Remote Logging, Electronic Mail, and File Transfer

REMOTE LOGGING

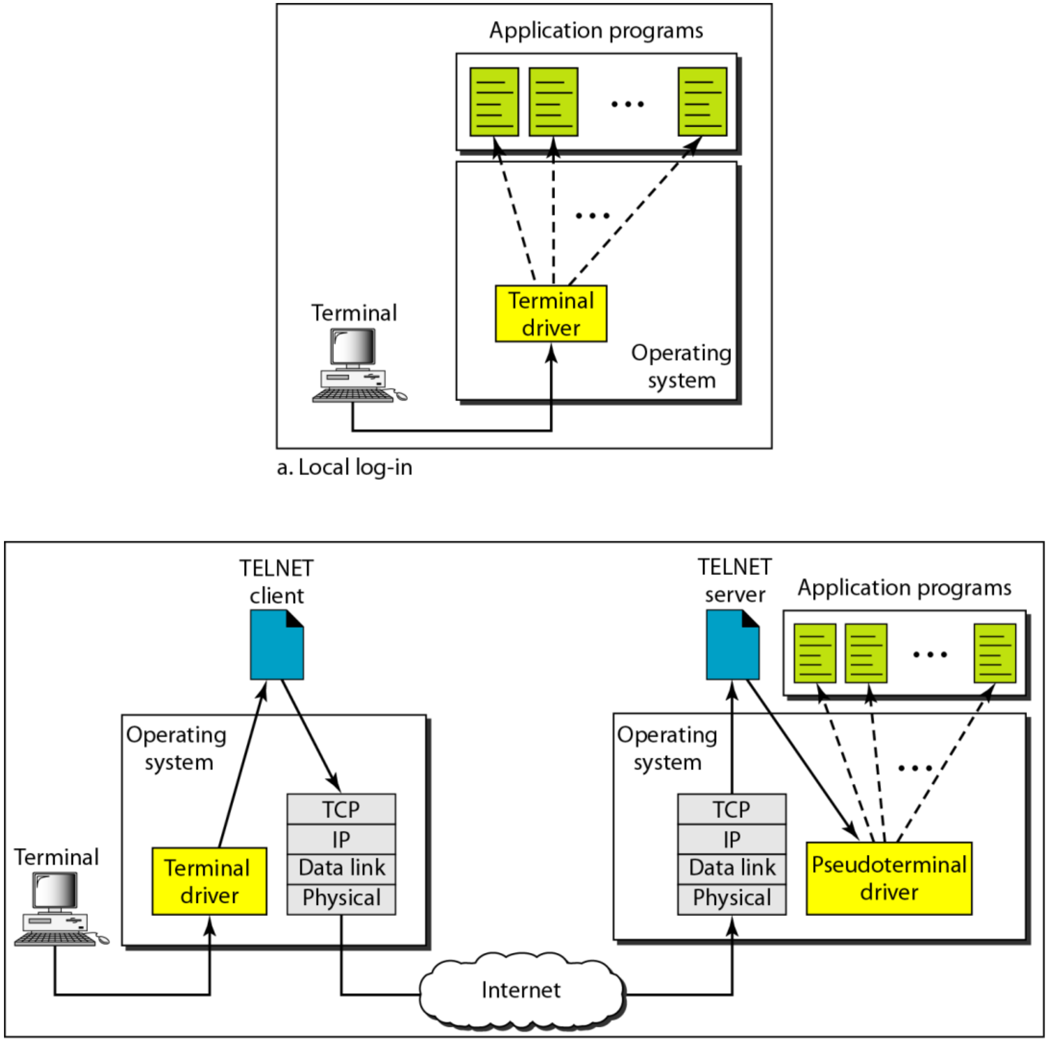
It would be impossible to write a specific client/server program for each demand. The better solution is a general-purpose client/server program that lets a user access any application program on a remote computer.

Topics discussed in this section:

TELNET

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

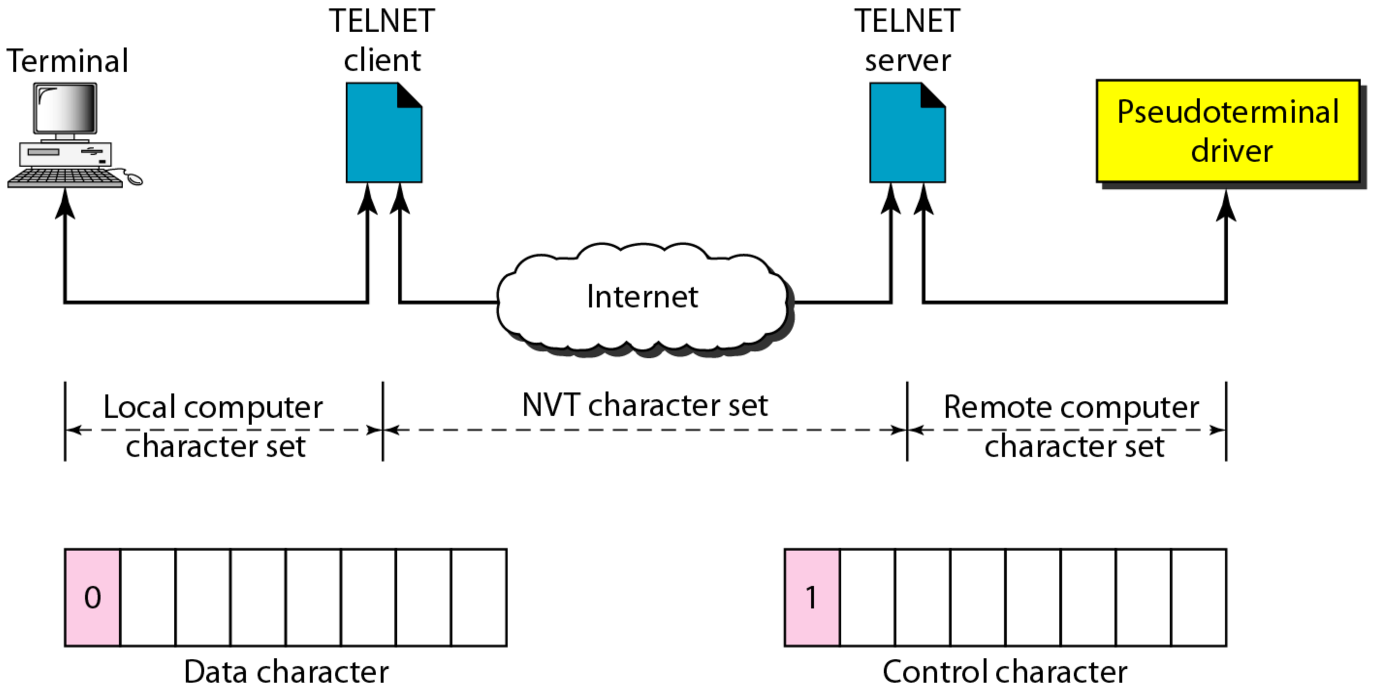
Figure *Local and remote log-in*



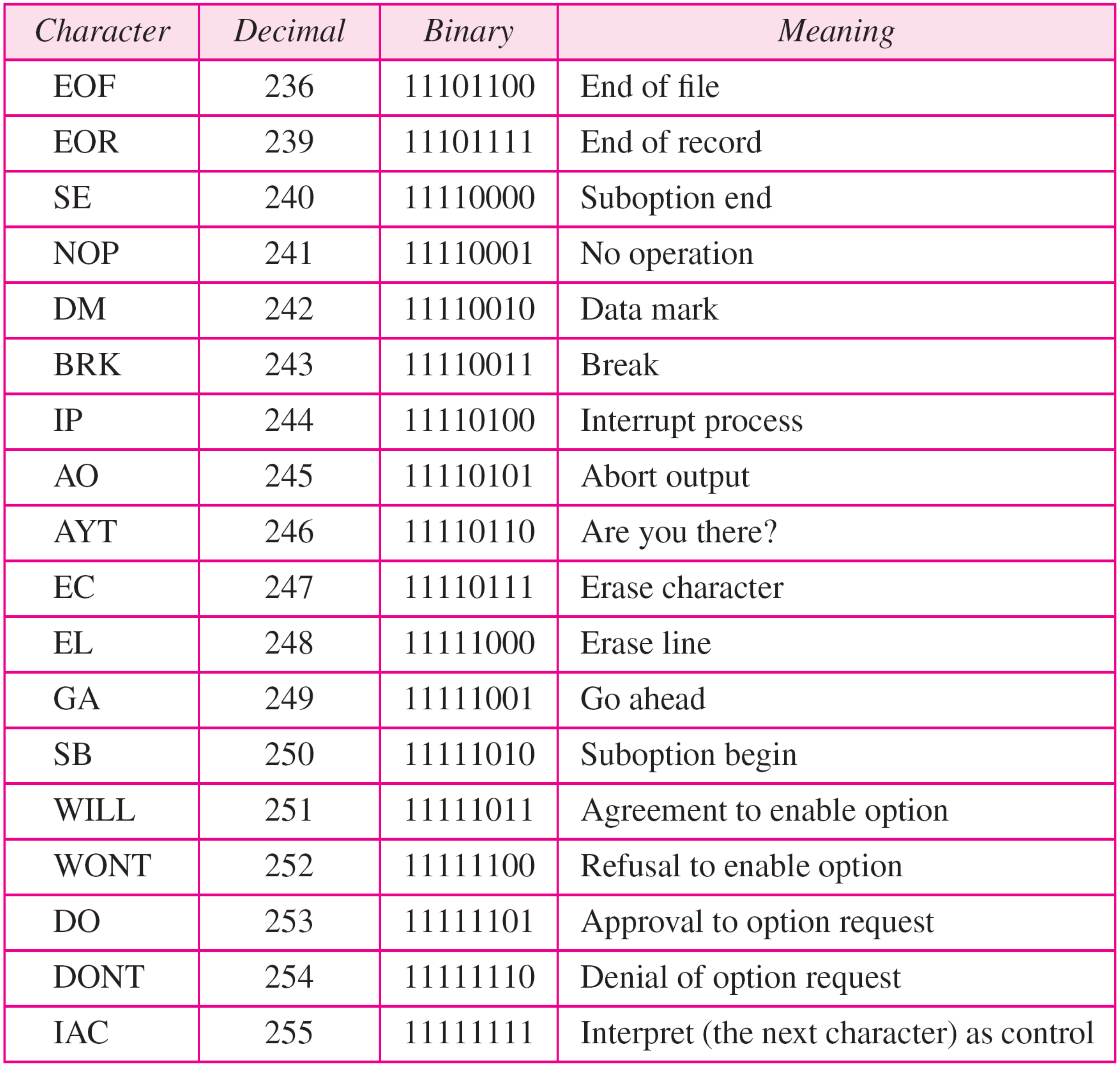
a. Local log-in

b. Remote log-in

Figure *Concept of NVT*

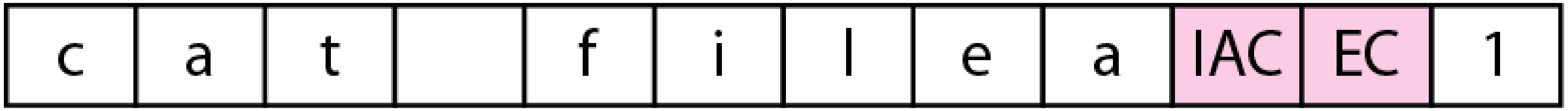


Some NVT control characters



Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

Figure *An example of embedding*



c	a	t		f	i	l	e	a	IAC	EC	1

Typed at the remote terminal

Table *Options*

Table ***NVT character set for option negotiation***

Figure *Example : Echo option*

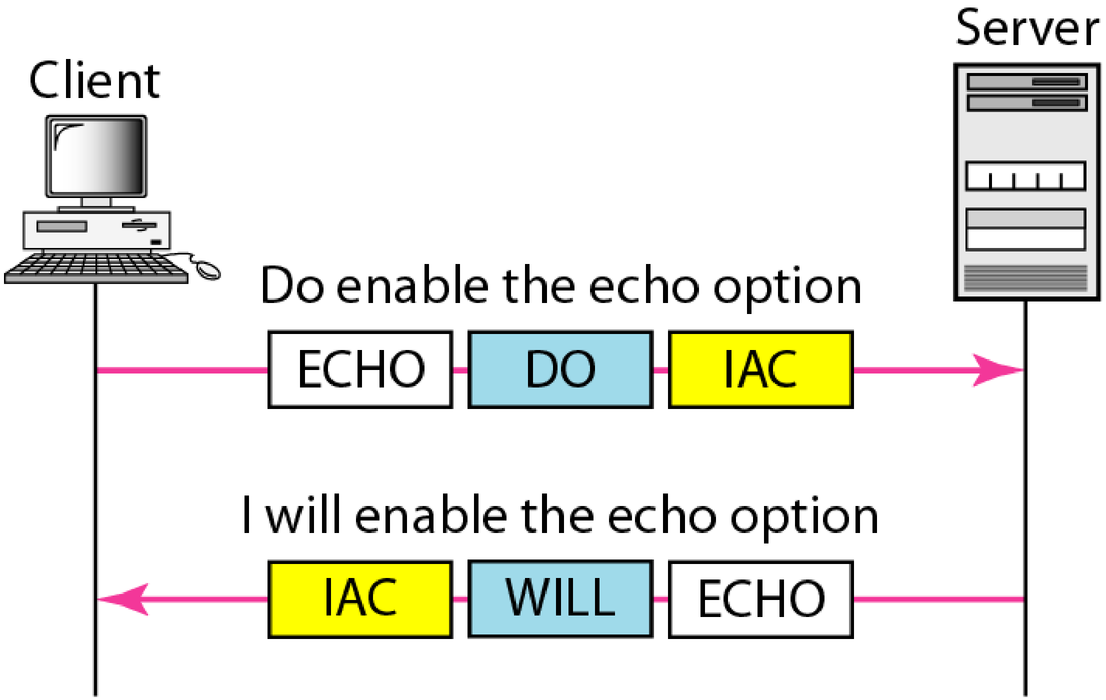
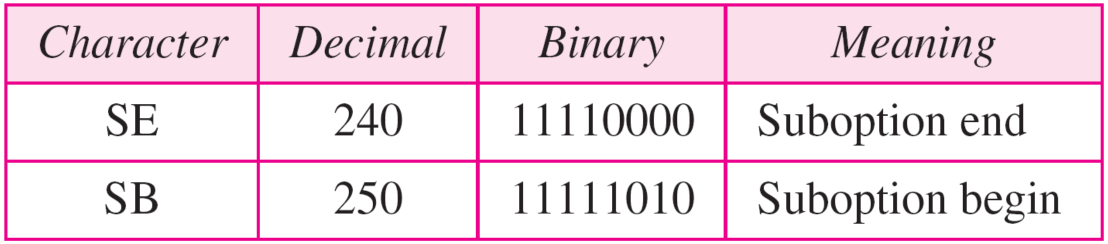


Table *Character set for suboptions*



Character	*Decimal*	*Binary*	*Meaning*
SE	240	11110000	Suboption end
SB	250	11111010	Suboption begin

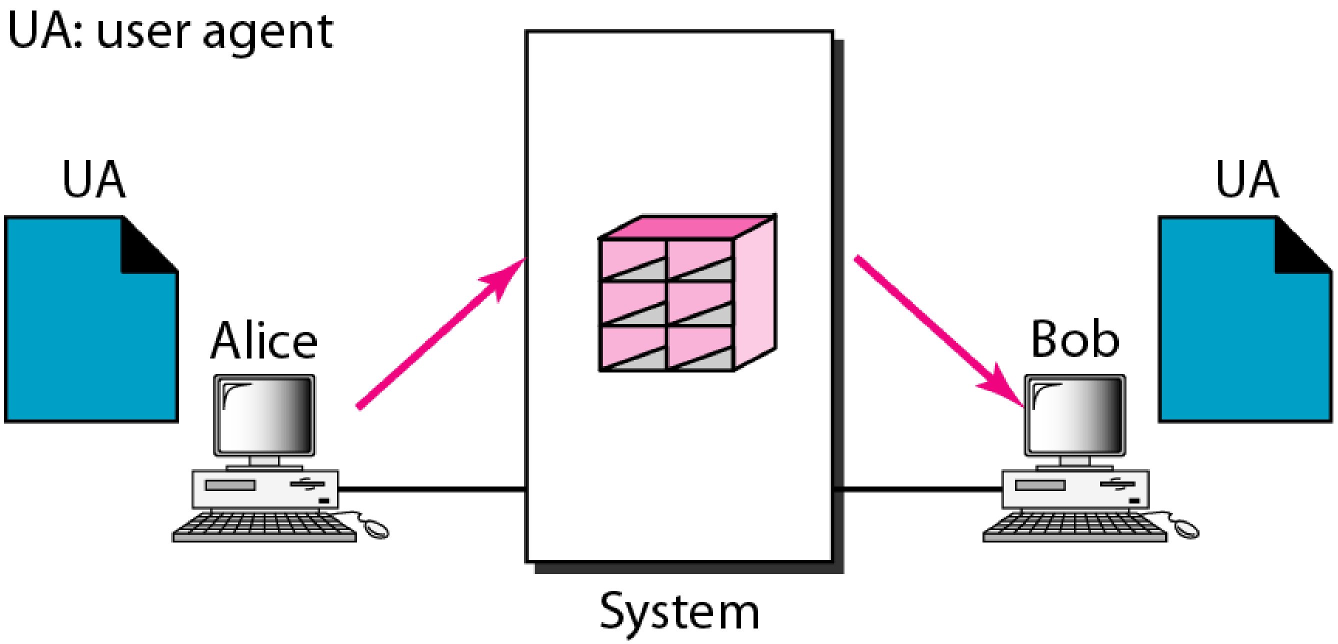
ELECTRONIC MAIL

One of the most popular Internet services is electronic mail (e-mail). The designers of the Internet probably never imagined the popularity of this application program. Its architecture consists of several components that we discuss in this chapter.

Topics discussed in this section:

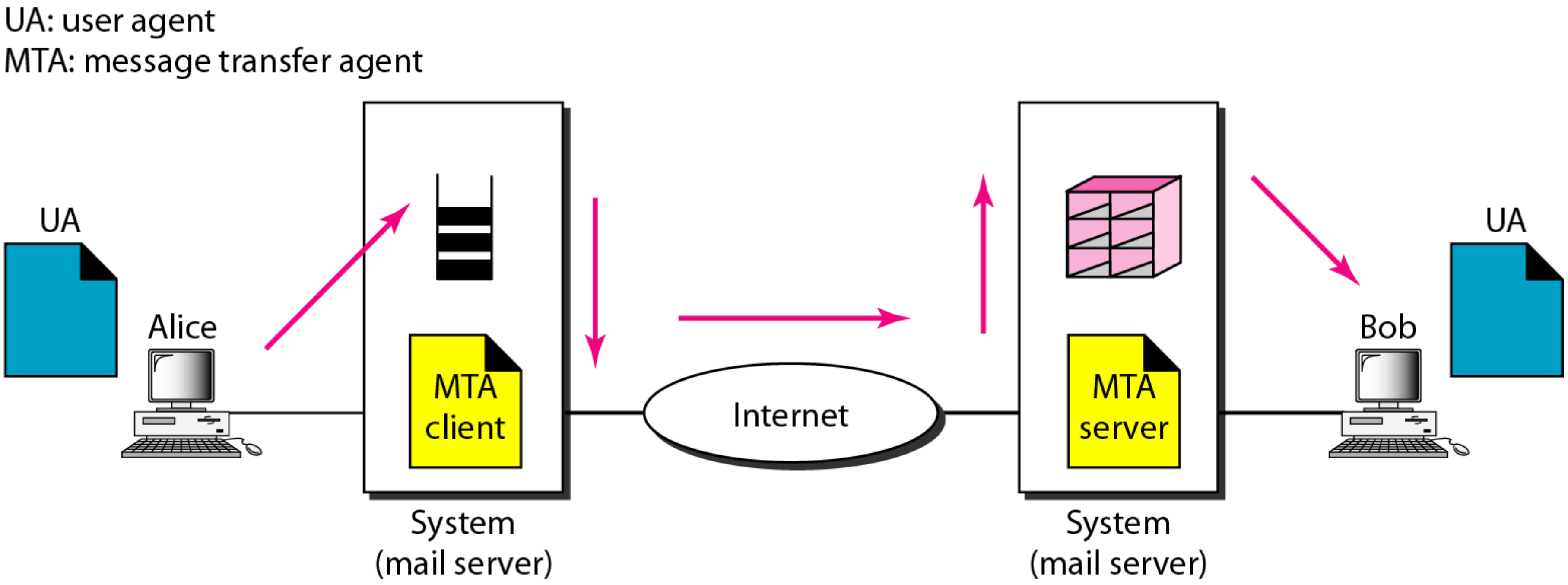
Architecture
User Agent
Message Transfer Agent: SMTP
Message Access Agent: POP and IMAP
Web-Based Mail

Figure *First scenario in electronic mail*



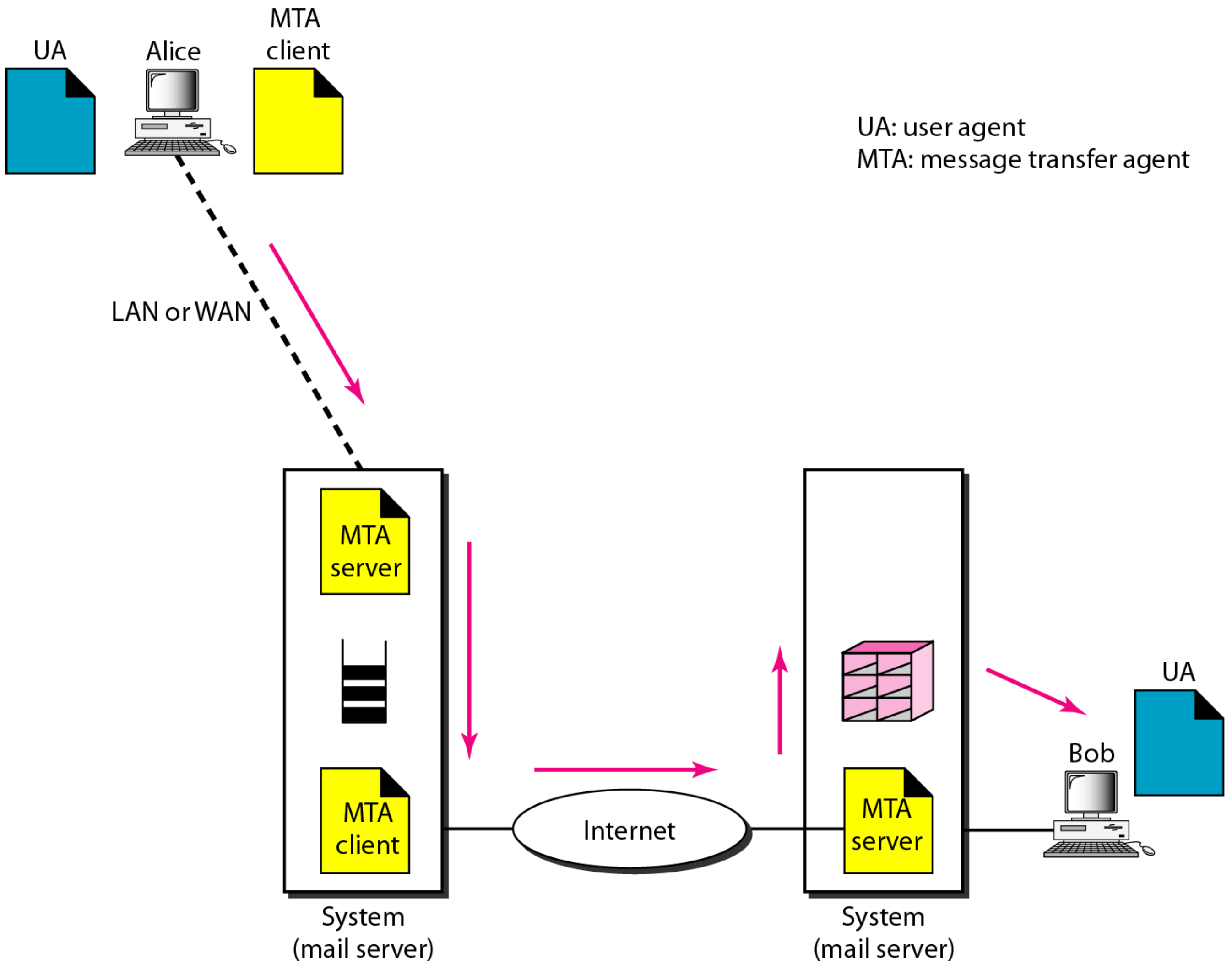
**When the sender and the receiver of an e-mail are on the same system,
we need only two user agents.**

Figure *Second scenario in electronic mail*



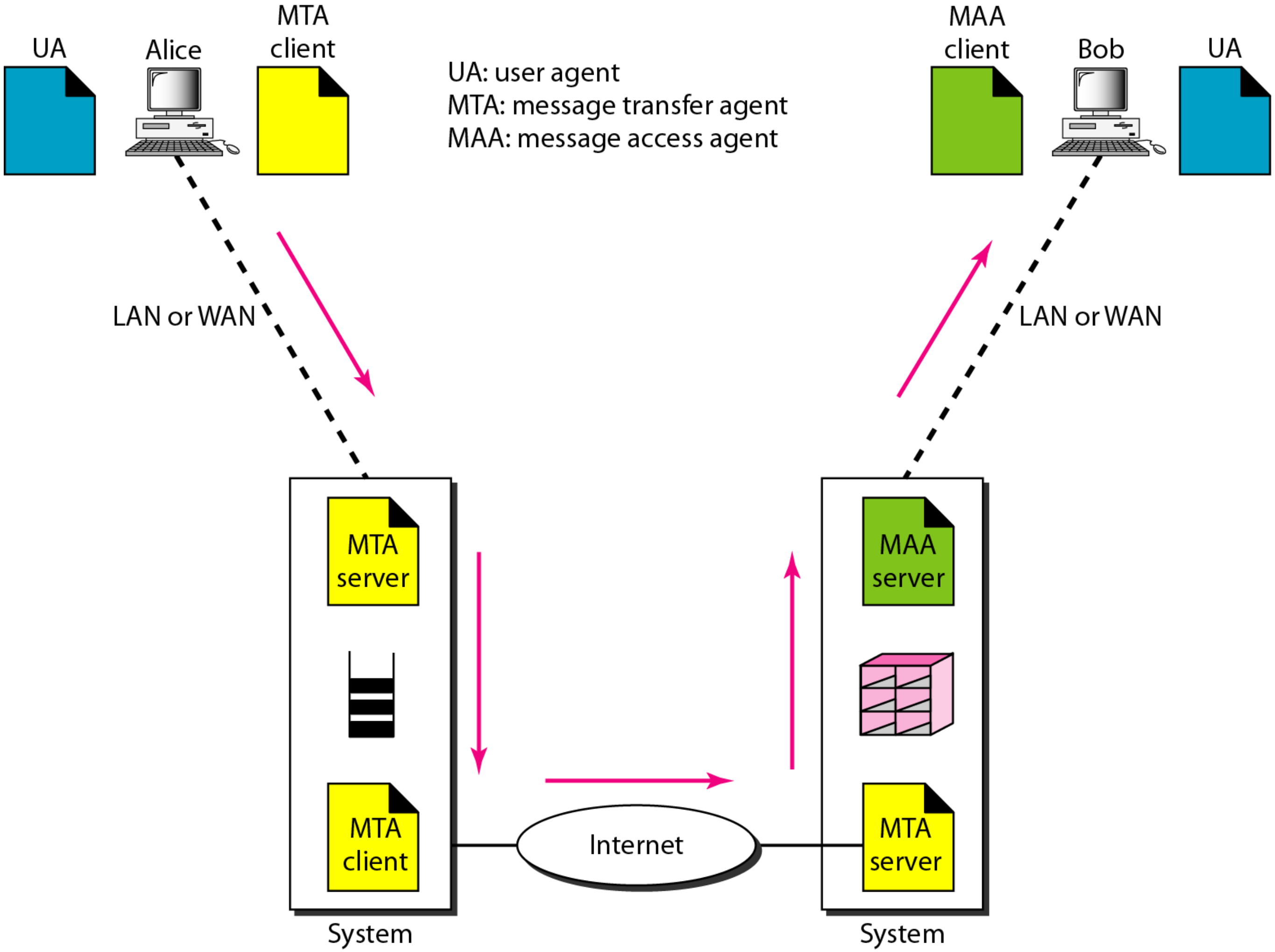
When the sender and the receiver of an e-mail are on different systems, we need two UAs and a pair of MTAs (client and server).

Figure *Third scenario in electronic mail*



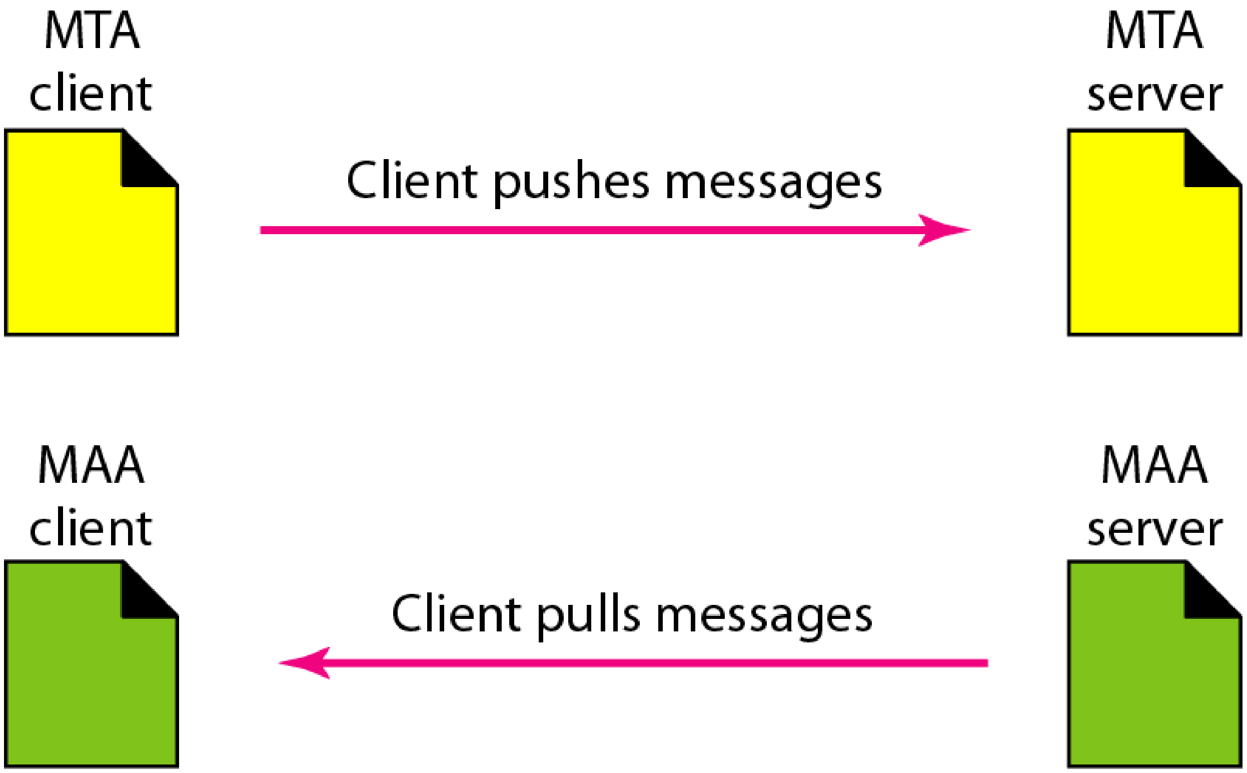
When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).

Figure *Fourth scenario in electronic mail*



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

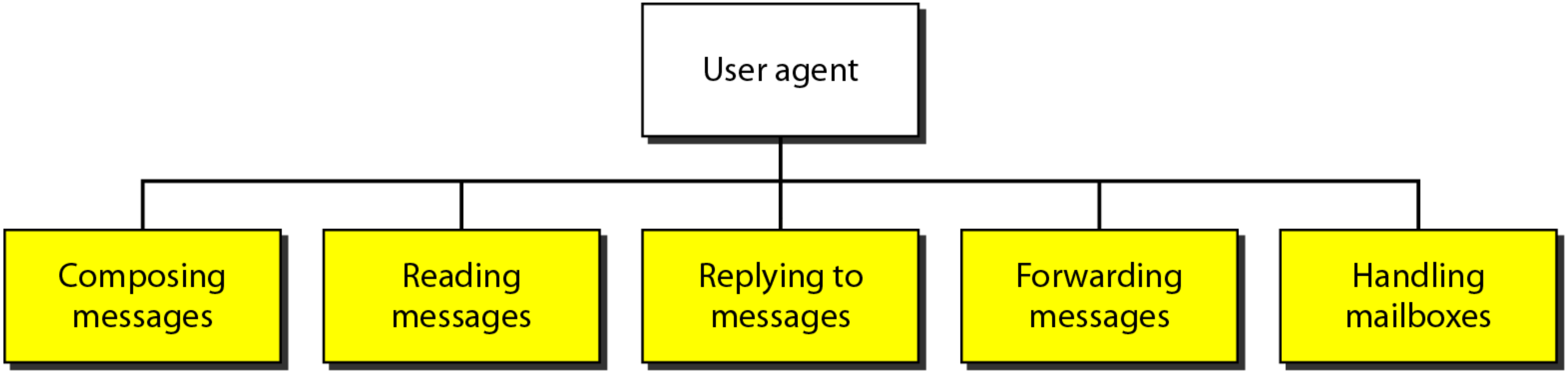
Figure *Push versus pull in electronic email*



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 and a pair of MAAs.

This is the most common situation today.

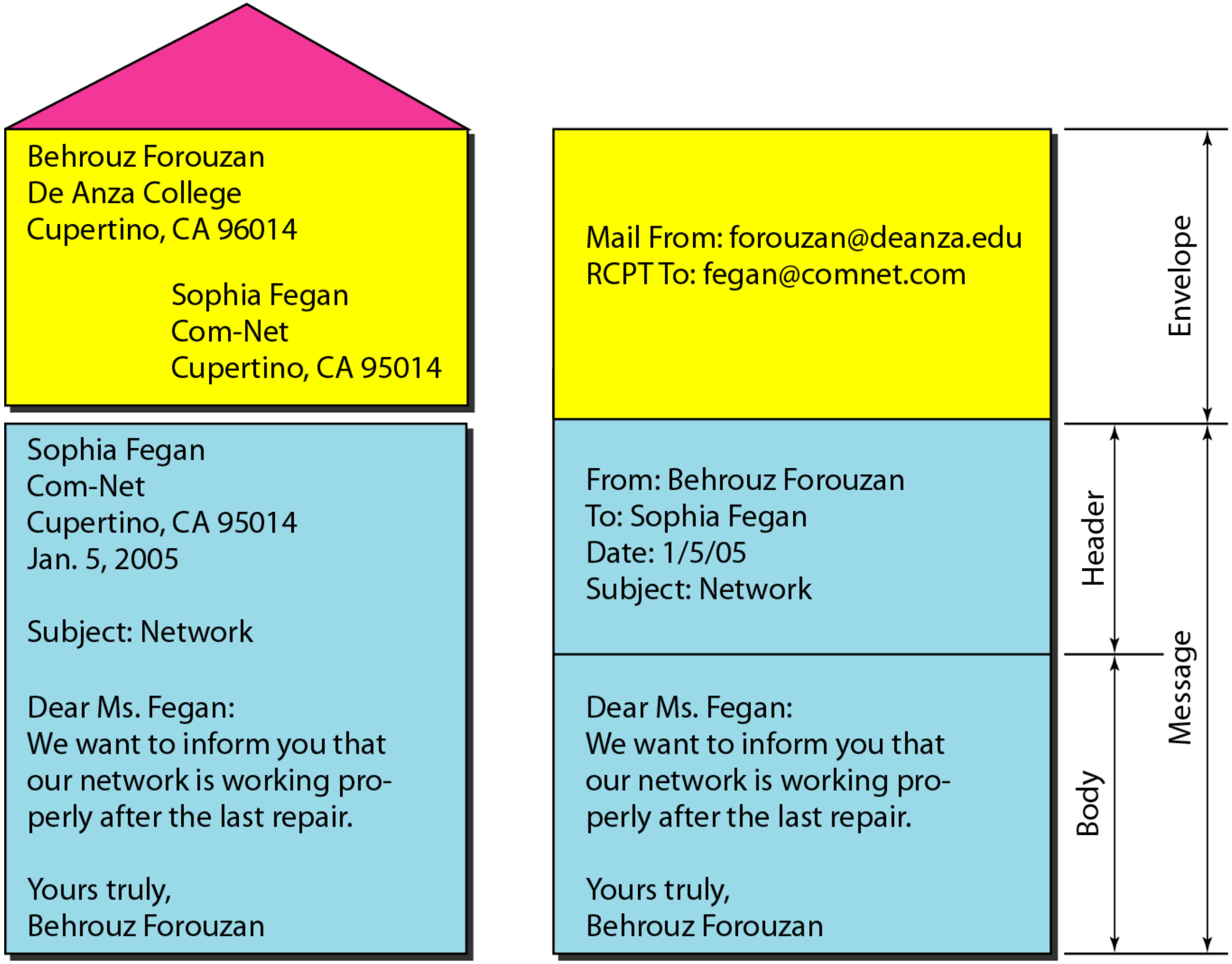
Figure *Services of user agent*



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

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

Figure *Format of an e-mail*



Behrouz Forouzan
De Anza College
Cupertino, CA 96014

Sophia Fegan
Com-Net
Cupertino, CA 95014

Sophia Fegan
Com-Net
Cupertino, CA 95014
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:
We want to inform you that our network is working properly after the last repair.

Yours truly,
Behrouz Forouzan

a. Postal mail

Mail From: forouzan@deanza.edu
RCPT To: fegan@comnet.com

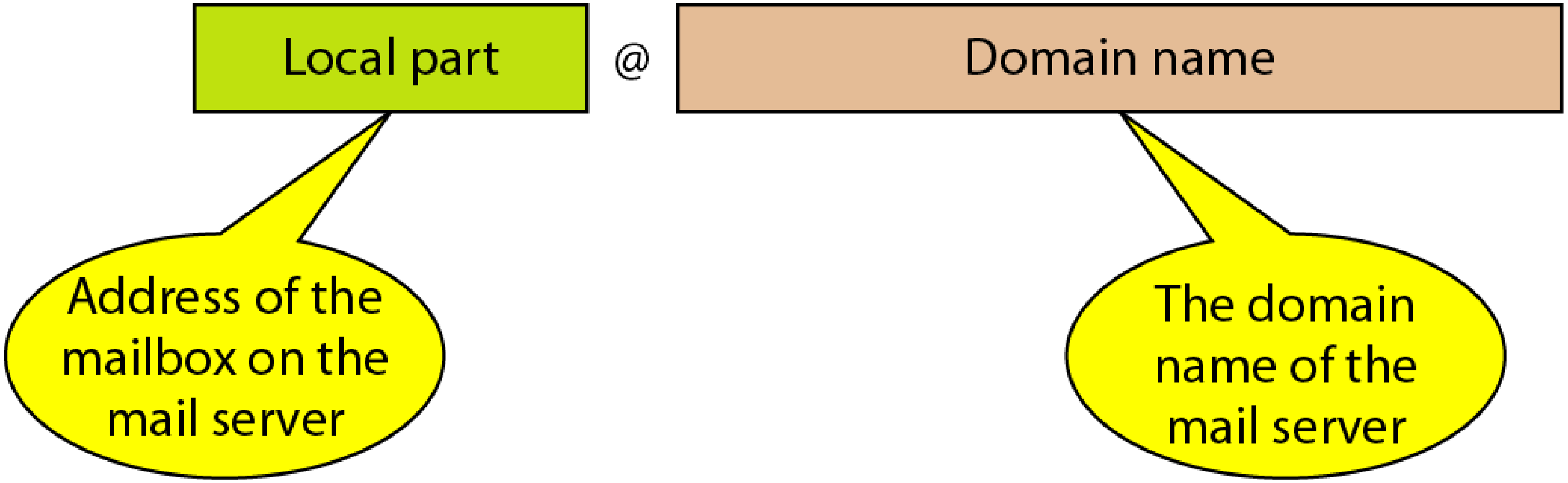
From: Behrouz Forouzan
To: Sophia Fegan
Date: 1/5/05
Subject: Network

Dear Ms. Fegan:
We want to inform you that our network is working properly after the last repair.

Yours truly,
Behrouz Forouzan

b. Electronic mail

Figure *E-mail address*


Local part
@
Domain name
Address of the mailbox on the mail server
The domain name of the mail server

Message body and MIME

- Early days, email was made up of English text messages expressed in ASCII

  - No support for foreign languages

  - No support for non text attachments (pdf, doc, jpg, audio files)

To solve this problem,

MIME (Multipurpose Internet mail extension) is used

- Additional headers
- Define content types and subtypes
- Add structure to message body

MIME Headers

Header	Meaning
MIME-Content	Identifies the MIME version
Content description	ASCII string that tells what is in the message
Content -ID	Unique identifier
Content Type	Type of data contained in the message
Content Transfer Encoding	How data in message is encoded

Figure *MIME*

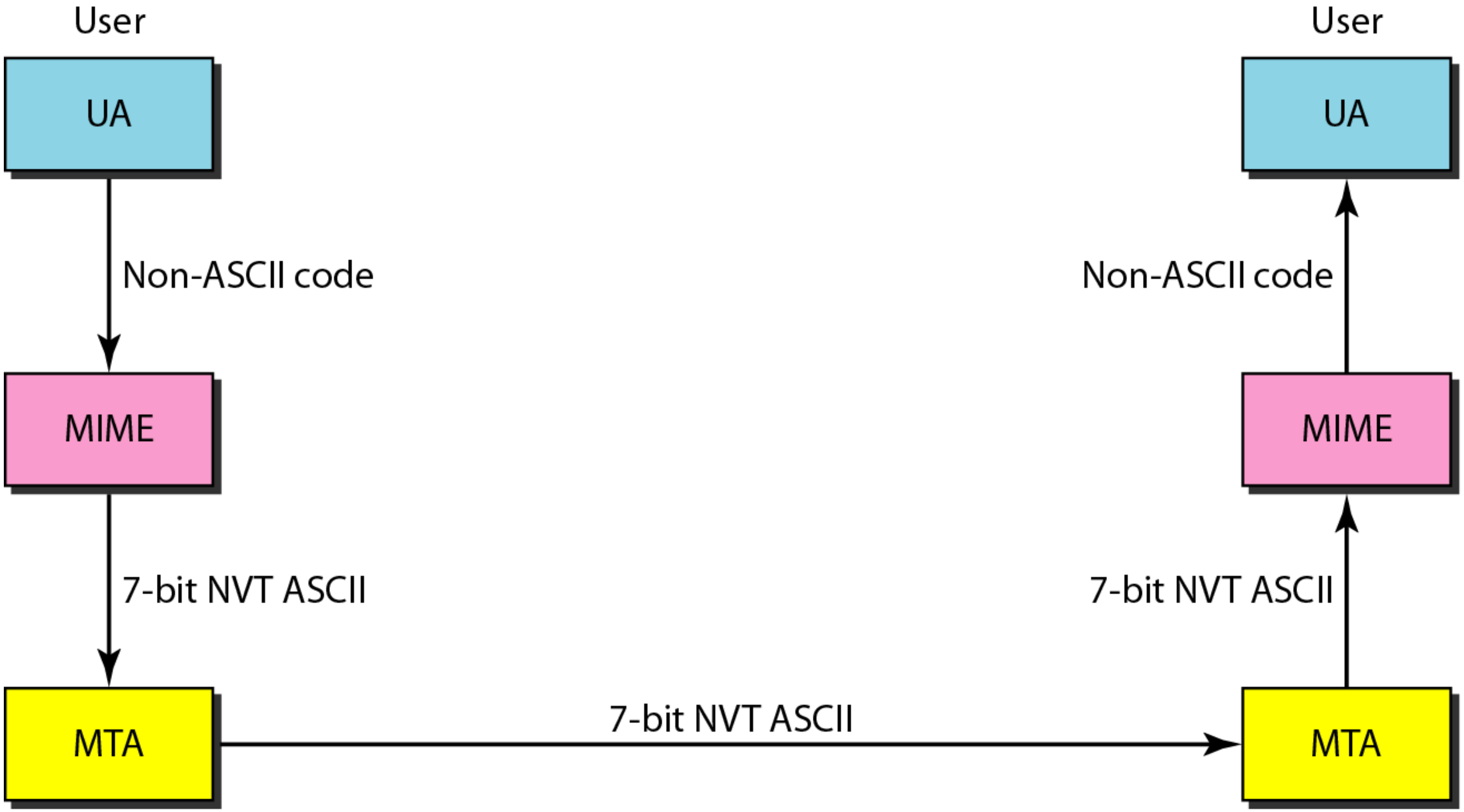
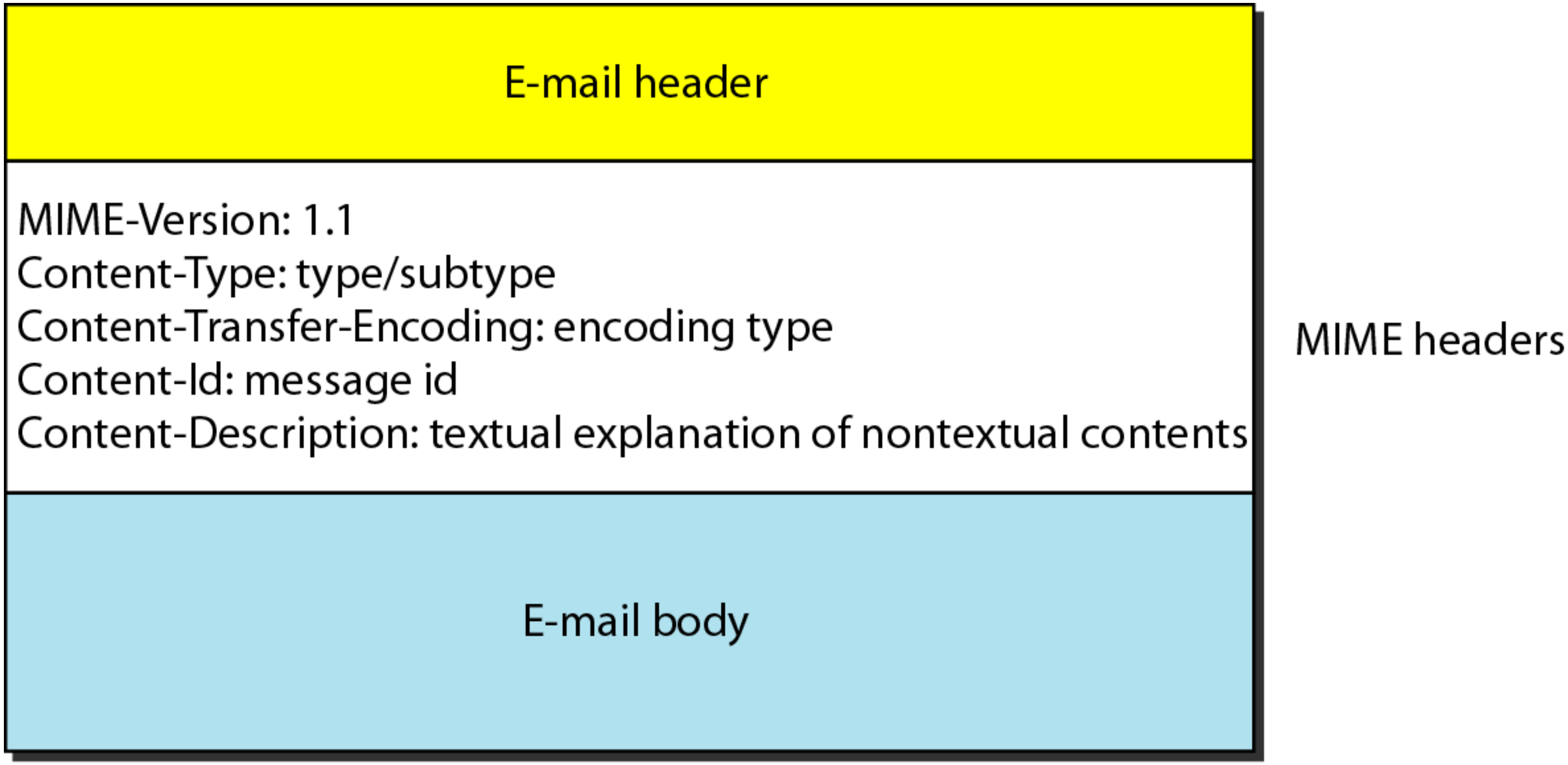

User
UA
Non-ASCII code
MIME
7-bit NVT ASCII
MTA
7-bit NVT ASCII
User
UA
Non-ASCII code
MIME
7-bit NVT ASCII
MTA

Figure *MIME header*



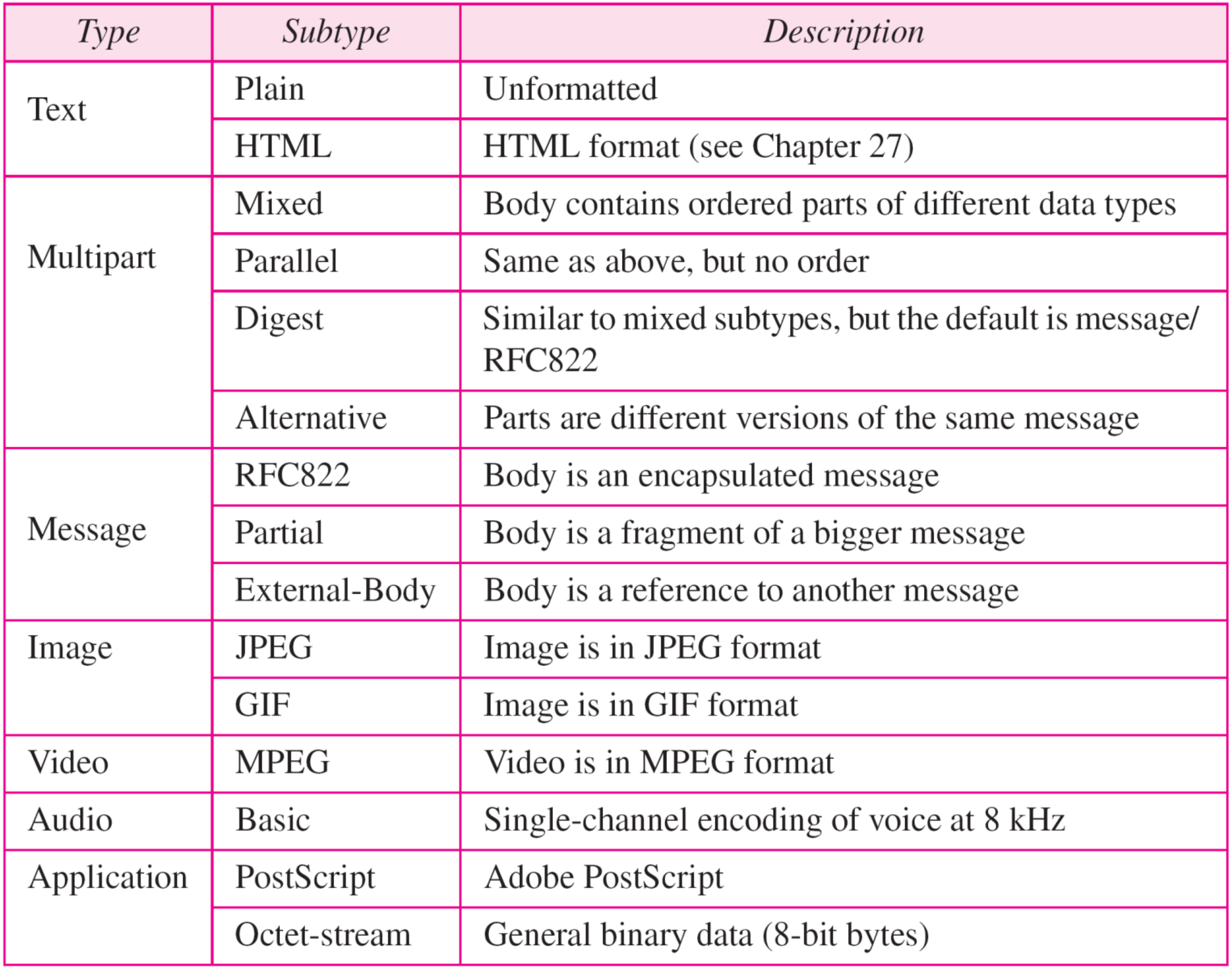
E-mail header

MIME-Version: 1.1
Content-Type: type/subtype
Content-Transfer-Encoding: encoding type
Content-Id: message id
Content-Description: textual explanation of nontextual contents

MIME headers

E-mail body

Table *Data types and subtypes in MIME*



Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Chapter 27)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed subtypes, 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 (8-bit bytes)

Table *Content-transfer-encoding*

Type	*Description*
7-bit	NVT ASCII characters and short lines
8-bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base-64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

SMTP (Simple mail transfer protocol)

- Uses TCP to reliably transfer email messages
- Operates on Port 25
- Three phases of transfer based on command

  - Handshaking

  - Transfer of messages

  - Close connection

Figure *SMTP range*

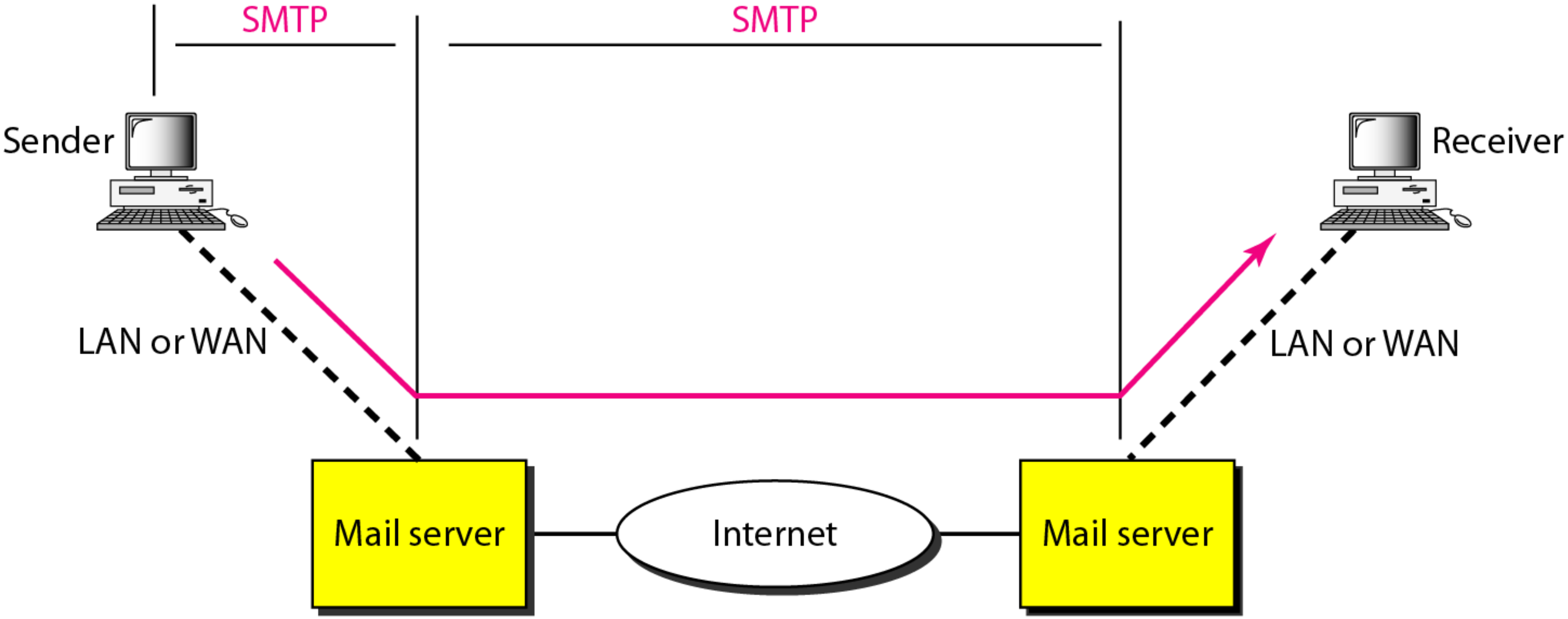


Figure *Commands and responses*

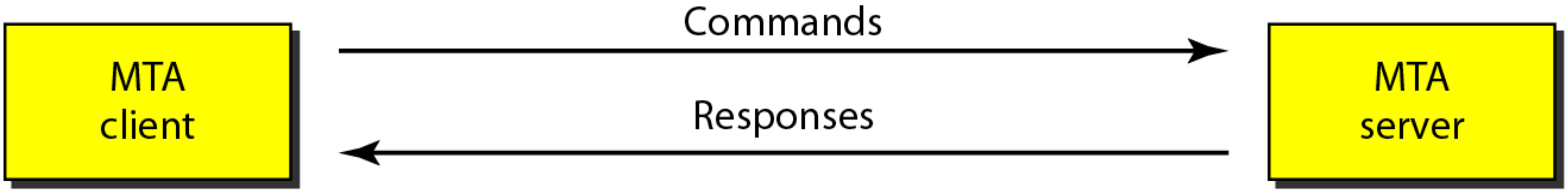
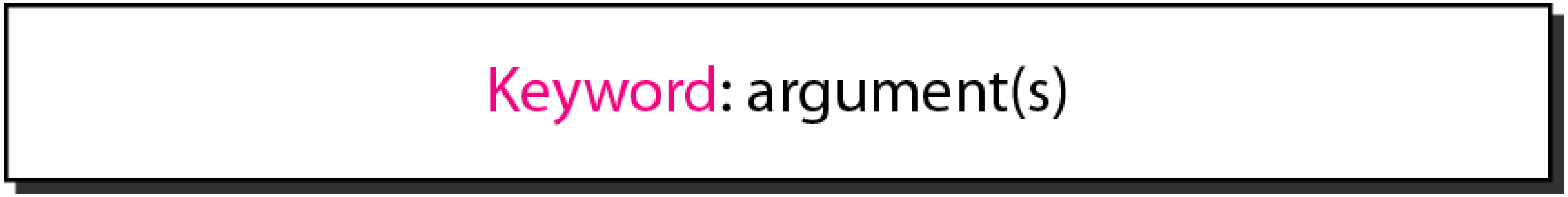
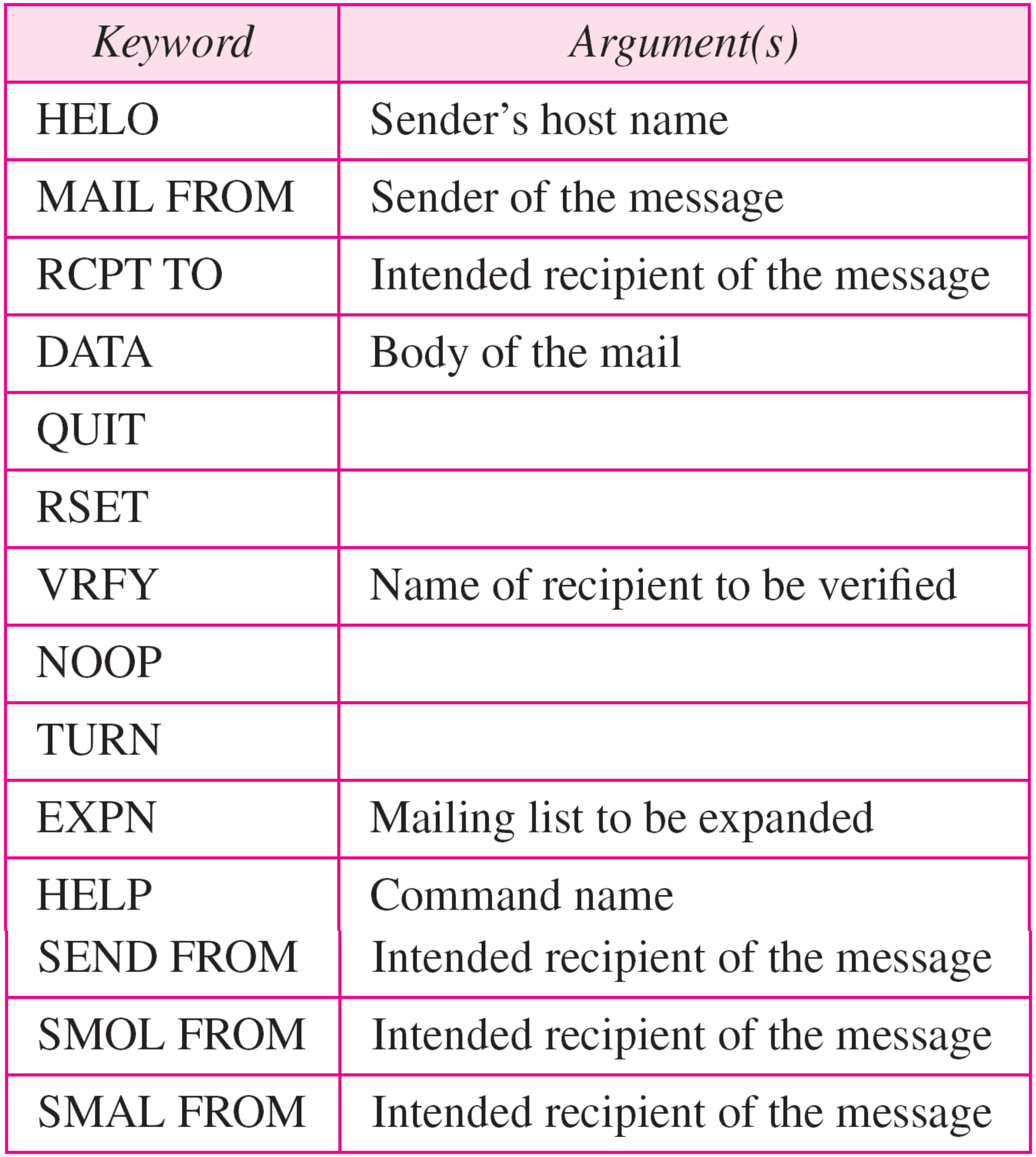


Figure *Command format*



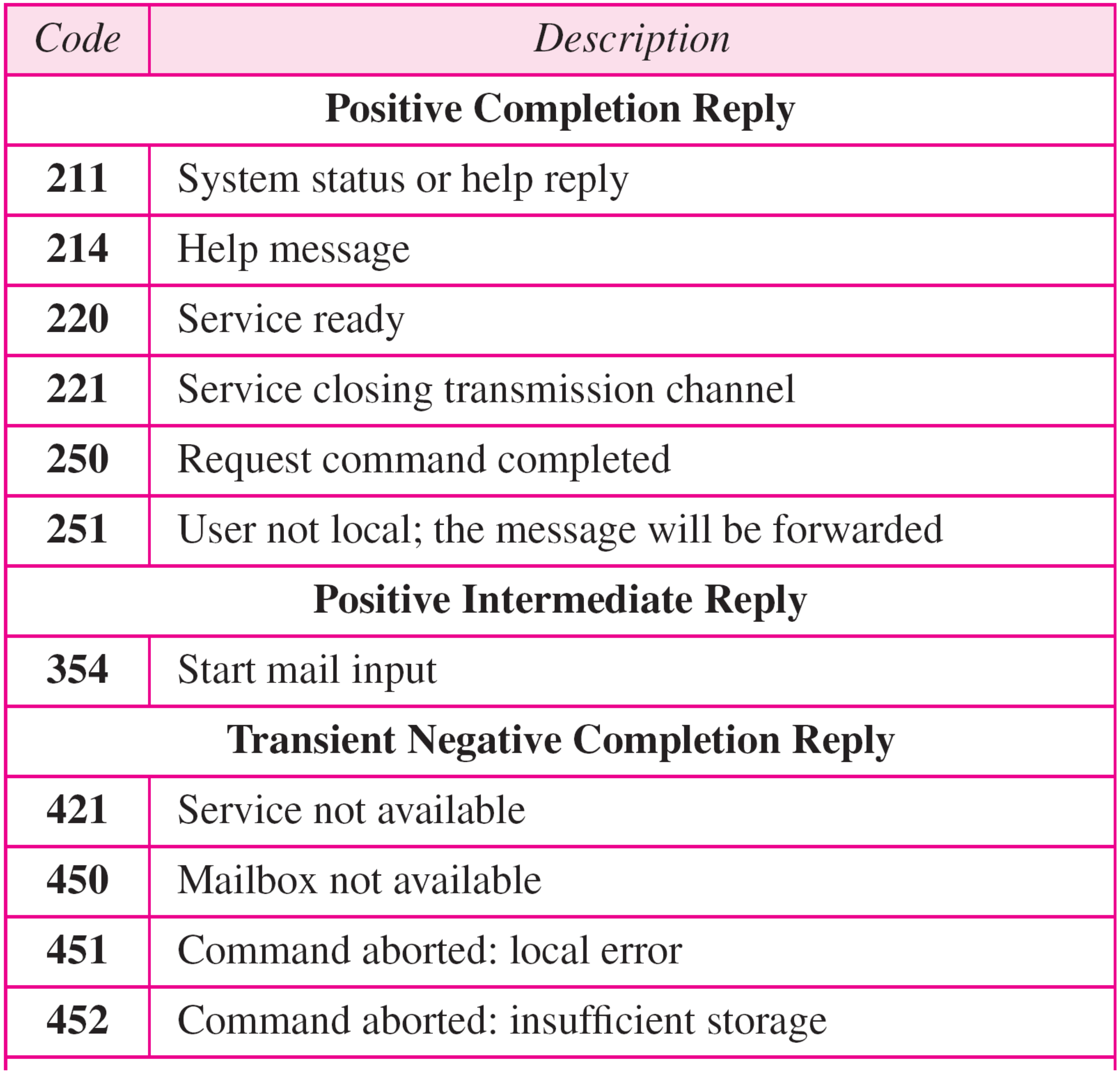
Keyword: argument(s)

Table *Commands*



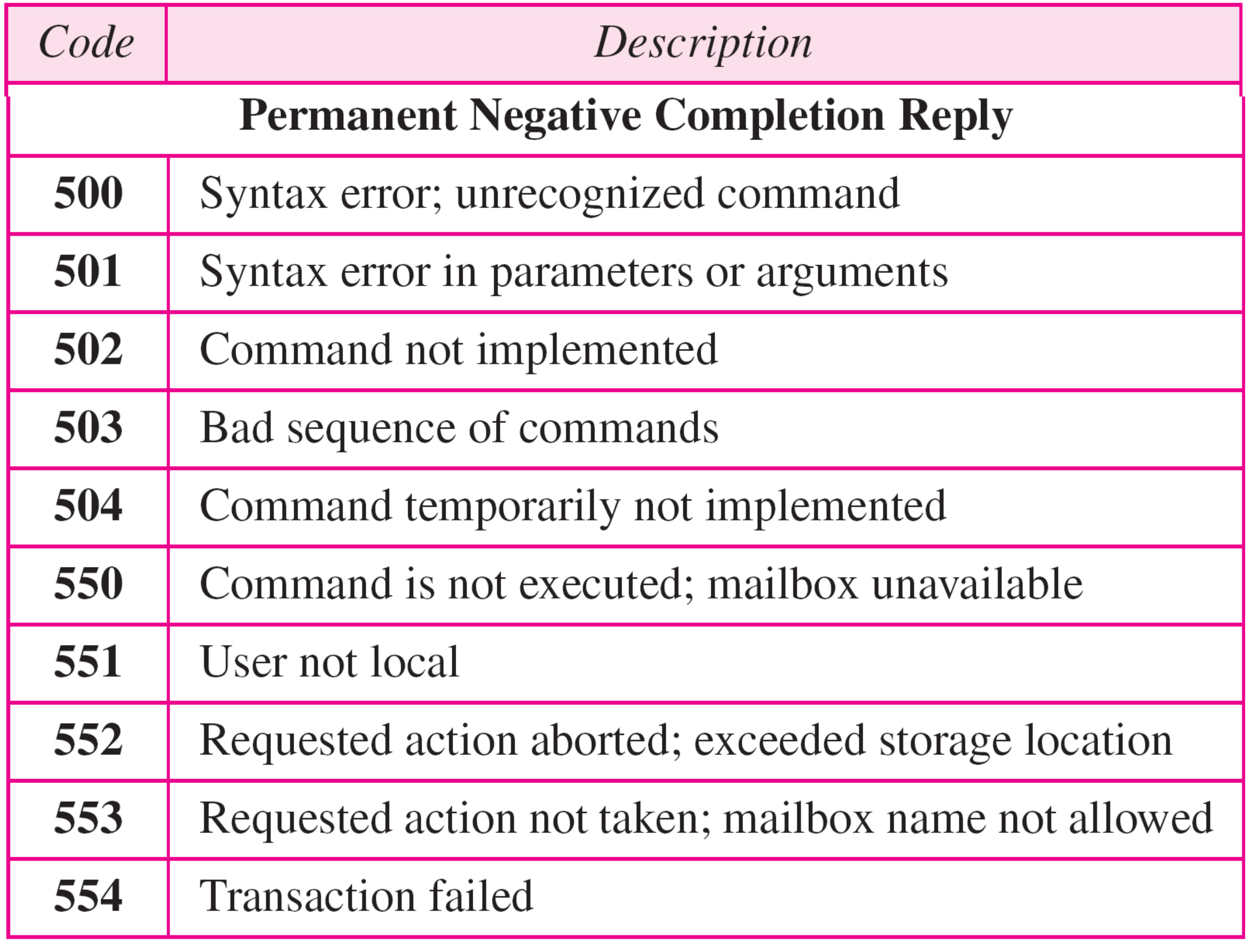
Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL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

Table *Responses*



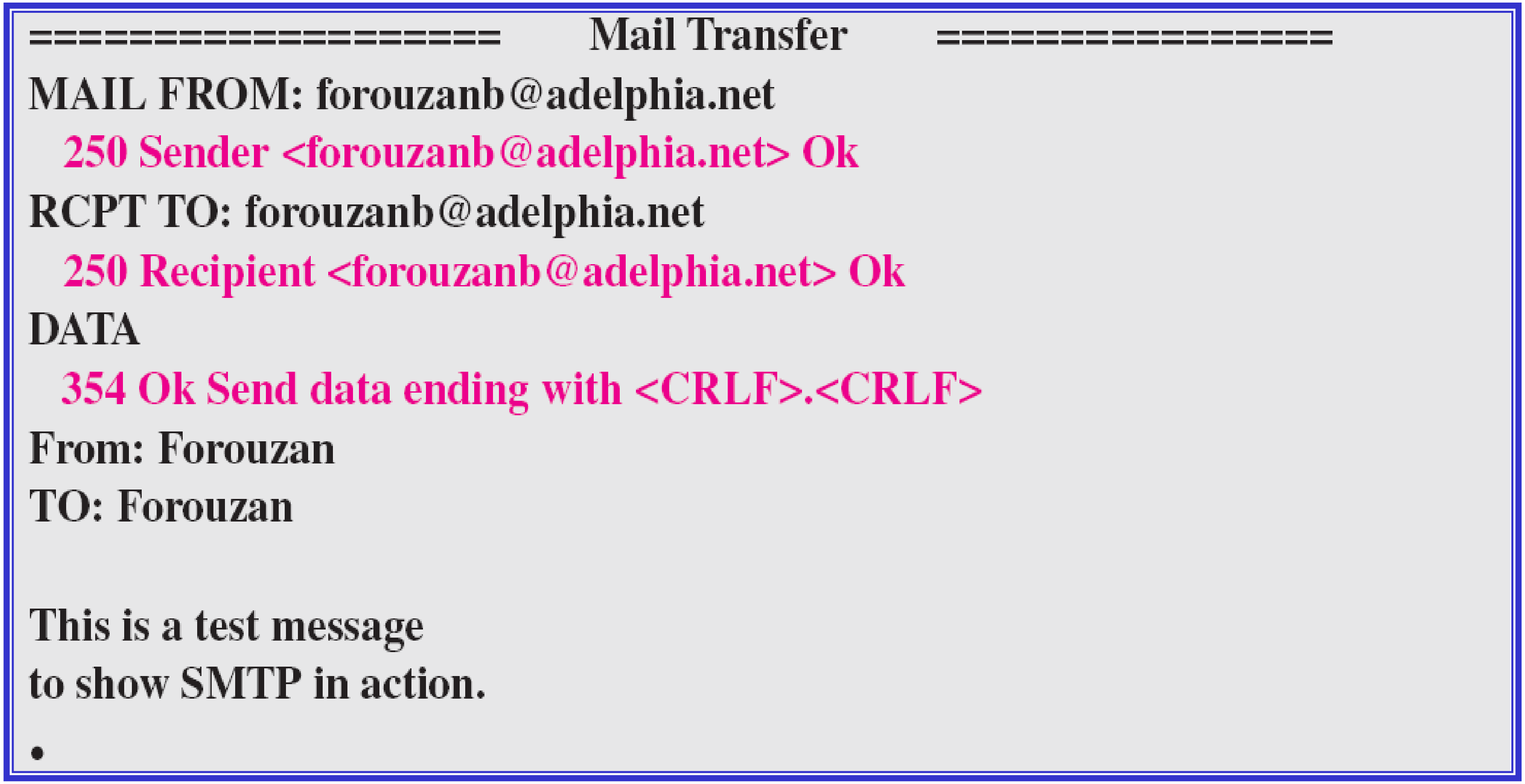
Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted: insufficient storage

Table *Responses* *(continued)*



Code	*Description*
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

Example (continued)



```
=====================      Mail Transfer      ===================
MAIL FROM: forouzanb@adelphia.net
   250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
   250 Recipient <forouzanb@adelphia.net> Ok
DATA
   354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
```

POP

- Very Simple protocol
- Typical usage : emails retrieved, stored, and read offline
- Does not maintain state across sessions
- Uses TCP and operates on port 110.
- Three phases : Authorization, transaction and Update

IMAP

- Provides more complete and complex remote mail box access.
- E-mails organized in multiple folders (mailboxes) on server.
- IMAP keeps user state across sessions.
- IMAP clients stay connected , results in faster response time.
- Permits multiple email clients to simultaneously connect to the mailbox.
- Permits partial fetch.
- Keep track of message status.

Figure *POP3 and IMAP4*

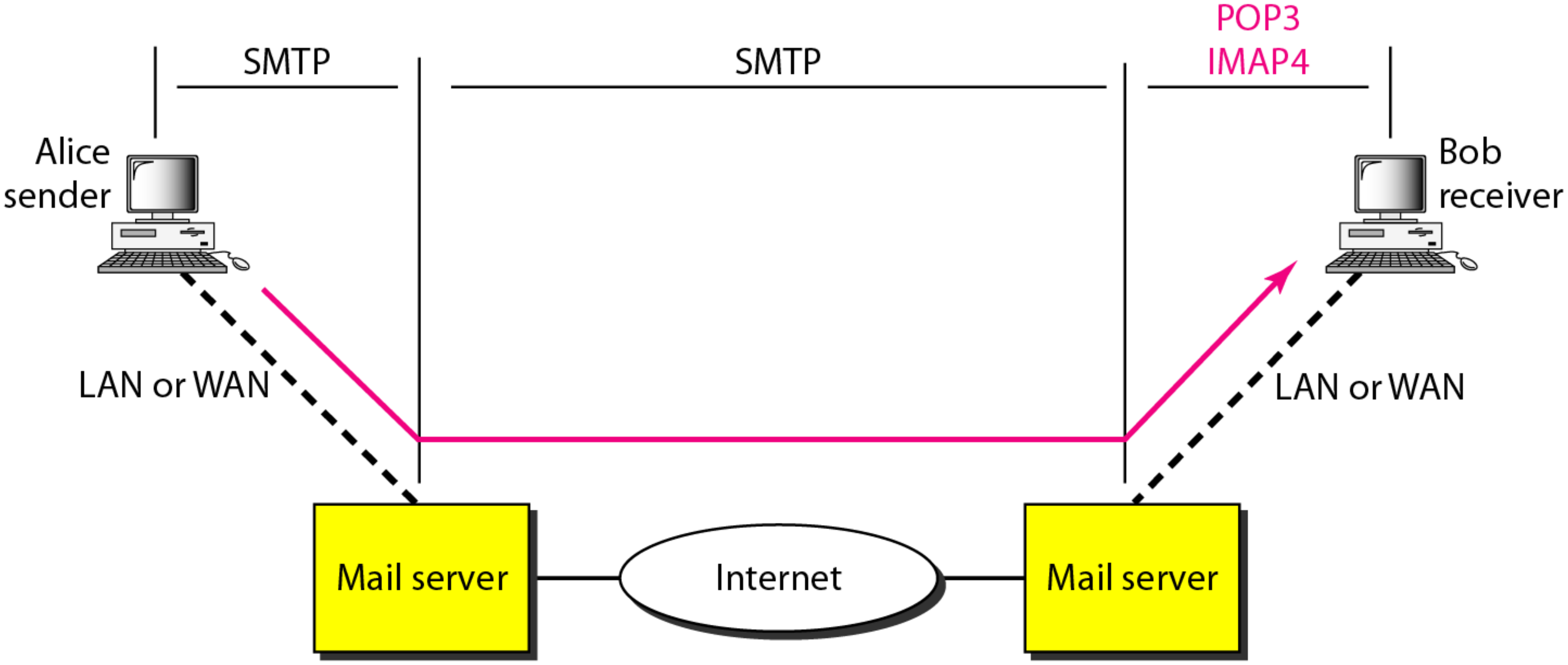
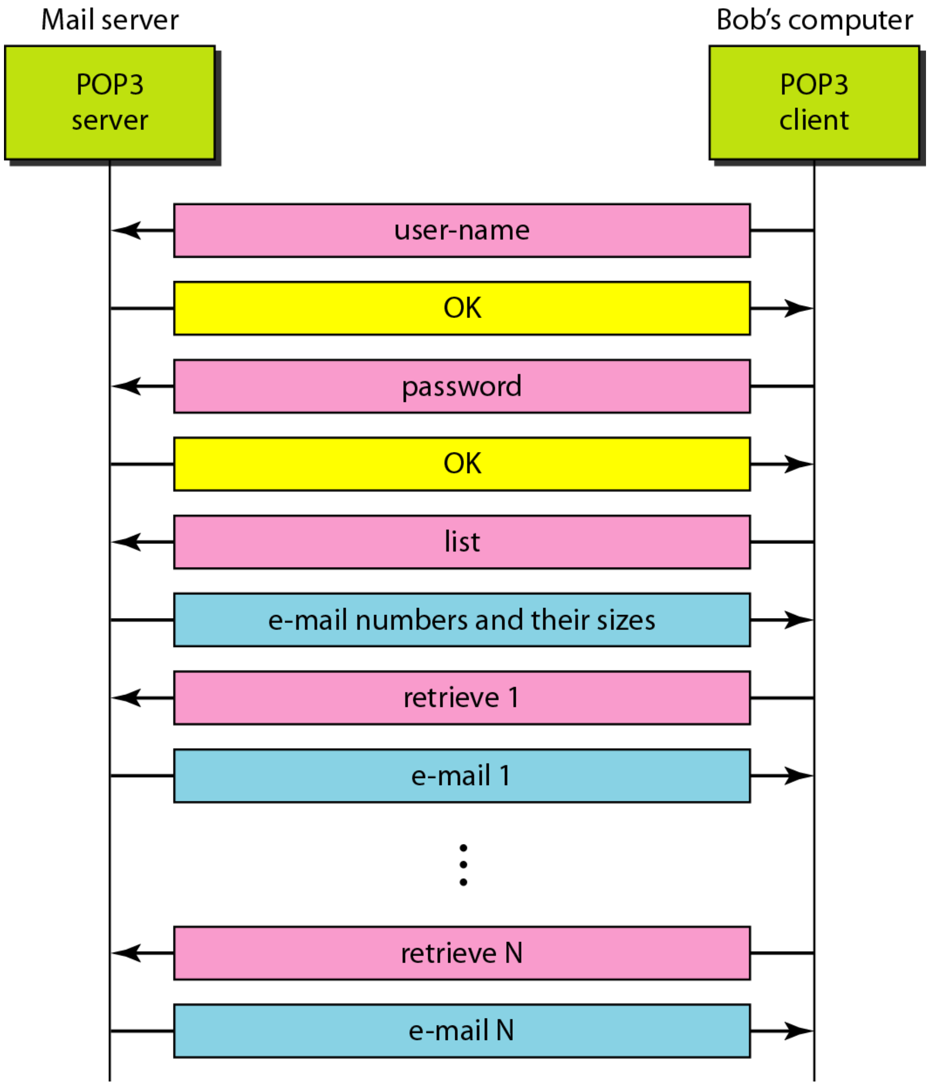


Figure *The exchange of commands and responses in POP3*



FILE TRANSFER

Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer.

Topics discussed in this section:

File Transfer Protocol (FTP)

FTP uses the services of TCP. It needs two TCP connections.

The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

FTP Protocol

- It helps to transfer files from one host to another.
- Based on client server architecture.
- Command line GUI based web browsers.
- Separate control and data channel.

  - Control channel used for authorization, browsing directory listing, kept open during a session.

  - data channel supports file transfer, closed after each transfer.
- Through a session, FTP protocol maintains state.

Figure *FTP*

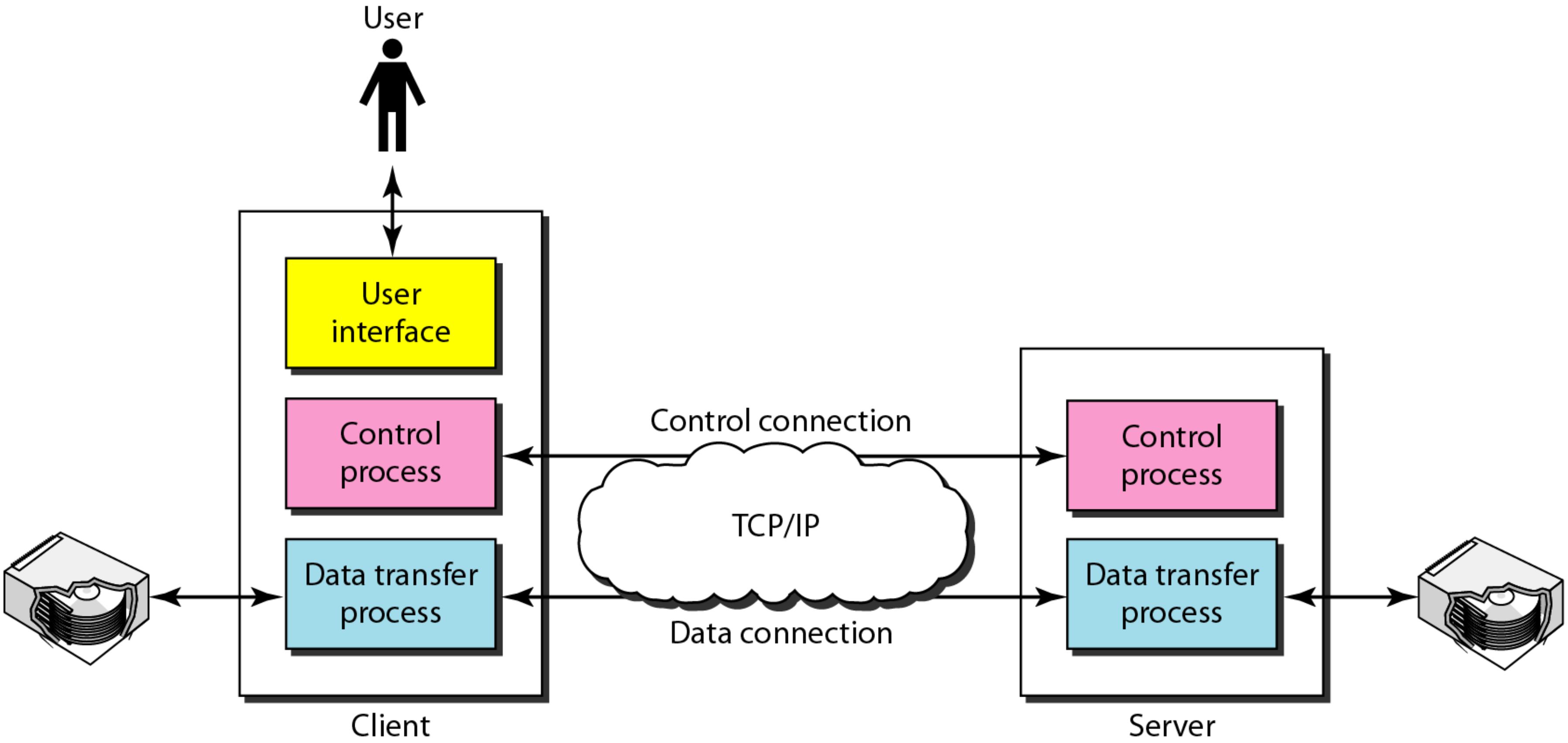

User
User interface
Control process
Data transfer process
Control connection
TCP/IP
Data connection
Control process
Data transfer process
Client
Server

Figure *Using the control connection*

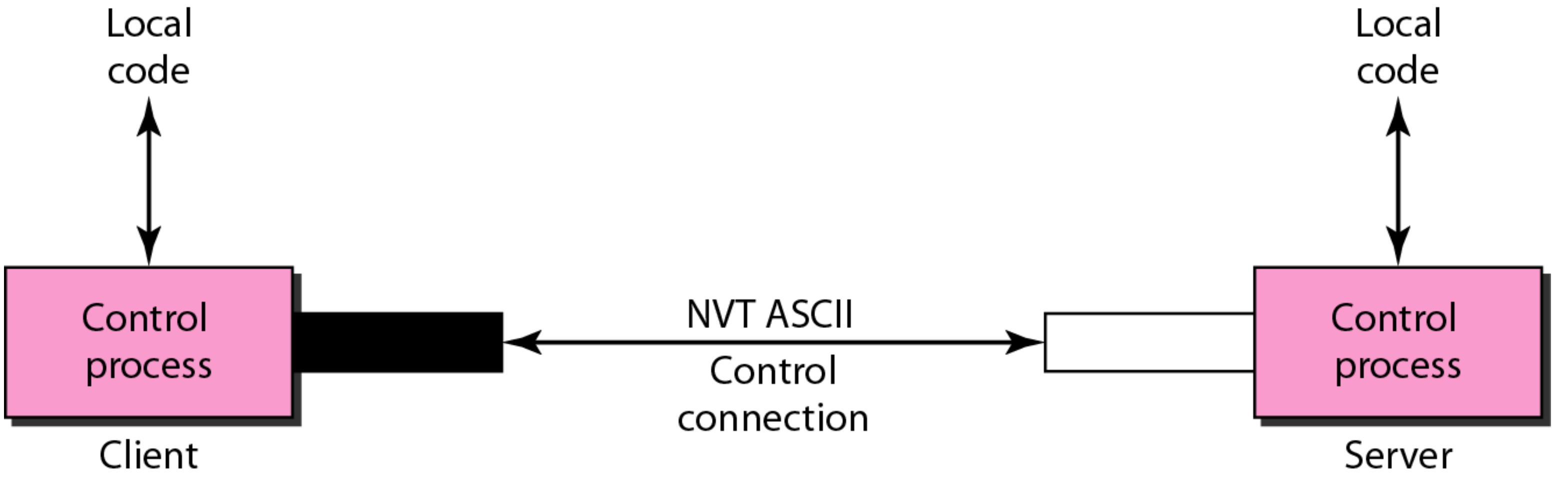
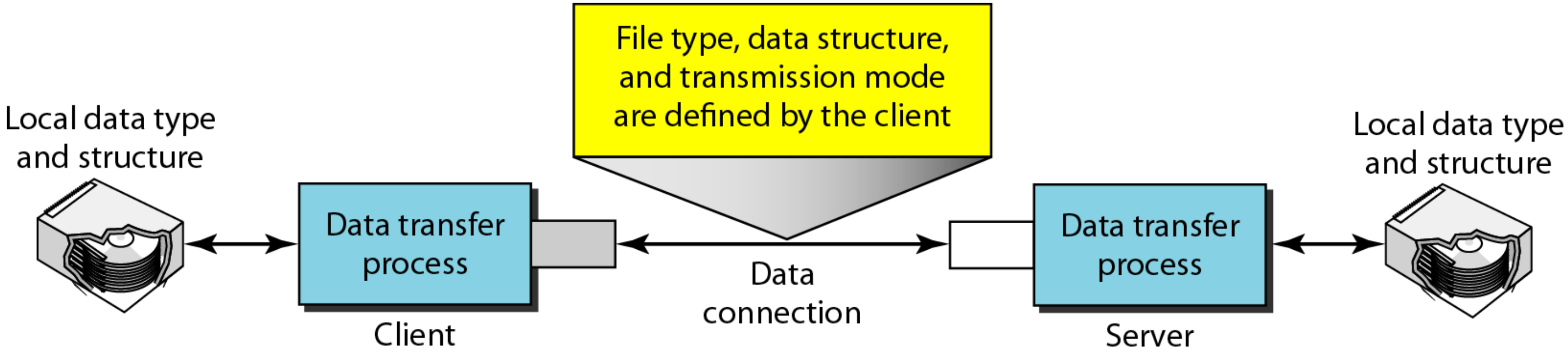


Figure *Using the data connection*



Example

The following shows an actual FTP session for retrieving a list of items in a directory. The colored lines show the responses from the server control connection; the black lines show the commands sent by the client. The lines in white with a black background show data transfer.

1. ***After the control connection is created, the FTP server sends the 220 response.***
2. ***The client sends its name.***
3. ***The server responds with 331.***

Example (continued)

4. The client sends the password (not shown).

5. The server responds with 230 (user log-in is OK).

6. The client sends the list command (ls reports) to find the list of files on the directory named report.

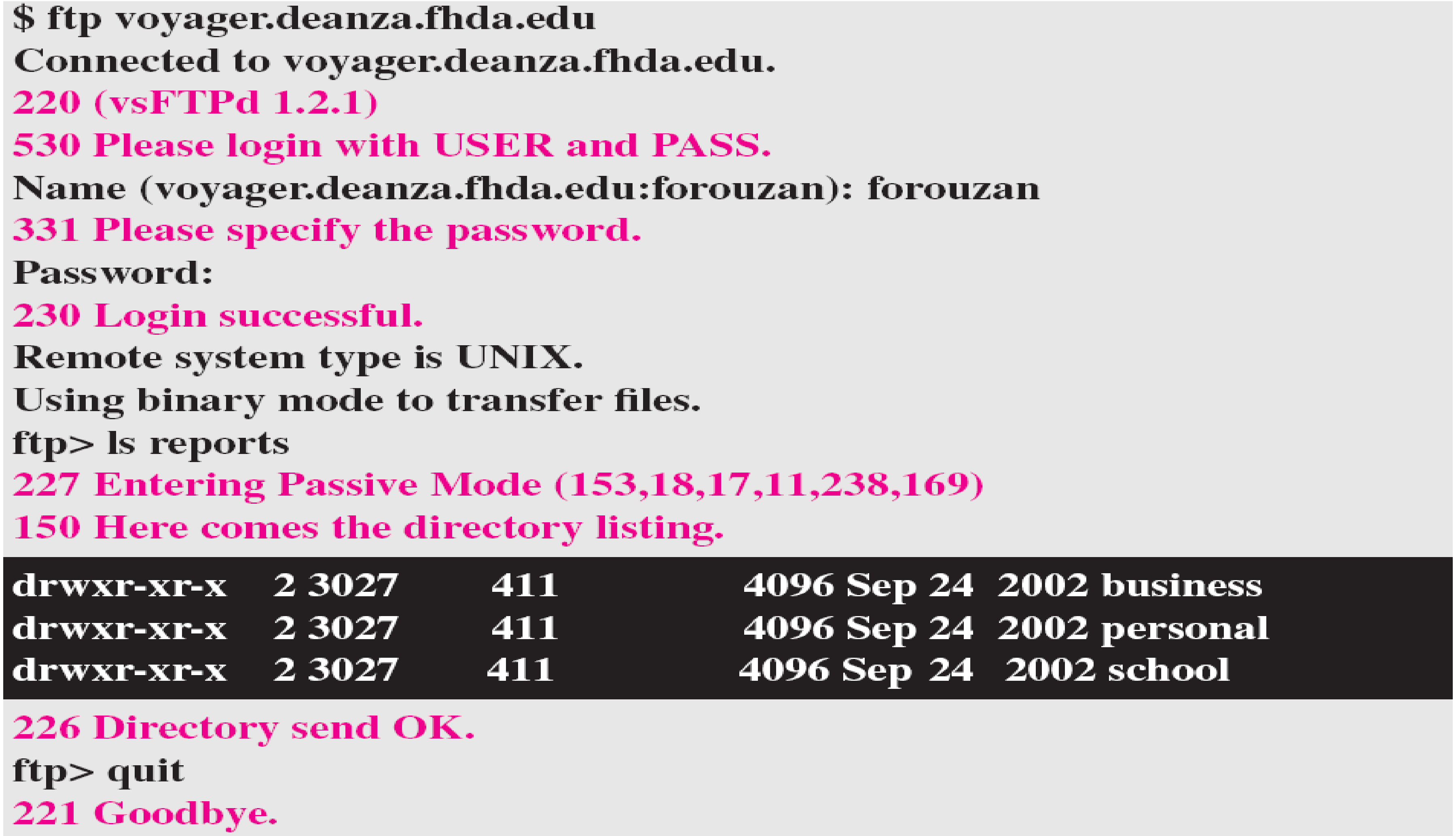
7. Now the server responds with 150 and opens the data connection.

8. The server then sends the list of the files or directories on the data connection.

9. The client sends a QUIT command.

10. The server responds with 221.

Example (continued)



```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x    2 3027      411             4096 Sep 24  2002 business
drwxr-xr-x    2 3027      411             4096 Sep 24  2002 personal
drwxr-xr-x    2 3027      411             4096 Sep 24  2002 school
226 Directory send OK.
ftp> quit
221 Goodbye.
```