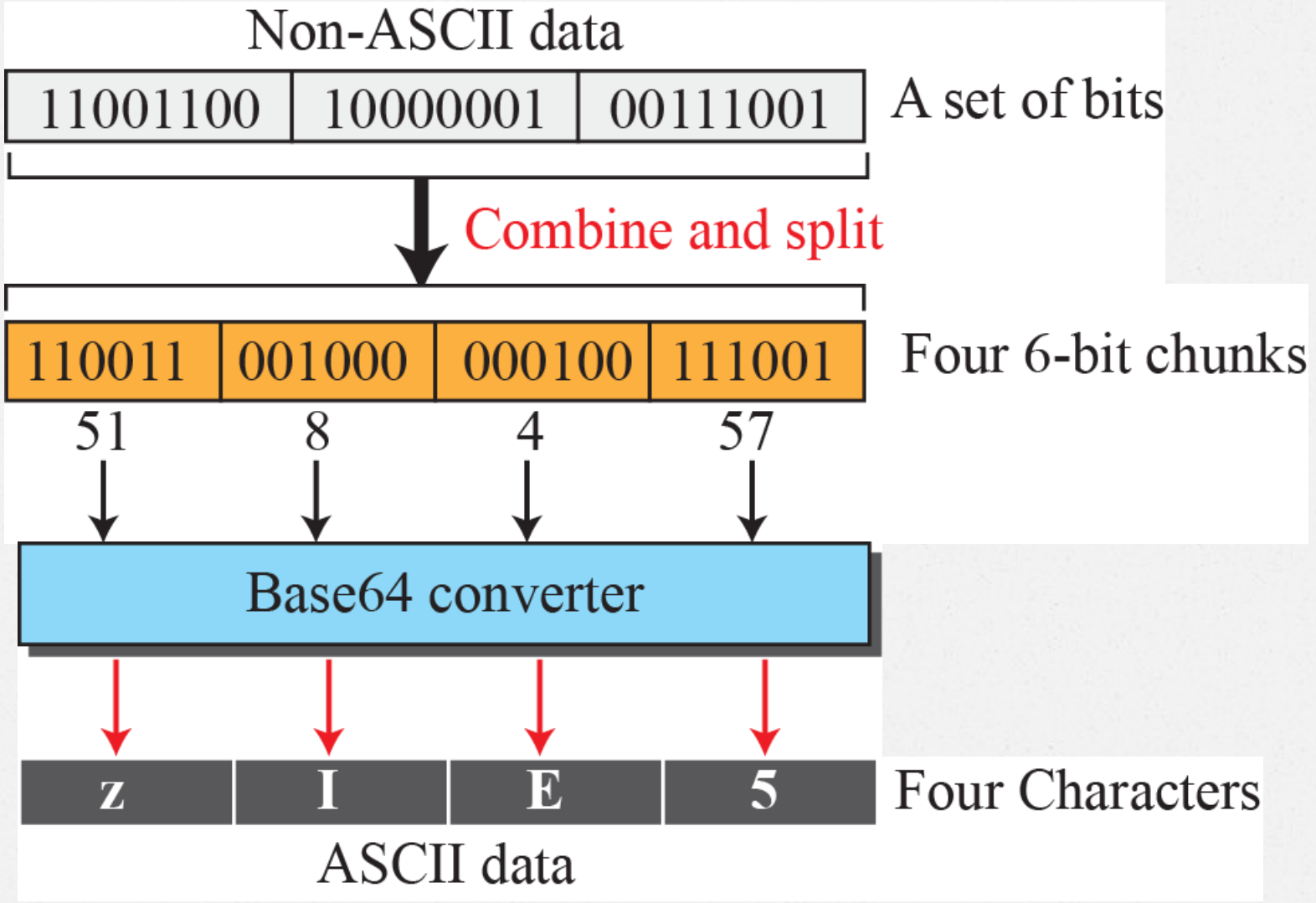
Methods for Content-Transfer-Encoding

Content-Transfer-Encoding: This header defines the method used to encode the messages into 0s and 1s for transport. The five types of encoding methods are listed in Table.

The last two encoding methods are interesting. In the Base64 encoding, data, as a string of bits, is first divided into 6-bit chunks

Type	*Description*
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

Base64 conversion



Base64 Converting Table

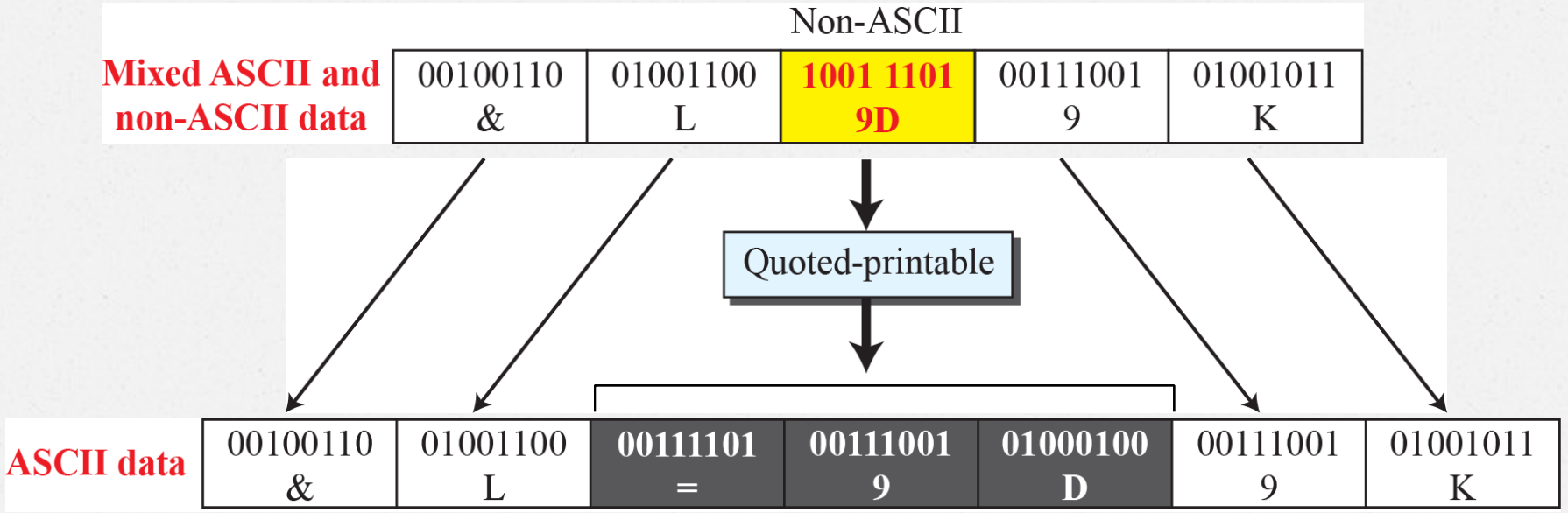
Value	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*
0	**A**	11	**L**	22	**W**	33	**h**	44	**s**	55	**3**
1	**B**	12	**M**	23	**X**	34	**i**	45	**t**	56	**4**
2	**C**	13	**N**	24	**Y**	35	**j**	46	**u**	57	**5**
3	**D**	14	**O**	25	**Z**	36	**k**	47	**v**	58	**6**
4	**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	**c**	39	**n**	50	**y**	61	**9**
7	**H**	18	**S**	29	**d**	40	**o**	51	**z**	62	+
8	**I**	19	**T**	30	**e**	41	**p**	52	**0**	63	/
9	**J**	20	**U**	31	**f**	42	**q**	53	**1**		
10	**K**	21	**V**	32	**g**	43	**r**	54	**2**		

- Base64 is a redundant encoding scheme; that is, every six bits become one ASCII character and are sent as eight bits. We have an overhead of 25 percent.
- If the data consist mostly of ASCII characters with a small non-ASCII portion, we can use quoted-printable encoding.
- In quoted-printable, if a character is ASCII, it is sent as is.
- If a character is not ASCII, it is sent as three characters. The first character is the equal sign (=). The next two characters are the hexadecimal representations of the byte.

Content-ID: This header uniquely identifies the whole message in a multiple message environment.

Content-Description: This header defines whether the body is image, audio, or video.

Quoted-printable



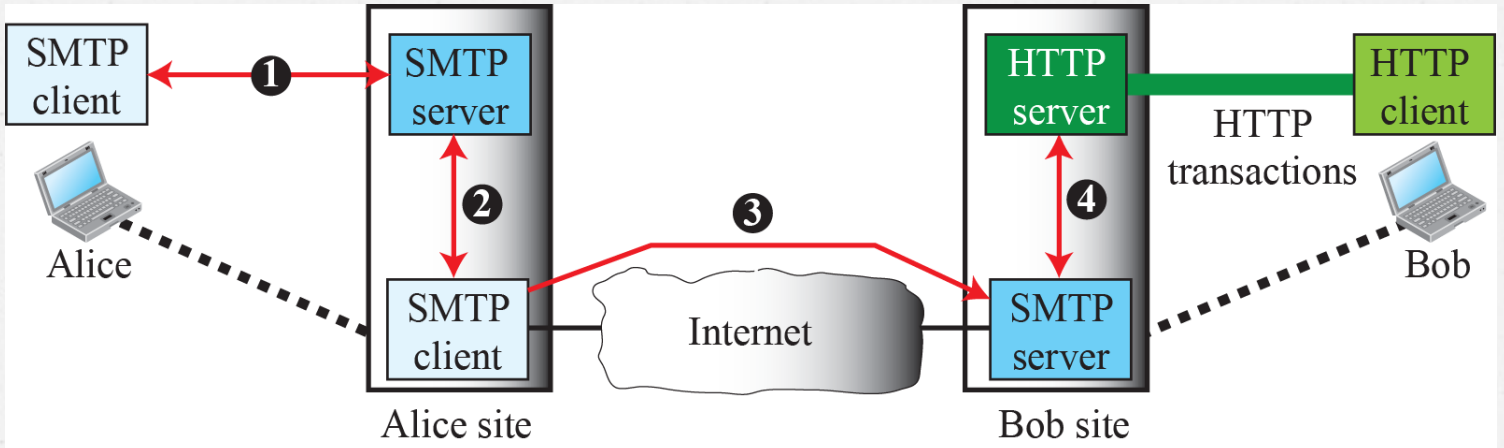
Web-Based Mail

- 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 mail.
- The idea is very simple. It has two cases:

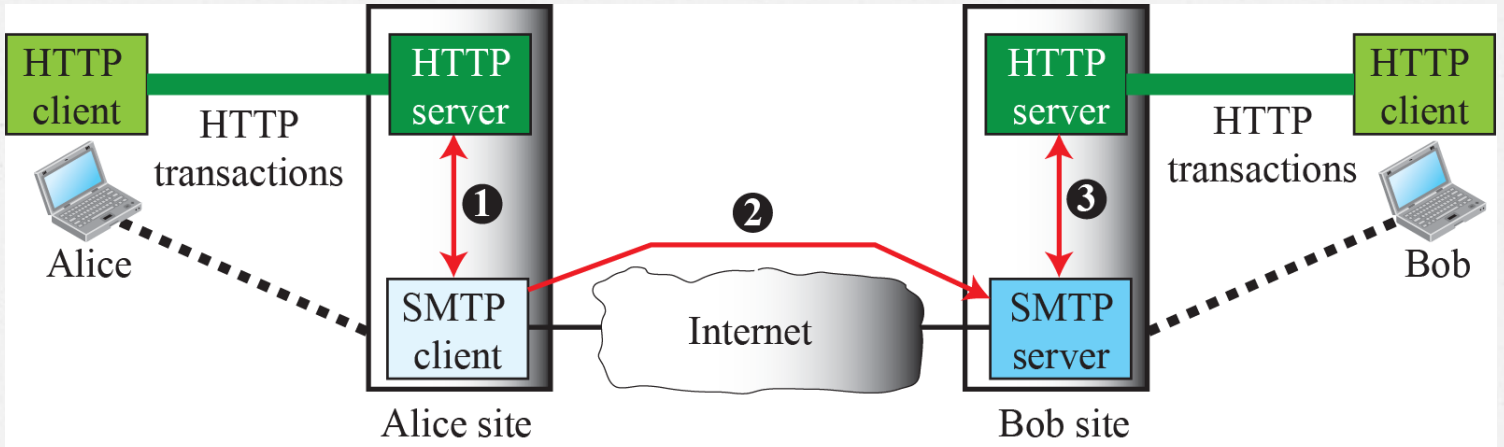
Case-1

Case-2

Web-based e-mail, cases I and II



Case 1: Only receiver uses HTTP



Case 2: Both sender and receiver use HTTP

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/Multipurpose Internet Mail Extensions (S/MIME)