STANDARD CLIENT-SERVER APPLICATIONS

FTP

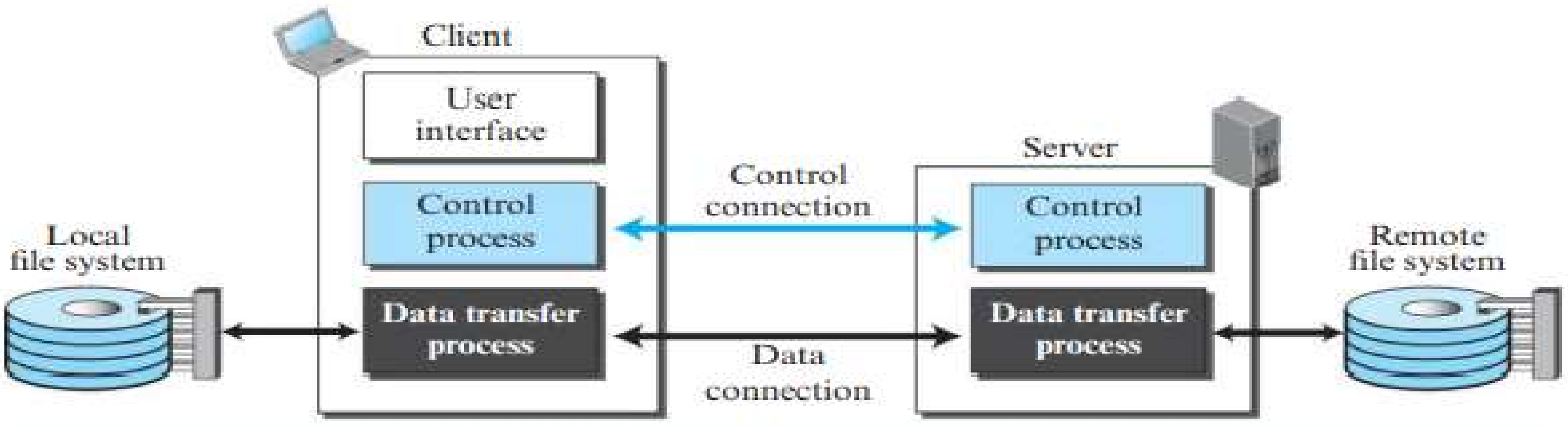
- File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent data. Two systems may have different directory structures.
- All of these problems have been solved by FTP in a very simple and elegant approach.
- Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

STANDARD CLIENT-SERVER APPLICATIONS

FTP

- Figure 1.44 shows the basic model of FTP.
- The client has three components: user interface, client control process, and the client data transfer process.
- The server has two components: the server control process and the server data transfer process.

Figure 1.44 *FTP*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

FTP

- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.
- Separation of commands and data transfer makes FTP more efficient.
- The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time.
- The data connection needs more complex rules due to the variety of data types transferred

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Lifetimes of Two Connections

- The two connections in FTP have different lifetimes.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transfer activity. It opens each time commands that involve transferring files are used, and it closes when the file is transferred.
- When a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

- For control communication, FTP uses the NVT ASCII character set as used by TELNET. [The Network Virtual Terminal (**NVT**) **ASCII** character set, defined in RFC 854, is used for transferring files with a Representation Type (TYPE) of ASCII.]
- Communication is achieved through commands and responses.
- This simple method is adequate for the control connection because we send one command (or response) at a time. Each line is terminated with a two-character (carriage return and line feed) end-of-line token.
- During the control connection, commands are sent from the client to the server and responses are sent from the server to the client.
- Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

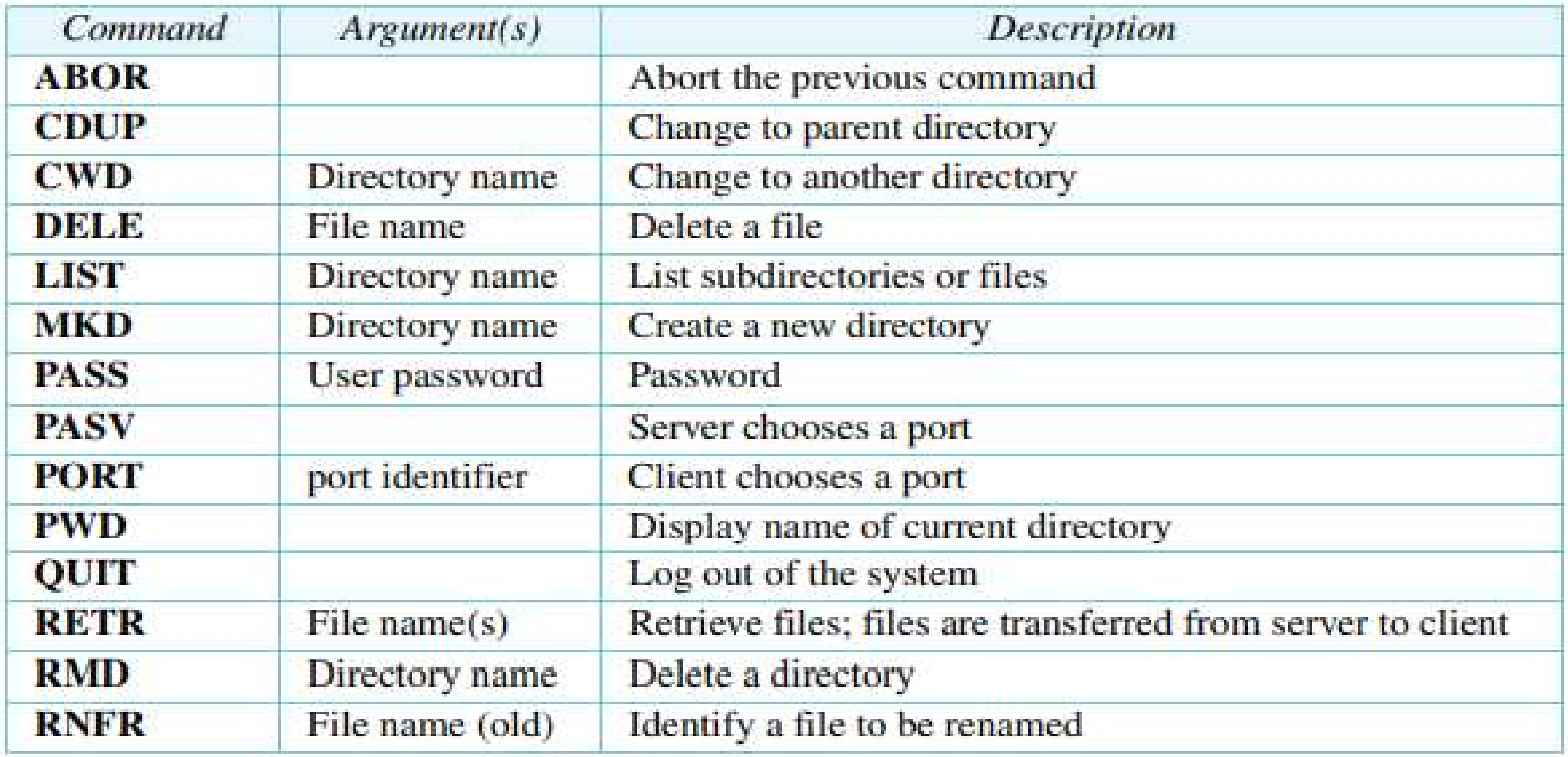
STANDARD CLIENT-SERVER APPLICATIONS

FTP

Control Connection

- Some of the most common commands are shown in Table 1.5.

Table 1.5 *Some FTP commands*



Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed

[Behrouz A Forouzan, Firouz Mosharraf, “Computer Networks: A top down Approach”, McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

- Every FTP command generates at least one response.
- A response has two parts: a three-digit number followed by text.
- The numeric part defines the code; the text part defines needed parameters or further explanations.
- The first digit defines the status of the command. The second digit defines the area in which the status applies. The third digit provides additional information.
- Table 1.6 shows some common responses.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

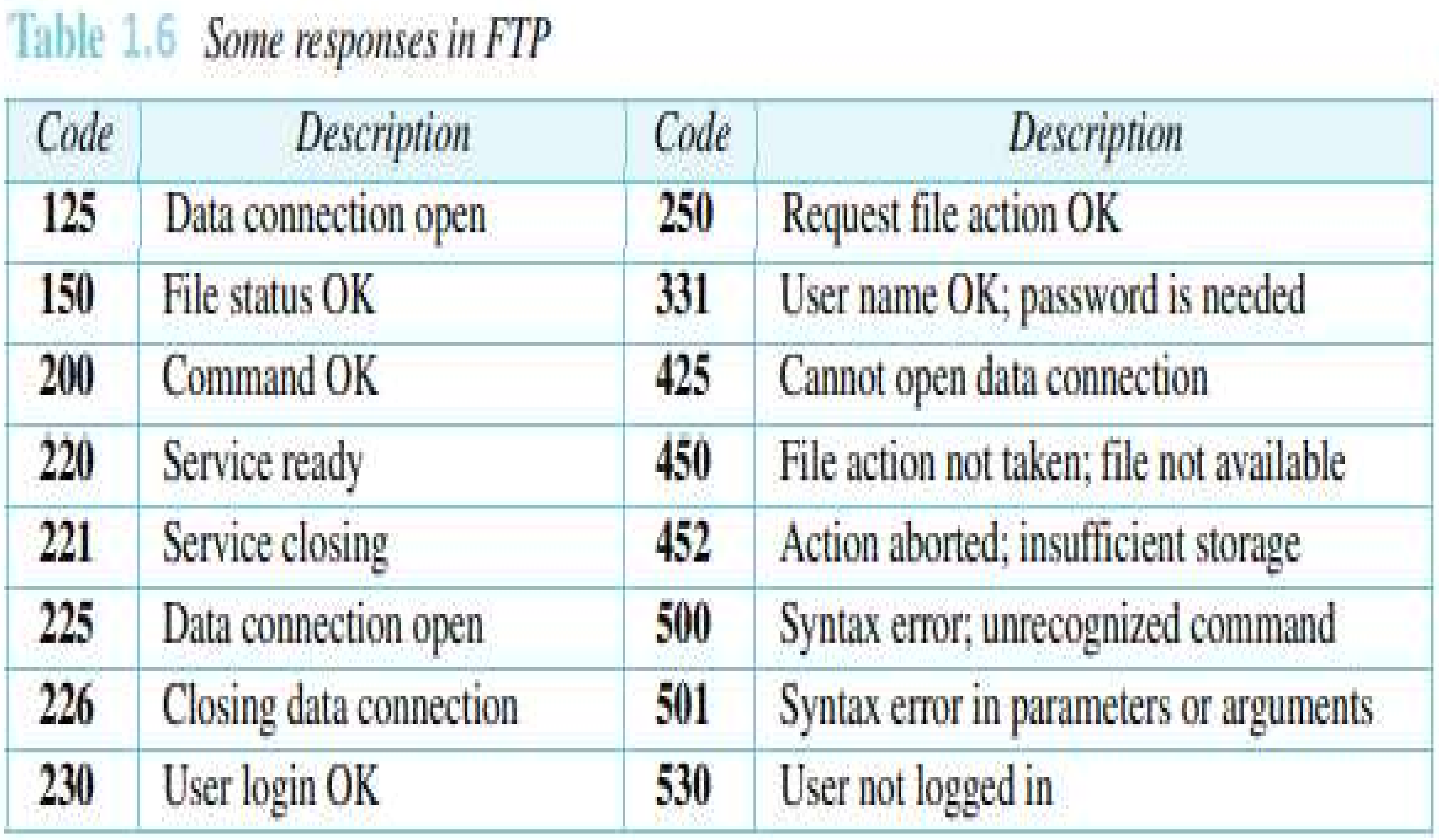


Table 1.6 *Some responses in FTP*

Code	*Description*	*Code*	*Description*
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Data Connection

- The data connection uses the well-known port 20 at the server site.
- The creation of a data connection is different from the control connection.
- The following shows the steps:

  1. The client issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
  2. The client sends this port number to the server using the PORT command.
  3. The server receives the port number and issues an active open using the well known port 20 and the received ephemeral port number.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Communication over Data Connection

- The purpose and implementation of the data connection are different from those of the control connection.
- We want to transfer files through the data connection.
- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
- Before sending the file through the data connection, we prepare for transmission through the control connection.
- The heterogeneity problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Data Structure

- FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data: file structure, record structure, or page structure.
- The file structure format has no structure. It is a continuous stream of bytes.
- In the record structure, the file is divided into records. This can be used only with text files.
- In the page structure, the file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

File Type

- FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Transmission Mode

- FTP can transfer a file across the data connection using one of the following three transmission modes: stream mode, block mode, or compressed mode.
- The stream mode is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes.
- In the block mode, data can be delivered from FTP to TCP in blocks.
- Each block is preceded by a 3-byte header. The first byte is called the block descriptor; the next two bytes define the size of the block in bytes.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection.

- File transfer in FTP means one of three things: retrieving a file (server to client), storing a file (client to server), and directory listing (server to client).

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Security for FTP

- The FTP protocol was designed when security was not a big issue.
- Although FTP requires a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker.
- The data transfer connection also transfers data in plaintext, which is insecure.
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

- Electronic mail (or e-mail) allows users to exchange messages.
- 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). The users run only client programs when they want and the intermediate servers apply the client/server paradigm.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Architecture

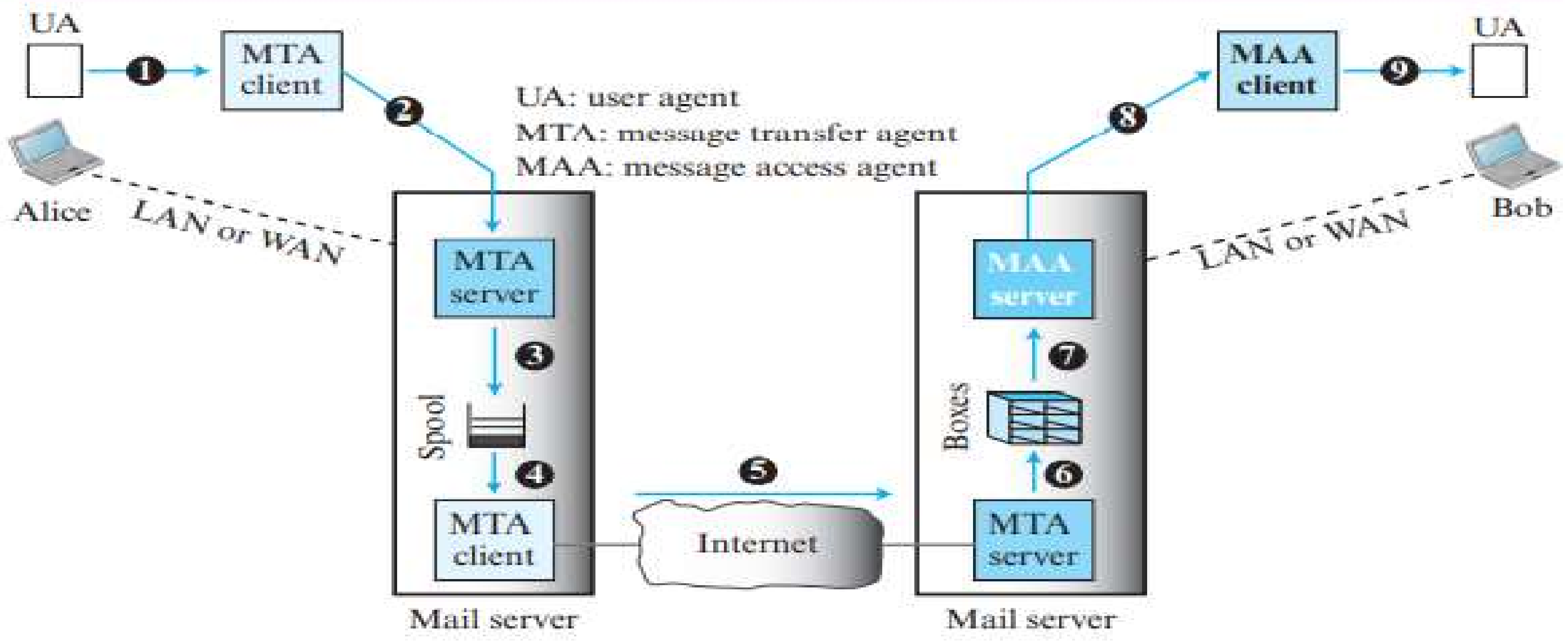
- In the common scenario, the sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers.
- The administrator has created one mailbox for each user where the received messages are stored.
- A mailbox is part of a server hard drive, a special file with permission restrictions.
- Only the owner of the mailbox has access to it.
- The administrator has also created a queue (spool) to store messages waiting to be sent.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Architecture

- A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure.
- Alice and Bob use three different agents: a User Agent (UA), a Mail Transfer Agent (MTA) and a Message Access Agent (MAA).

Figure 1.45 *Common scenario*



[Behrouz A Forouzan, Firouz Mosharraf, “Computer Networks: A top down Approach”, McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Architecture

- When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server.
- The mail server at her site uses a queue (spool) to store messages waiting to be sent.
- The message, however, needs to be sent through the Internet from Alice’s site to Bob’s site using an MTA.
- Here two message transfer agents are needed: one client and one server.
- Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Architecture

- The MTA client can be triggered by the system when there is a message in the queue to be sent.
- The user agent at the Bob site allows Bob to read the received message.
- Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.
- There are two important points we need to emphasize. First, Bob cannot bypass the mail server and use the MTA server directly
- Second, Bob needs another pair of client-server programs: message access programs.
- This is because an MTA client-server program is a push program: the client pushes the message to the server. Bob needs a pull program. The client needs to pull the message from the server.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

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

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

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.
- 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.
- There are two types of user agents: command-driven and GUI-based.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

User Agent

- Command driven user agents belong to the early days of electronic mail. A command-driven user agent normally accepts a one-character command from the keyboard to perform its task. Some examples of command driven user agents are mail, pine, and elm.

- Modern user agents are 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. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora and Outlook

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

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.
- The header of the message defines the sender, the receiver, the subject of the message, and some other information.
- The body of the message contains the actual information to be read by the recipient.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

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.
- The summary usually includes the sender mail address, the subject, and the time the mail was sent or received.
- The user can select any of the messages and display its contents on the screen.

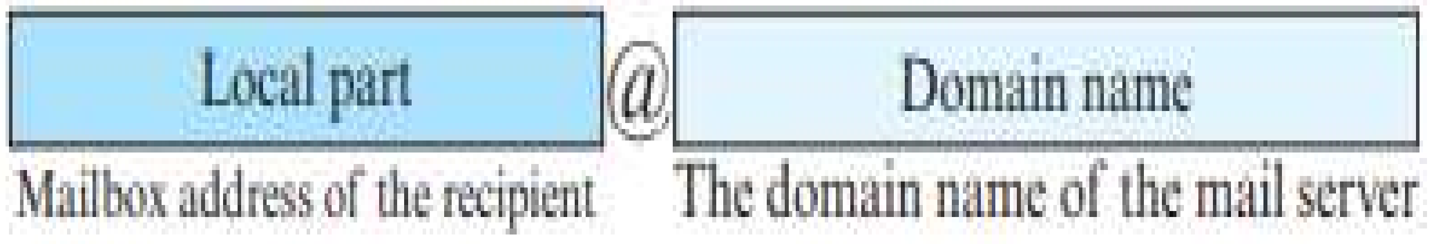
STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Addresses

- To deliver mail, a mail handling system must use an addressing system with unique addresses.
- In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign (see Figure 1.46).

Figure 1.46 *E-mail address*


Local part
@
Domain name
Mailbox address of the recipient
The domain name of the mail server

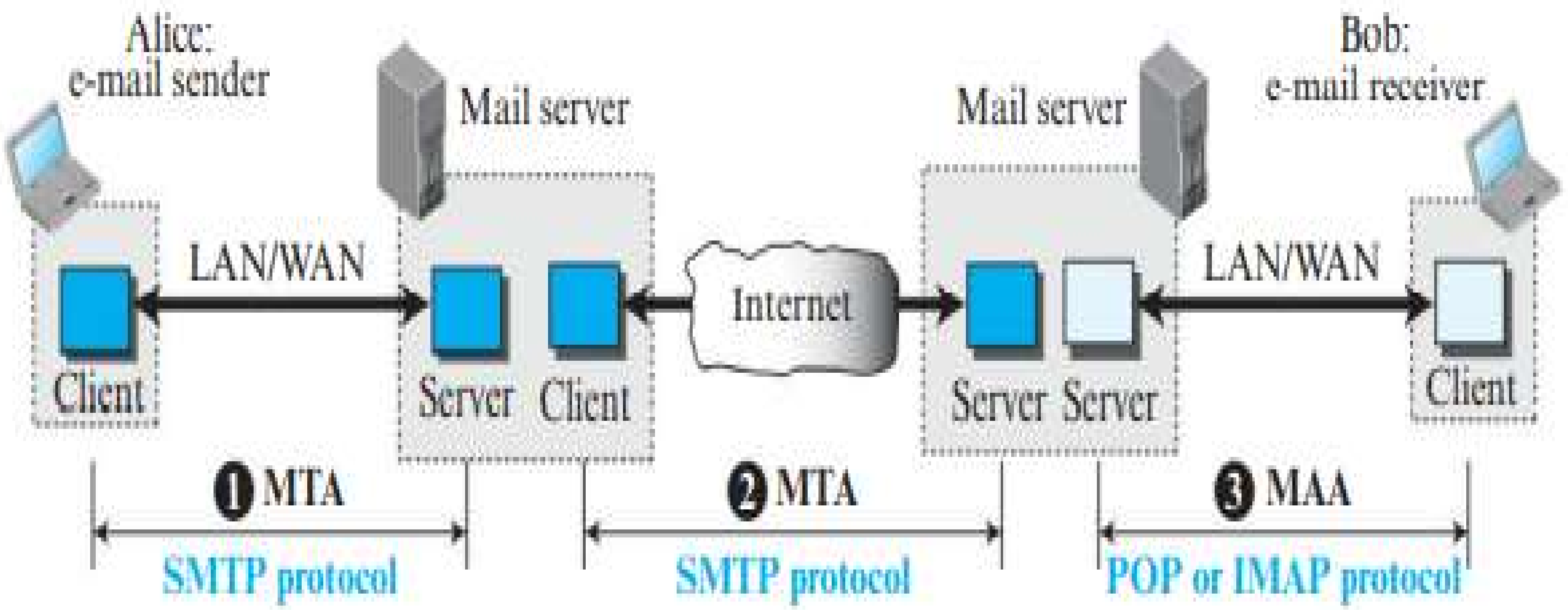
[Behrouz A Forouzan, Firouz Mosharraf, “Computer Networks: A top down Approach”, McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

BACK

Figure 1.47 *Protocols used in electronic mail*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

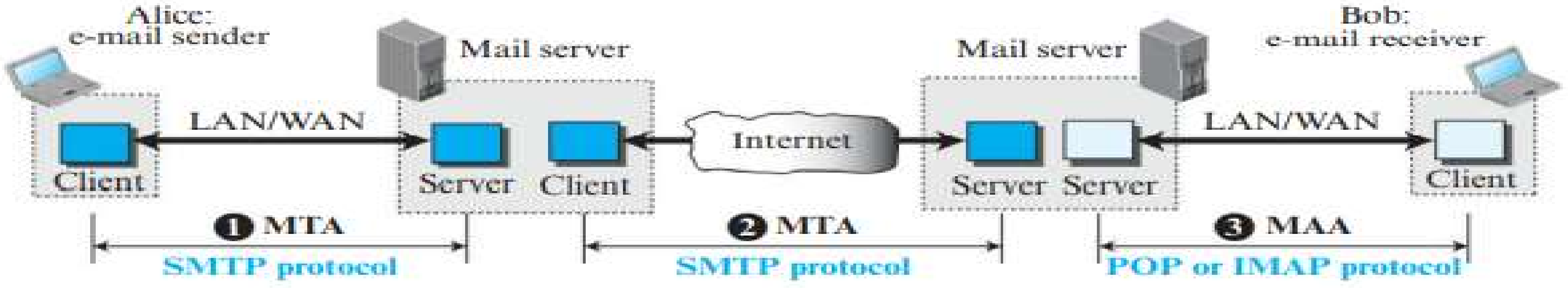
STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

- The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP).
- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.
- Another protocol(POP/IMAP) is needed between the mail server and the receiver.
- SMTP simply defines how commands and responses must be sent back and forth.

Figure 1.47 *Protocols used in electronic mail*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

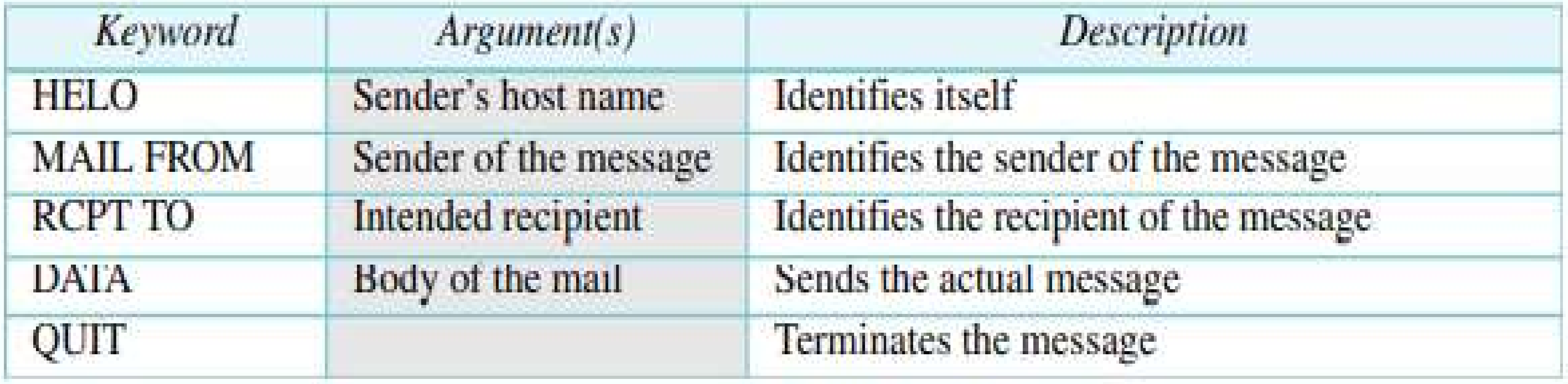
Commands and Responses

- 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.
- Commands are sent from the client to the server.
- The format of a command is shown below:
    - Keyword: argument(s)
    - consists of a keyword followed by zero or more arguments.
- SMTP defines 14 commands, listed in Table 1.7.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Table 1.7 *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

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail Message Transfer Agent: 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.
- Table 1.8 shows the most common response types.

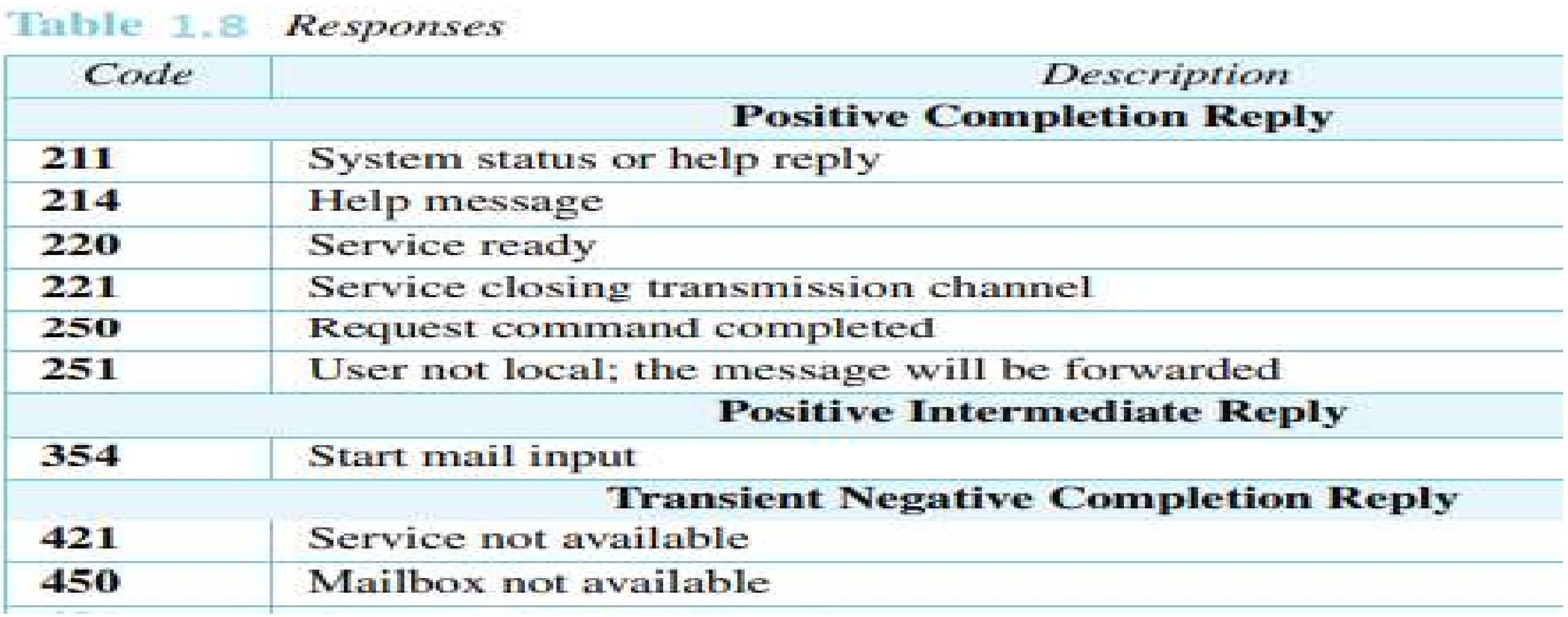


Table 1.8 *Responses*

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

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

Mail Transfer Phases

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

Message Transfer Agent: SMTP

Connection Establishment

- After a client has made a TCP connection to the well known port 25, the SMTP server starts the connection phase. [port numbers are 25, 465, 587, and 2525. 587 and 2525 are the more reliable options]
- This phase involves 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.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

Mail Transfer Phases

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. Steps 3 and 4 are repeated if there is more than one recipient.
  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.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

Mail Transfer Phases

Message Transfer

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(A dot).
8. The server responds with code 250 (OK) or some other appropriate code.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail Message Transfer Agent: SMTP

Mail Transfer Phases

Connection Termination

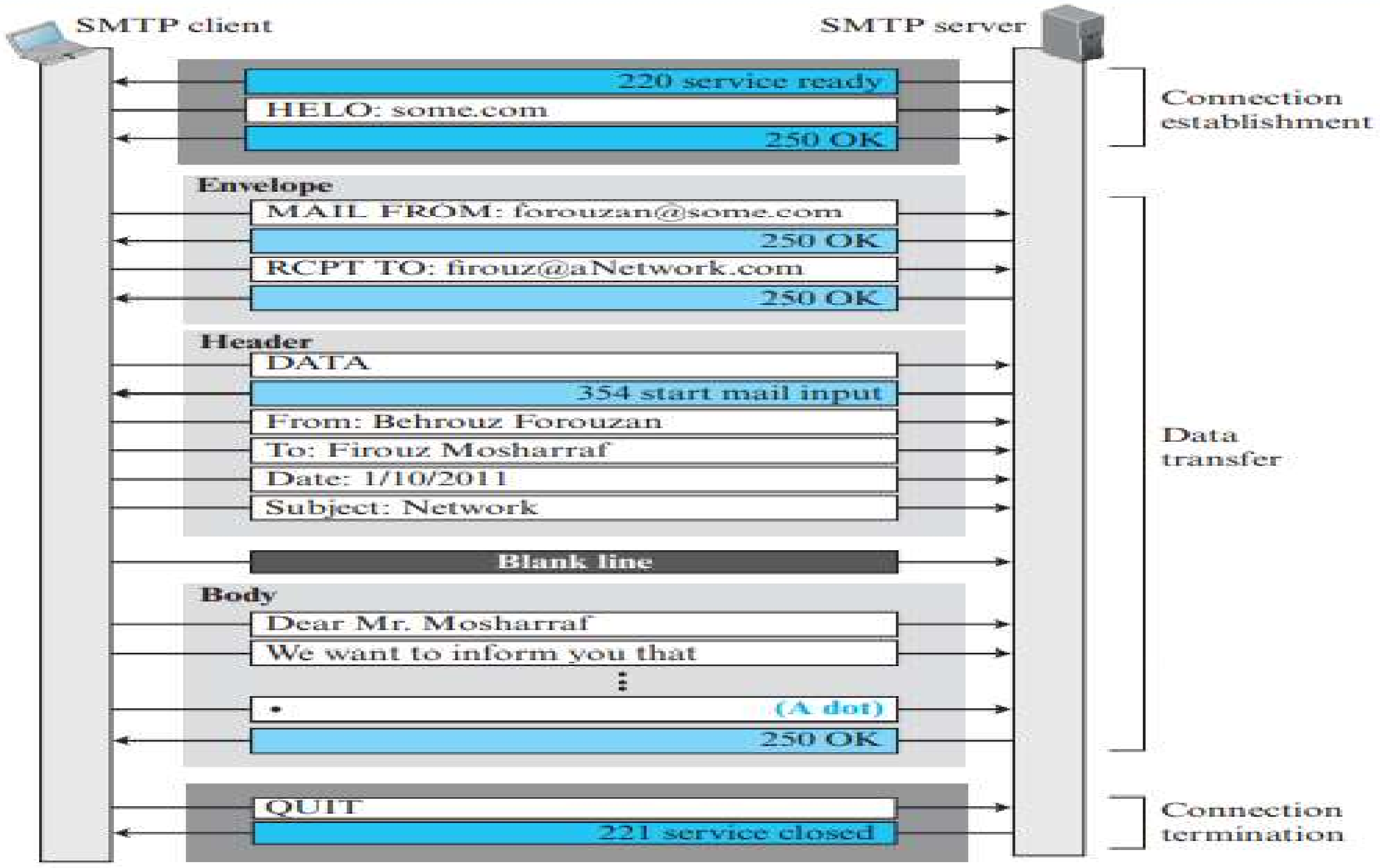
- After the message is transferred successfully, the client terminates the connection.
- This phase involves two steps.
  1. The client sends the QUIT command.
  2. The server responds with code 221 (service closing transmission channel) or some other appropriate code

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

Message Transfer Agent: SMTP

Figure 1.48 *Example*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

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).
- Figure 1.47 shows the position of these two protocols

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

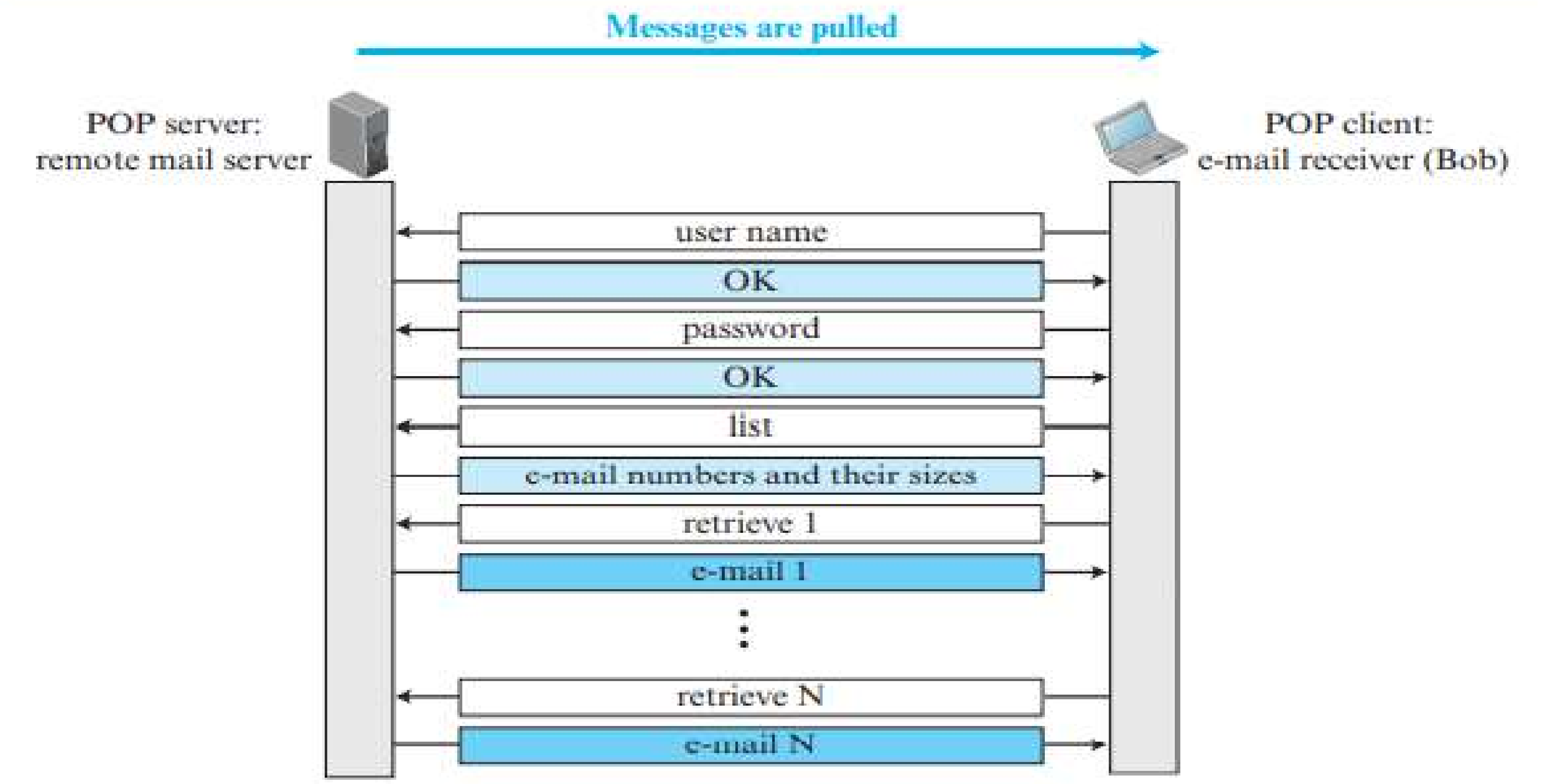
POP3

- Post Office Protocol, version 3 (POP3) is simple but limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download its e-mail from the mailbox 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.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail POP3

Figure 1.49 *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

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail — POP3

- 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.
- In the keep mode, the mail remains in the mailbox after 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.
- 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.
- Internet Mail Access Protocol, version 4 (IMAP4)
- Multipurpose Internet Mail Extensions (MIME)