COMPUTER NETWORK (BCA301)

DEPARTMENT OF COMPUTER SCIENCE

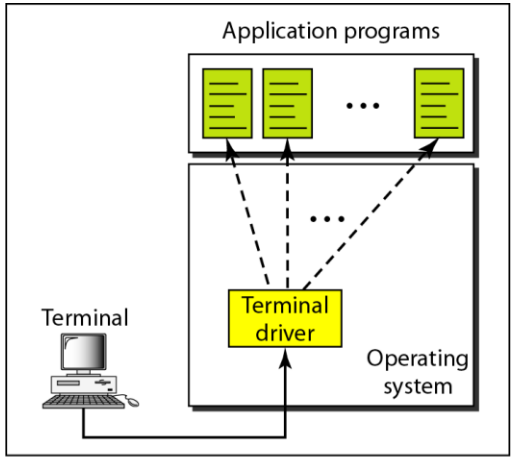
PROGRAMME: BCA

CENTRAL UNIVERSITY OF ORISSA
KORAPUT

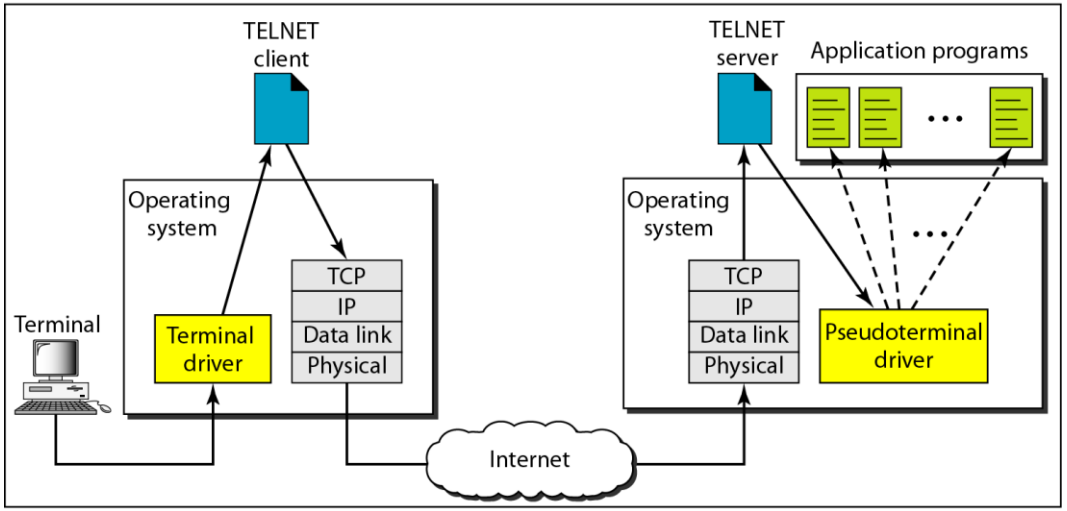
TELNET

- TELNET is an abbreviation for TErminaL NETwork.
- It is a general-purpose client/server application program.
- It is the standard TCP/IP protocol for virtual terminal service as proposed by the ISO.
- It enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

Local and remote log-in



a. Local log-in



b. Remote log-in

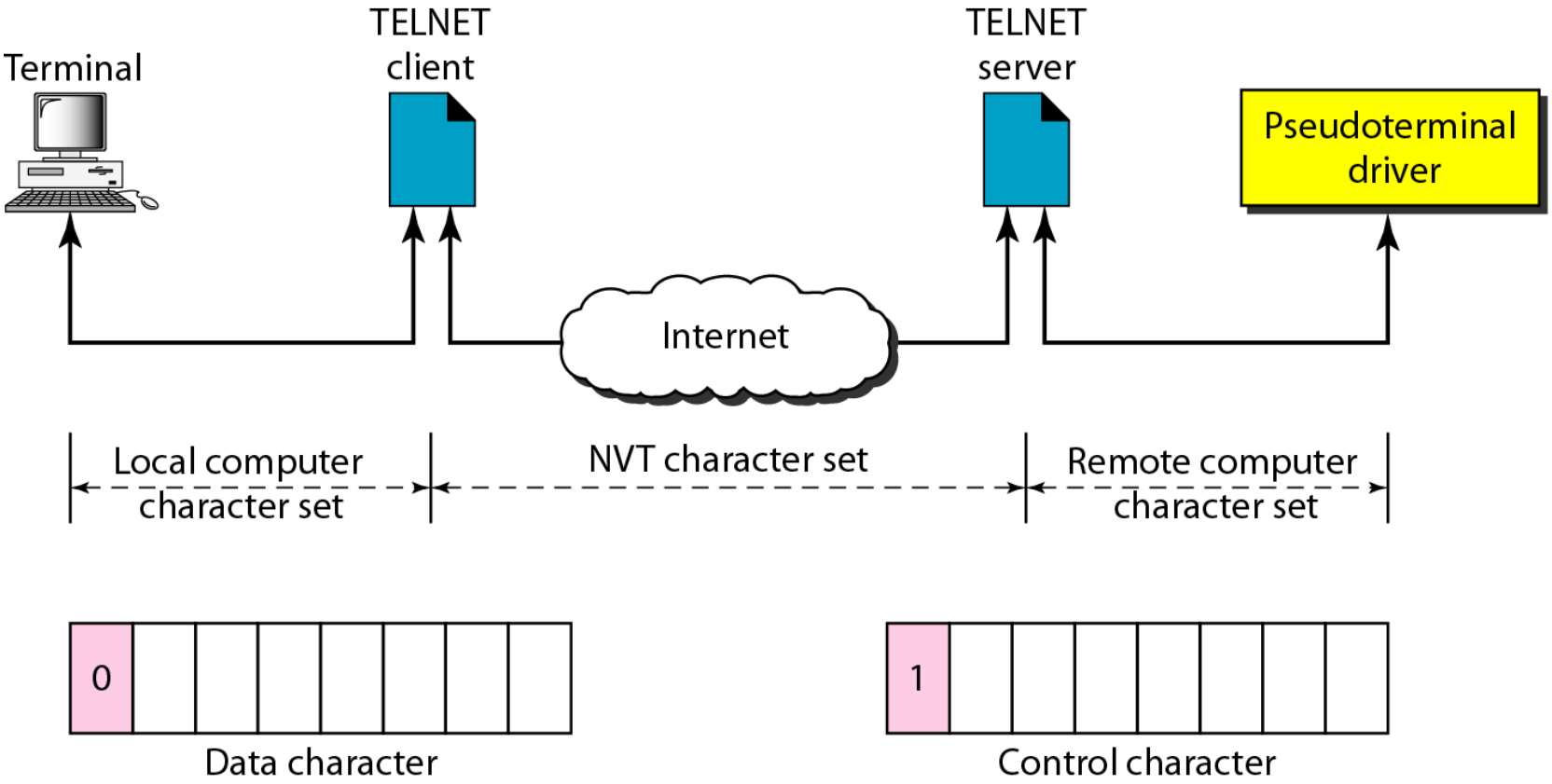
- When a user wants to access an application program or utility located on a remote machine, she performs remote log-in. Here the TELNET client and server programs come into use.
- The user sends the keystrokes to the terminal driver, where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called network virtual terminal (NVT) characters and delivers them to the local TCP/IP protocol stack.

- The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine.
- Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer.
- However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server:
- It is designed to receive characters from a terminal driver. The solution is to add a piece of software called a pseudoterminal driver which pretends that the characters are coming from a terminal.
- The operating system then passes the characters to the appropriate application program.

Network Virtual Terminal(NVT)

- If we want to access any remote computer in the world, we must first know what type of computer we will be connected to, and we must also install the specific terminal emulator used by that computer.
- TELNET solves this problem by defining a universal interface called the network virtual terminal (NVT) character set.
- Via this interface, the client TELNET translates characters(data or commands) that come from the local terminal into NVT form and delivers them to the network.
- The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.

Concept of NVT



NVT Character Set

- NVT uses two sets of characters, one for data and the other for control. Both are 8-bit bytes.
- For data, NVT is an 8-bit character set in which the 7 lowest-order bits are the same as ASCII and the highest-order bit is 0.
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest-order bit is set to 1.

Embedding

- TELNET uses only one TCP connection. The server uses the well-known port 23, and the client uses an ephemeral port. The same connection is used for sending both data and control characters.
- TELNET accomplishes this by embedding the control characters in the data stream.
- However, to distinguish data from control characters, each sequence of control characters is preceded by a special control character called interpret as control (IAC).

An example of embedding

Typed at the remote terminal

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

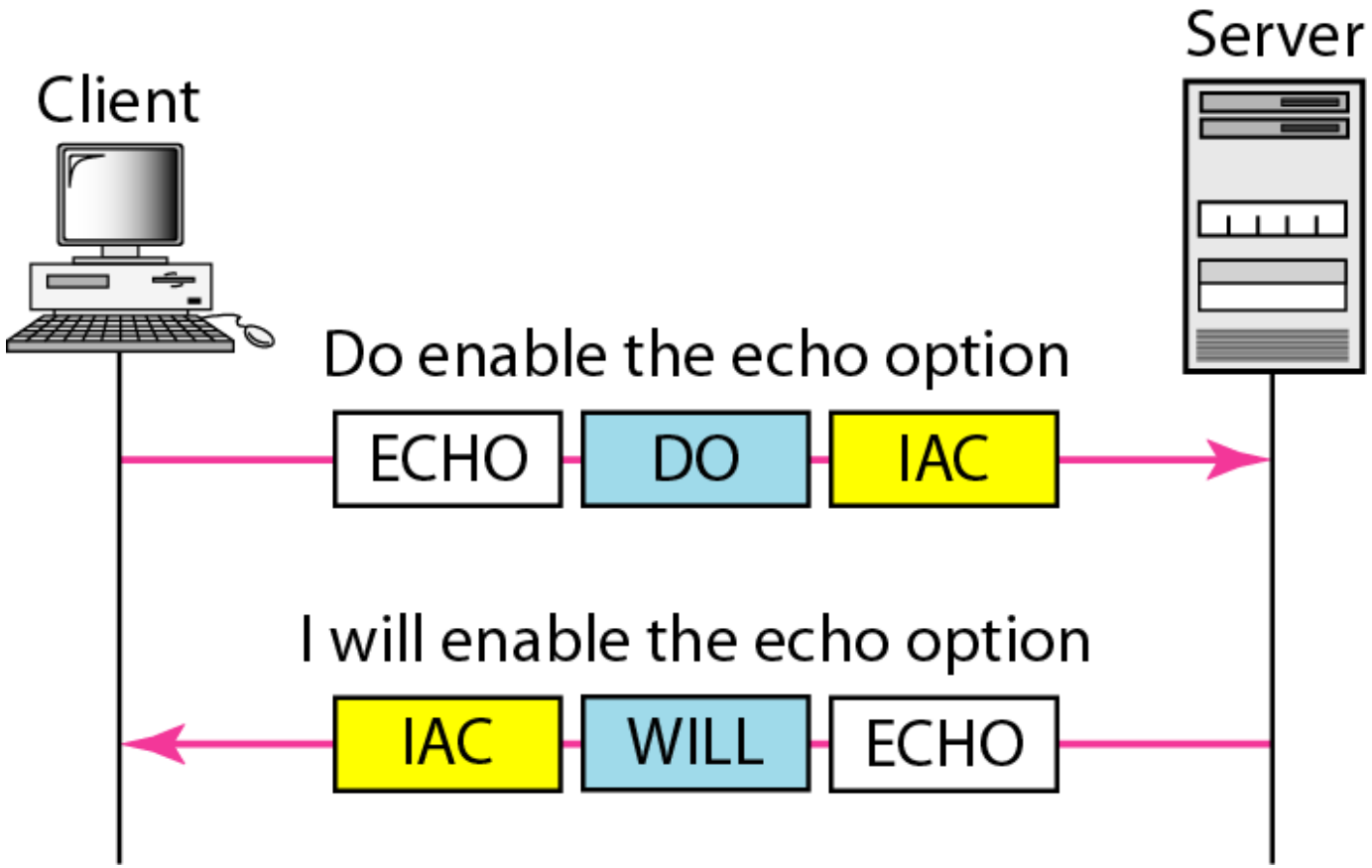
Options

Code	*Option*	*Meaning*
0	Binary	Interpret as 8-bit binary transmission.
1	Echo	Echo the data received on one side to the other.
3	Suppress go ahead	Suppress go-ahead signals after data.
5	Status	Request the status of TELNET.
6	Timing mark	Define the timing marks.
24	Terminal type	Set the terminal type.
32	Terminal speed	Set the terminal speed.
34	Line mode	Change to line mode.

NVT character set for option negotiation

Character	*Decimal*	*Binary*	*Meaning*
WILL	251	11111011	1. Offering to enable 2. Accepting a request to enable
WONT	252	11111100	1. Rejecting a request to enable 2. Offering to disable 3. Accepting a request to disable
DO	253	11111101	1. Approving an offer to enable 2. Requesting to enable
DONT	254	11111110	1. Disapproving an offer to enable 2. Approving an offer to disable 3. Requesting to disable

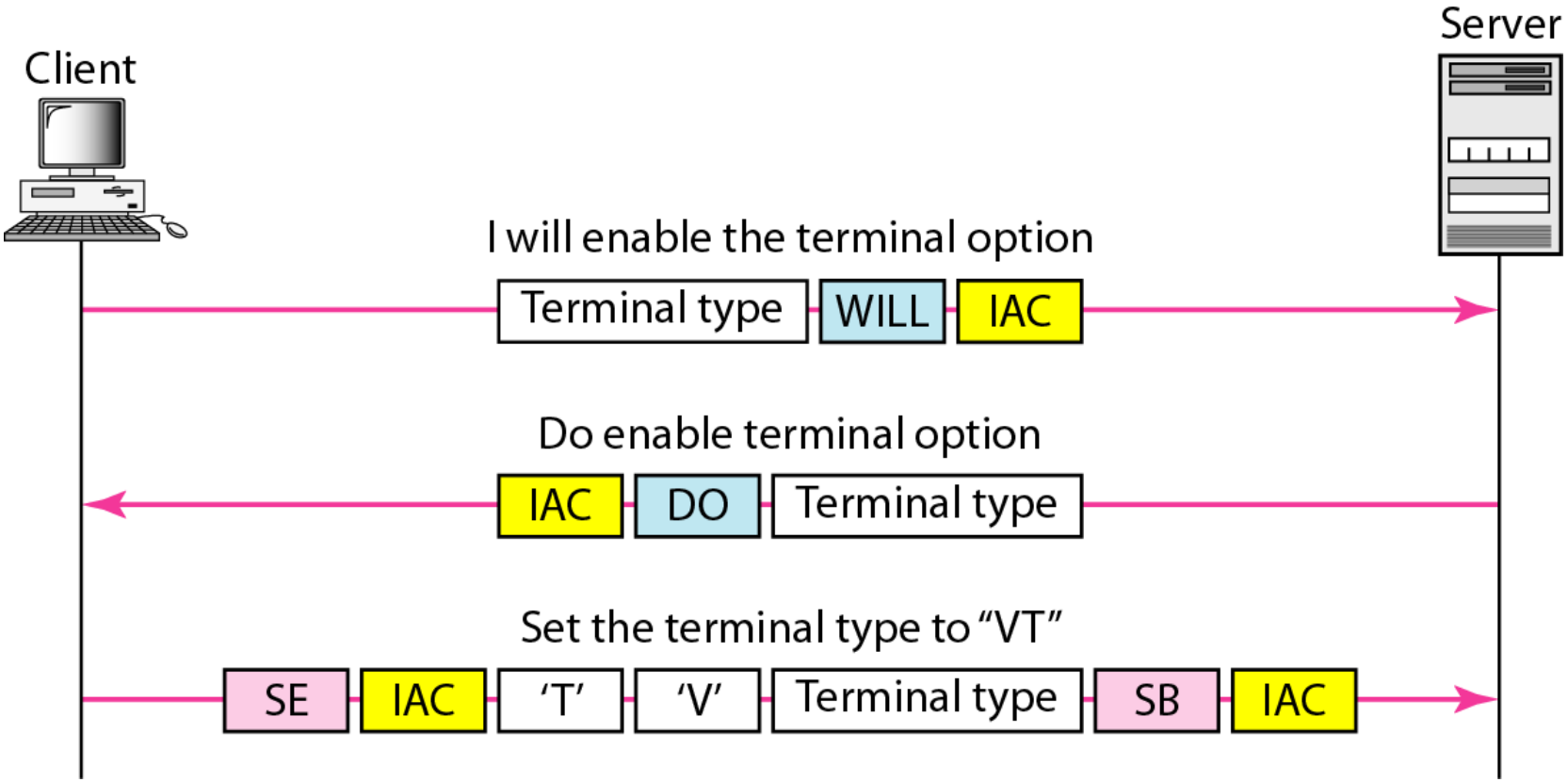
Example : Echo option



Character set for suboptions

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

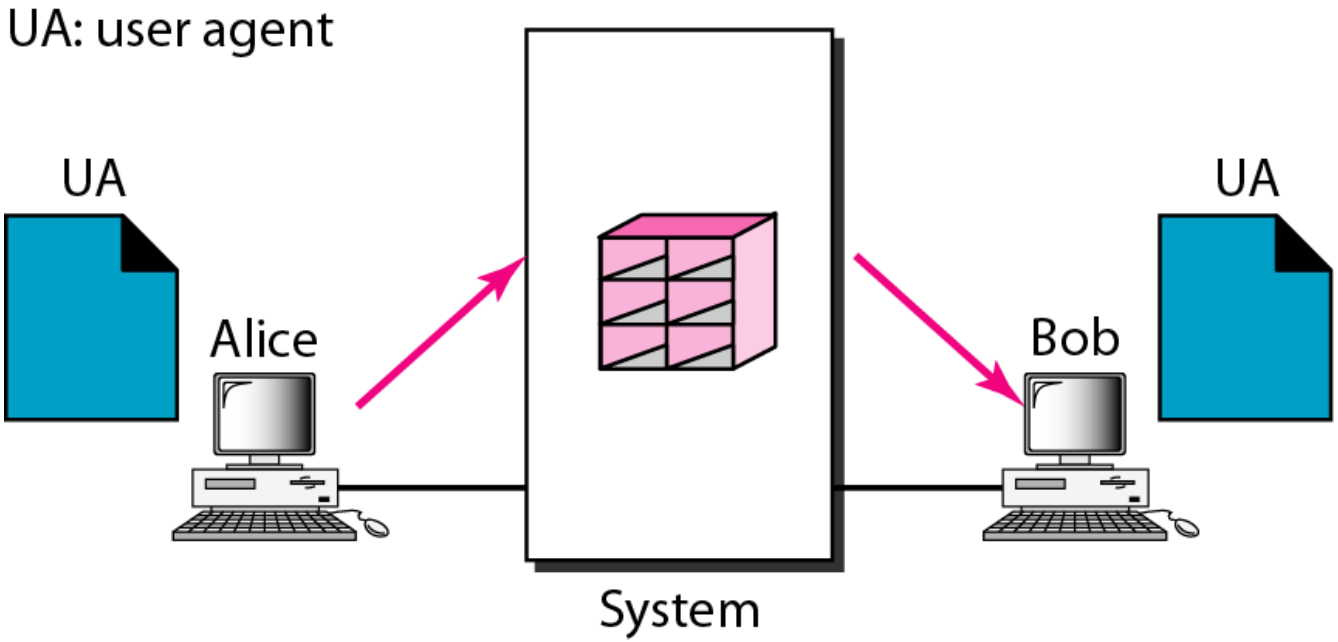
Example of suboption negotiation



Electronic Mail

- Three main components: user agent, message transfer agent, and message access agent.
-

First scenario in electronic mail

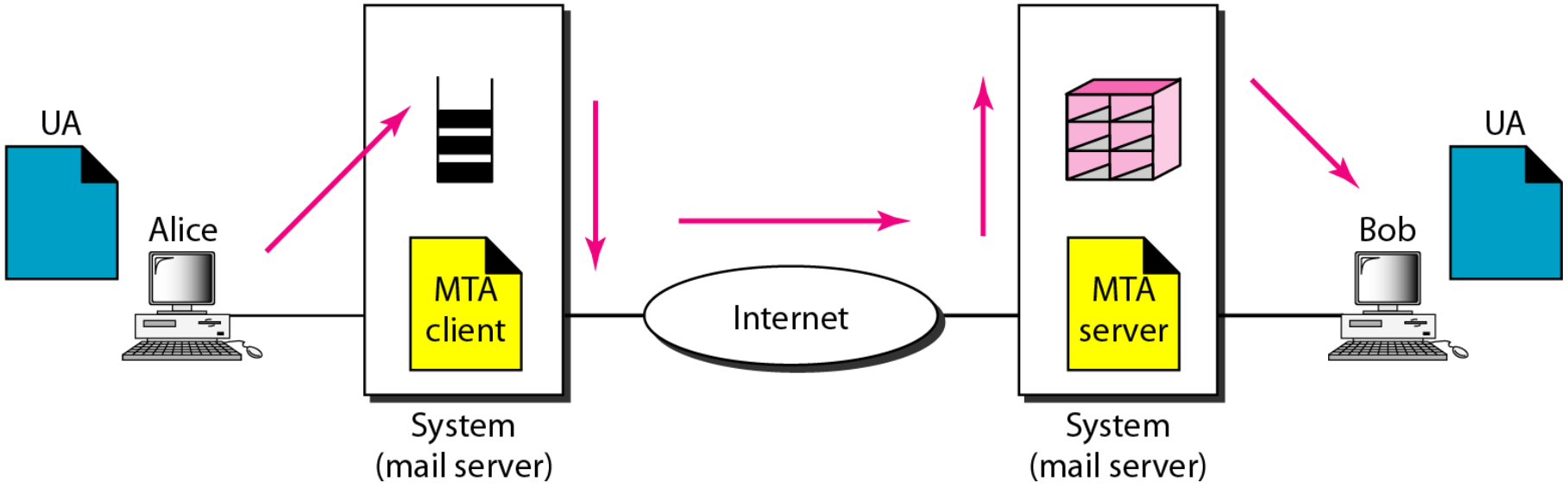


First scenario

- The sender and the receiver of the e-mail are users (or application programs) on the same system; they are directly connected to a shared system.
- The administrator has created one mailbox for each user where the received messages are stored.
- A *mailbox* is part of a local hard drive, a special file with permission restrictions.
- Only the owner of the mailbox has access to it. When Alice, a user, needs to send a message to Bob, another user, Alice runs a *user agent (VA)* program to prepare the message and store it in Bob's mailbox.
- The message has the sender and recipient mailbox addresses (names of files).
- Bob can retrieve and read the contents of his mailbox at his convenience, using a user agent.

Second scenario in electronic mail

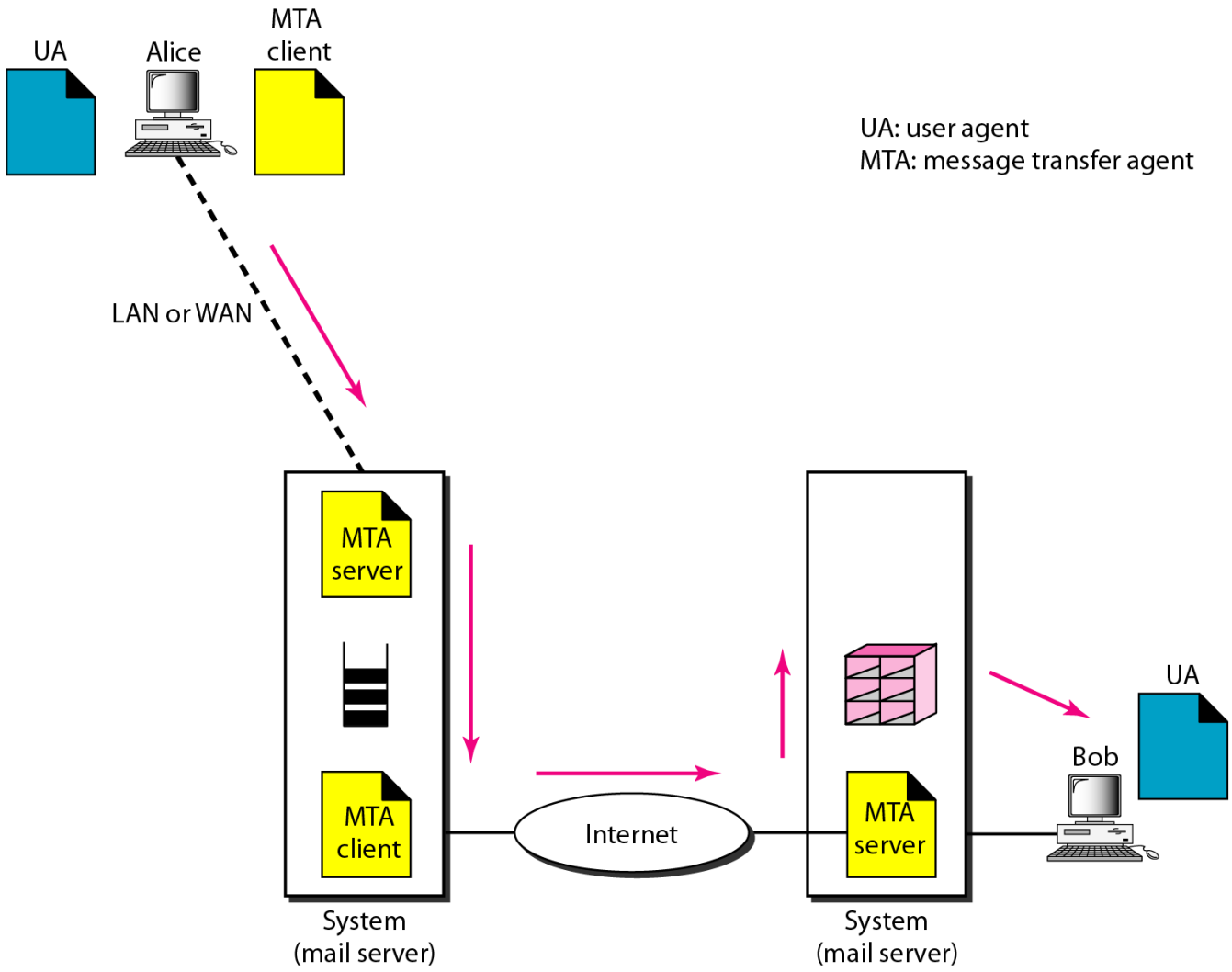
UA: user agent
MTA: message transfer agent



Second scenario

- The sender and the receiver of the e-mail are users (or application programs) on two different systems.
- The message needs to be sent over the Internet.
- Here we need user agents (VAs) and message transfer agents (MTAs).
- Alice needs to use a user agent program to send her message to the system at her own site.
- The system (sometimes called the mail server) at her site uses a queue to store messages waiting to be sent.
- Bob also needs a user agent program to retrieve messages stored in the mailbox of the system at his site.
- The server needs to run all the time because it does not know when a client will ask for a connection.
- The client, on the other hand, can be alerted by the system when there is a message in the queue to be sent.

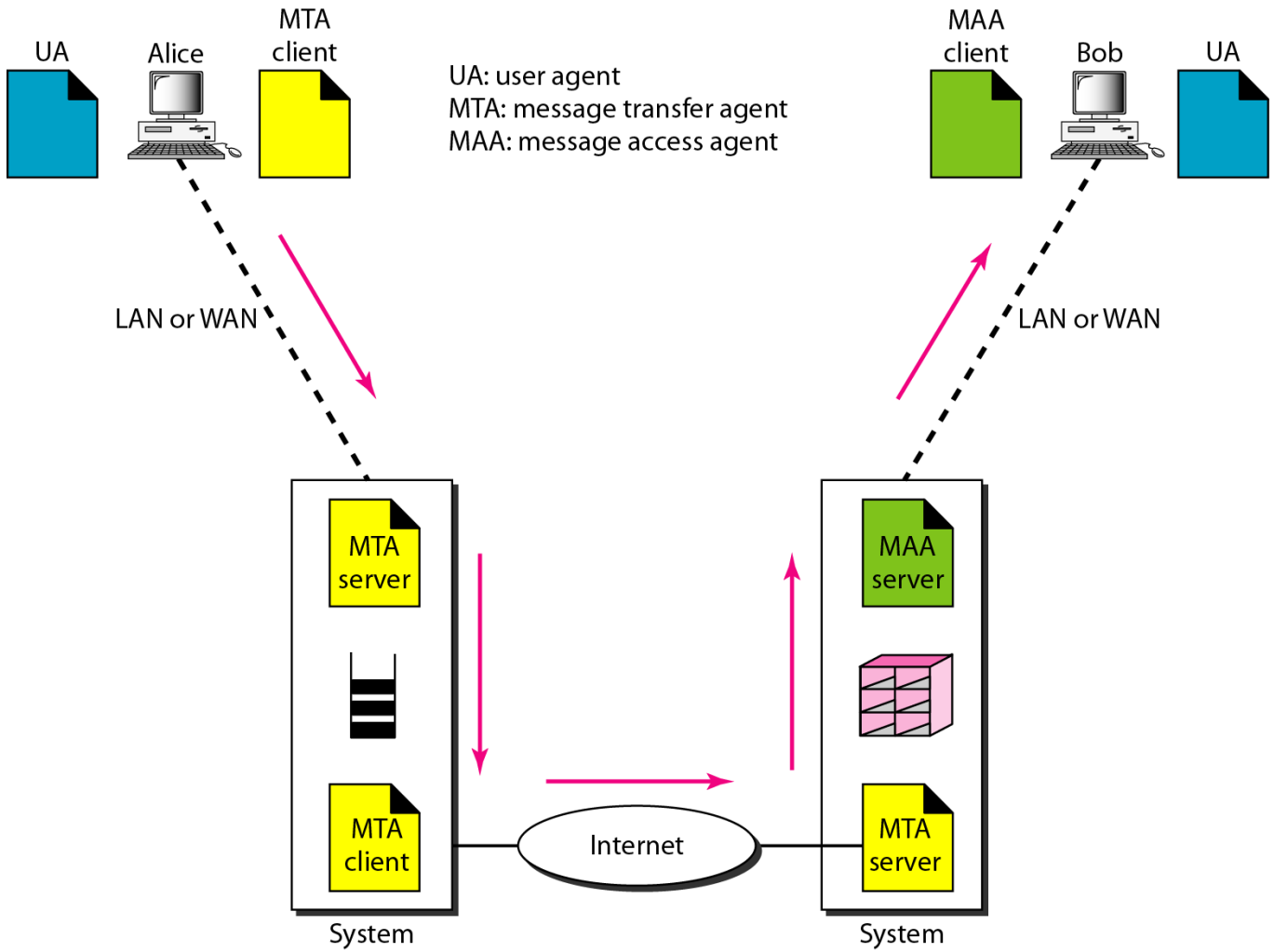
Third scenario in electronic mail



Third scenario

- Either Alice is connected to the system via a point-to-point WAN, such as a dial-up modem, a DSL, or a cable modem; or she is connected to a LAN in an organization that uses one mail server for handling e-mails-all users need to send their messages to this mail server.
- Alice still needs a user agent to prepare her message. She then needs to send the message through the LAN or WAN.
- This can be done through a pair of message transfer agents (client and server).
- Whenever Alice has a message to send, she calls the user agent which, in turn, calls the MTA client.
- The MTA client establishes a connection with the MTA server on the system, which is running all the time.
- The system at Alice's site queues all messages received.
- It then uses an MTA client to send the messages to the system at Bob's site; the system receives the message and stores it in Bob's mailbox.
- At his convenience, Bob uses his user agent to retrieve the message and reads it.

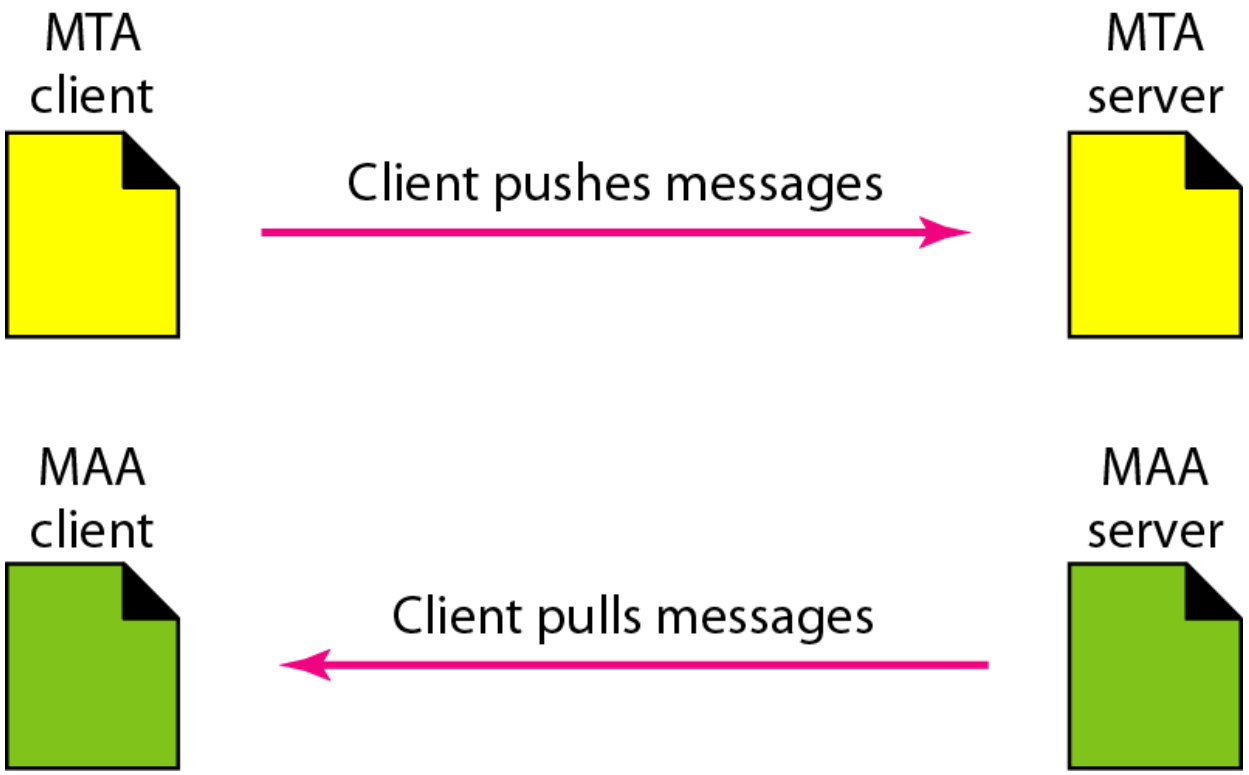
Fourth scenario in electronic mail



Fourth scenario

- Bob is also connected to his mail server by a WAN or a LAN.
- After the message has arrived at Bob's mail server, Bob needs to retrieve it.
- Here, we need another set of client/server agents, which we call message access agents (MAAs).
- Bob uses an MAA client to retrieve his messages. The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages.
- First, Bob cannot bypass the mail server and use the MTA server directly.
- To use MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive.
- Second, note that Bob needs another pair of client/server programs: message access programs. This is so 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.

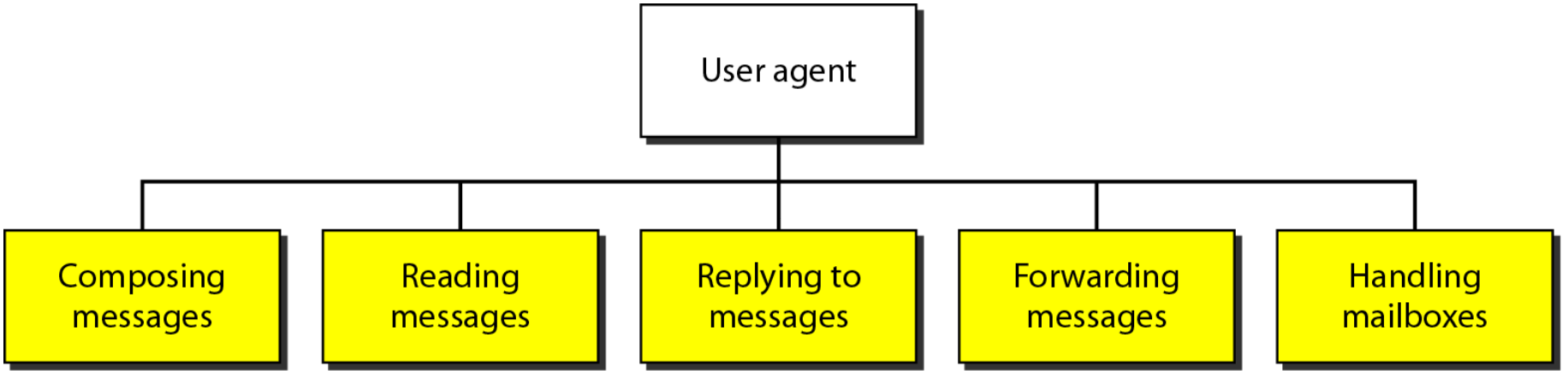
Push versus pull in electronic email



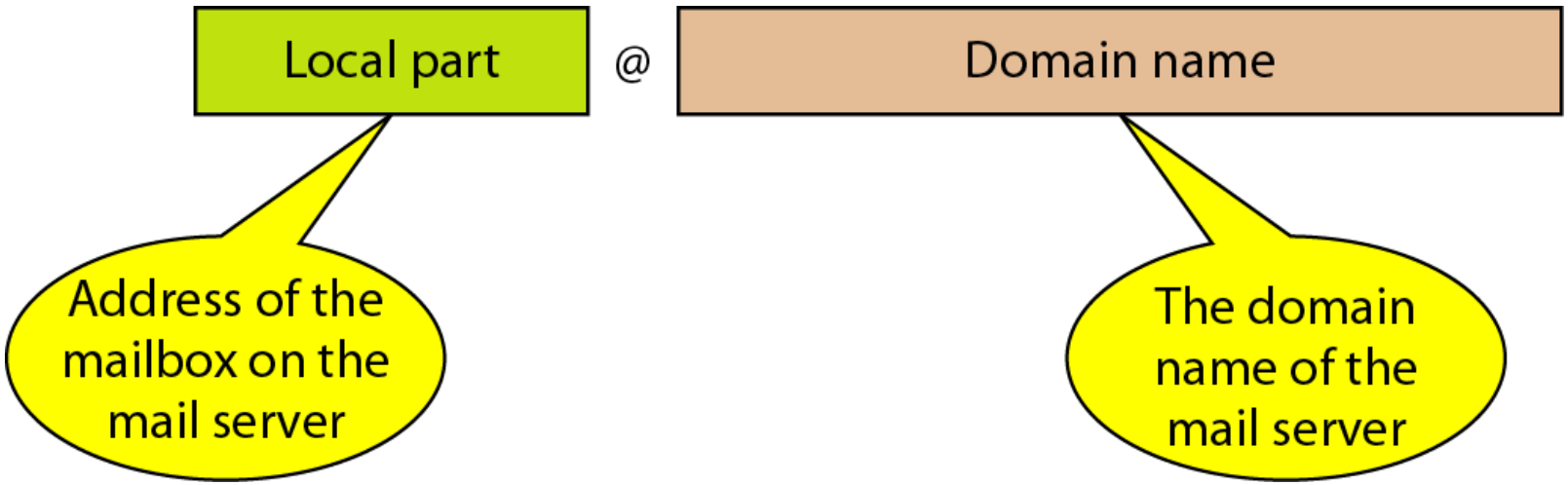
User Agent

- 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.
- **Services Provided by a User Agent:** A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles mailboxes.
- **User Agent Types :** There are two types of user agents: ***command-driven and GUI-based***.

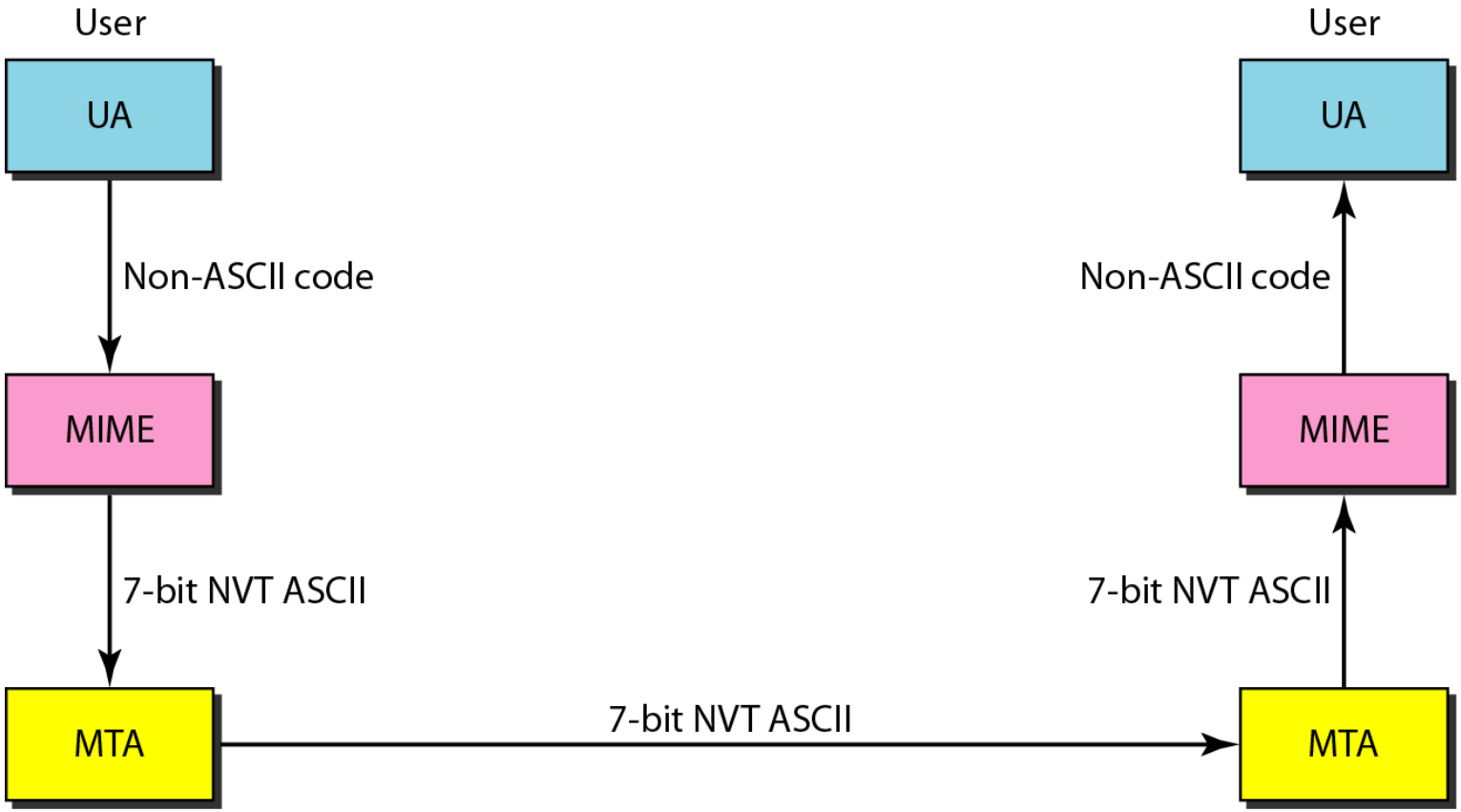
Services of User Agent



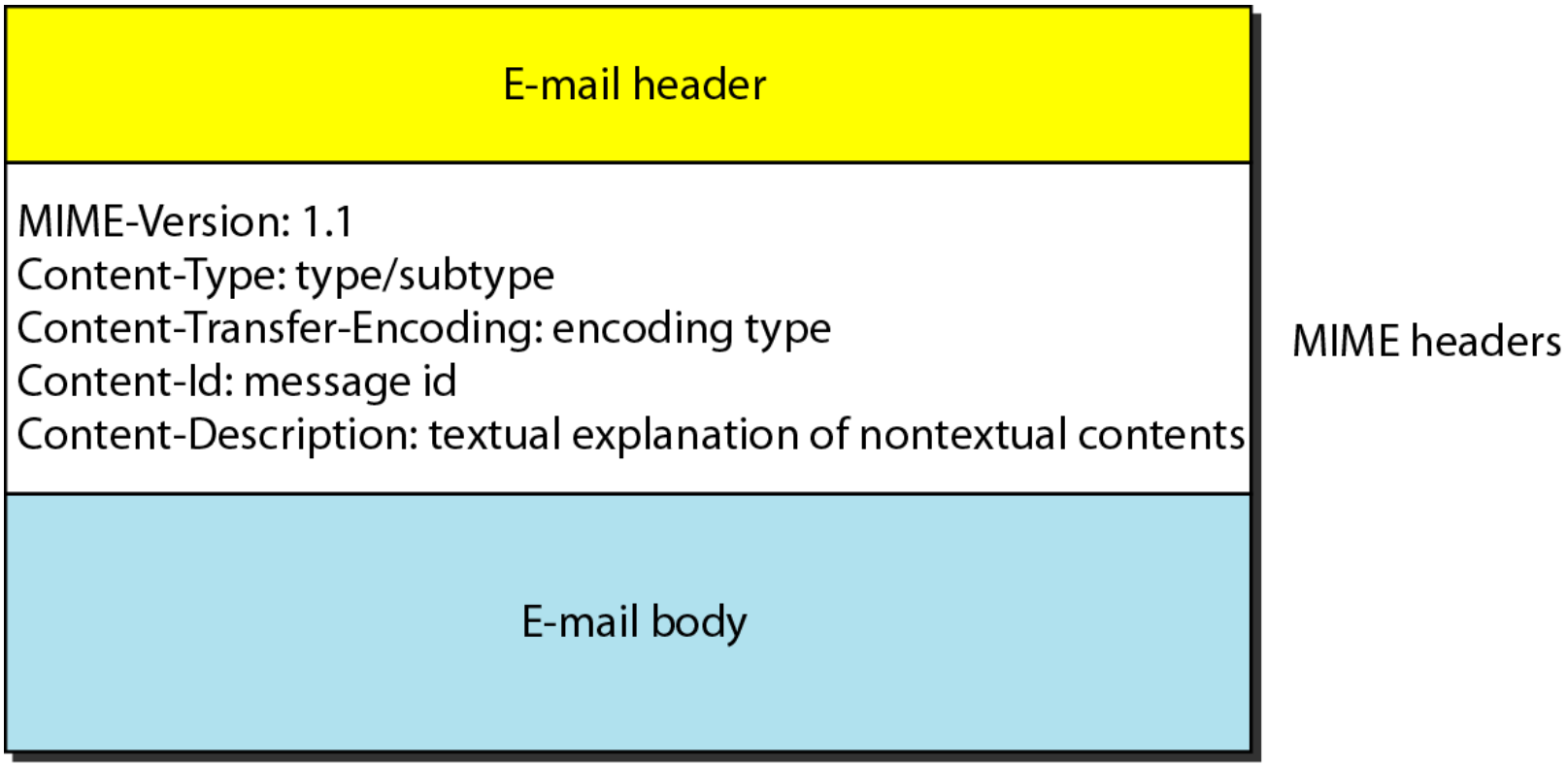
E-mail address



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

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

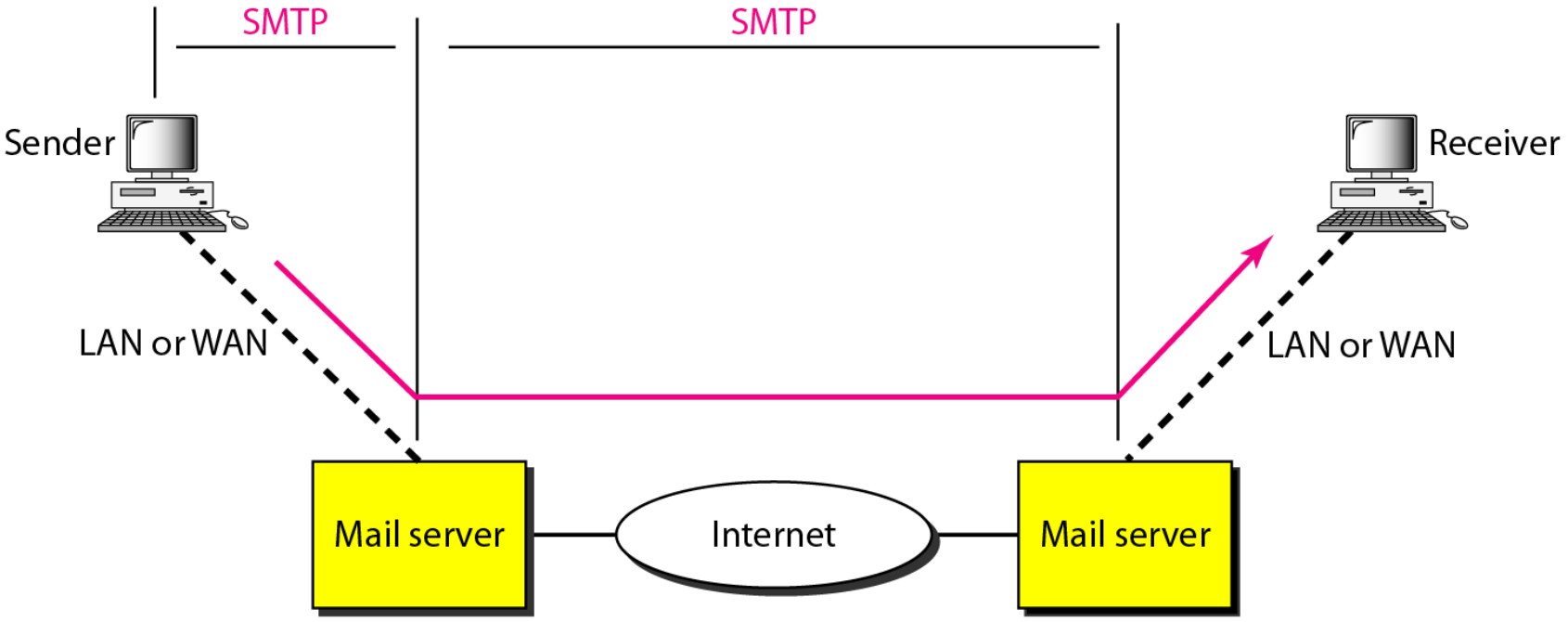
Data types and subtypes in MIME

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)

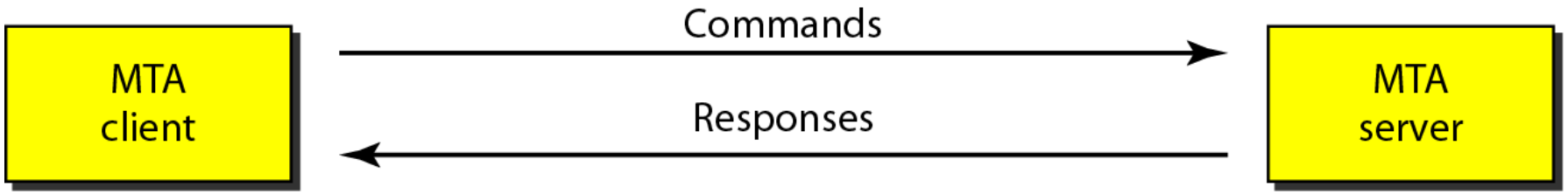
Content-transfer-encoding

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 range


SMTP
SMTP
Sender
Receiver
LAN or WAN
LAN or WAN
Mail server
Internet
Mail server

Commands and responses



Command format

Keyword: argument(s)

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

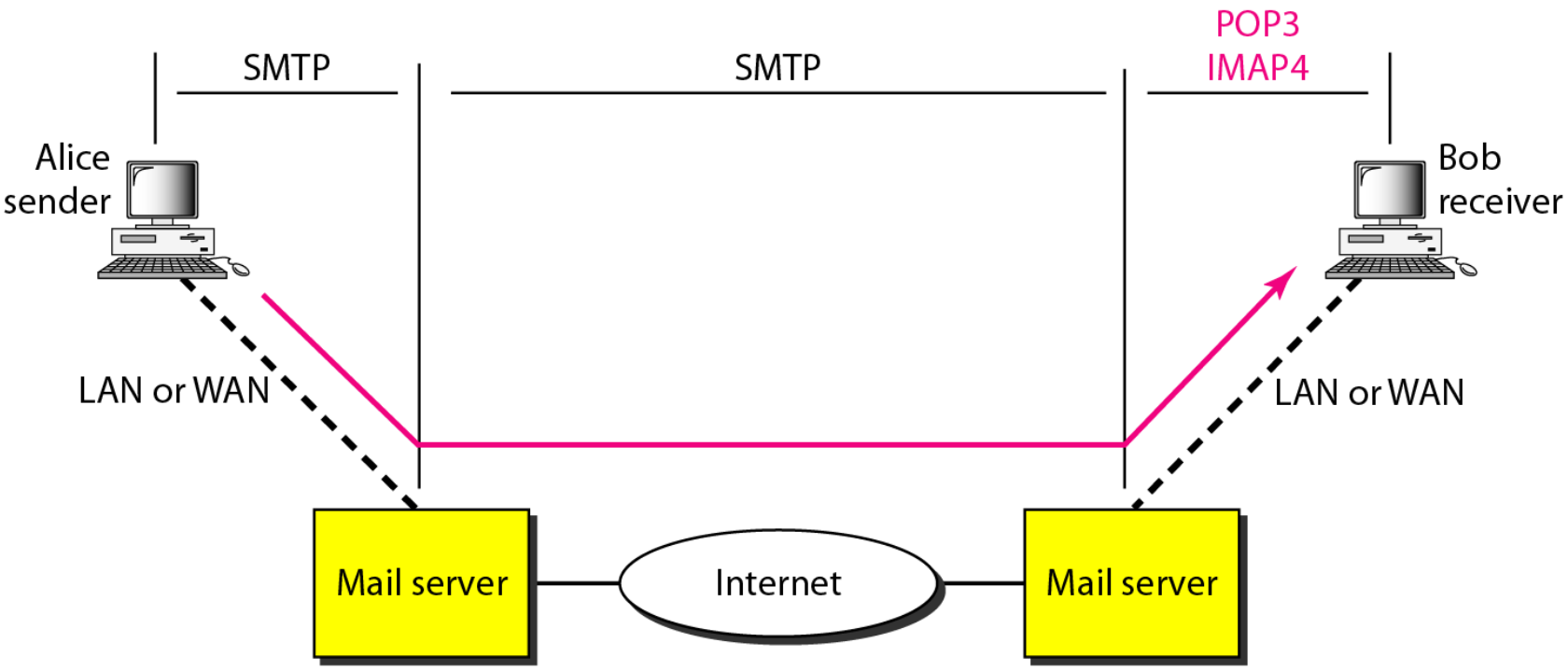
Responses

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

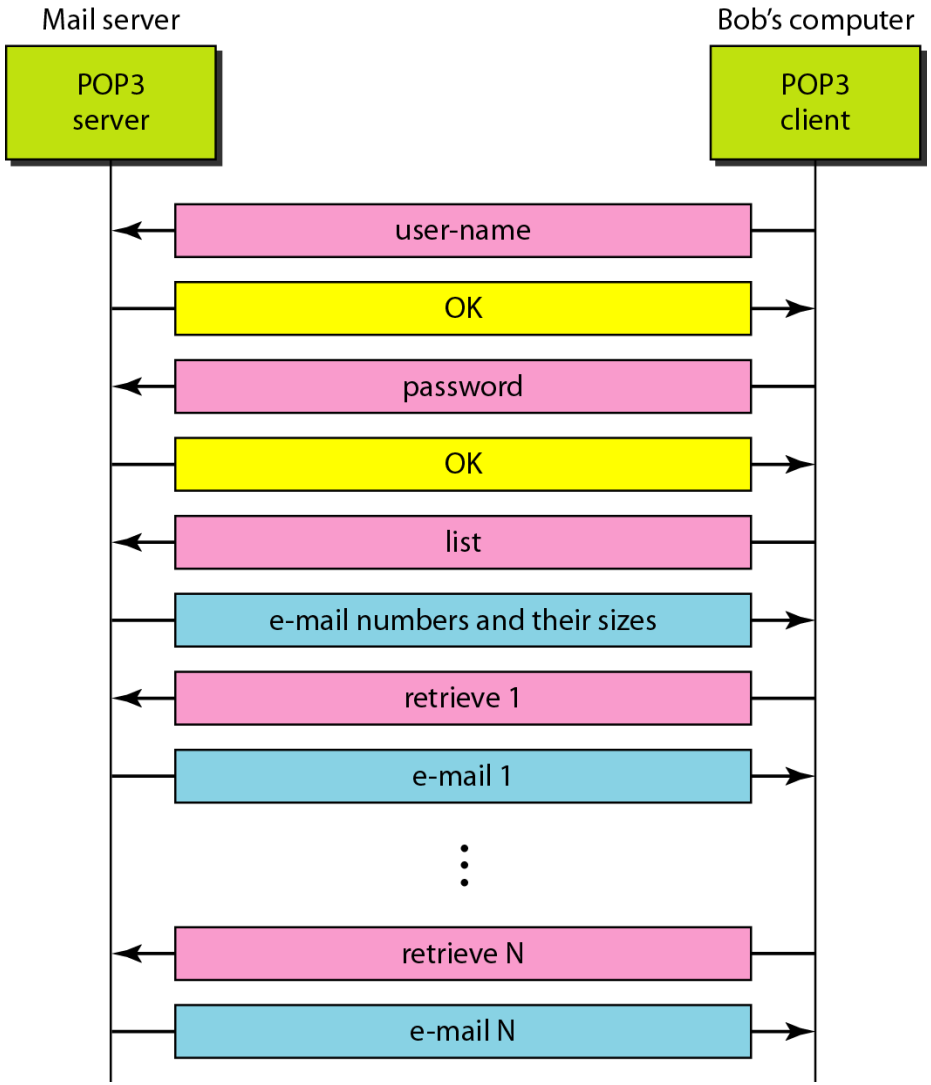
Responses (continued)

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

POP3 and IMAP4



The exchange of commands and responses in POP3



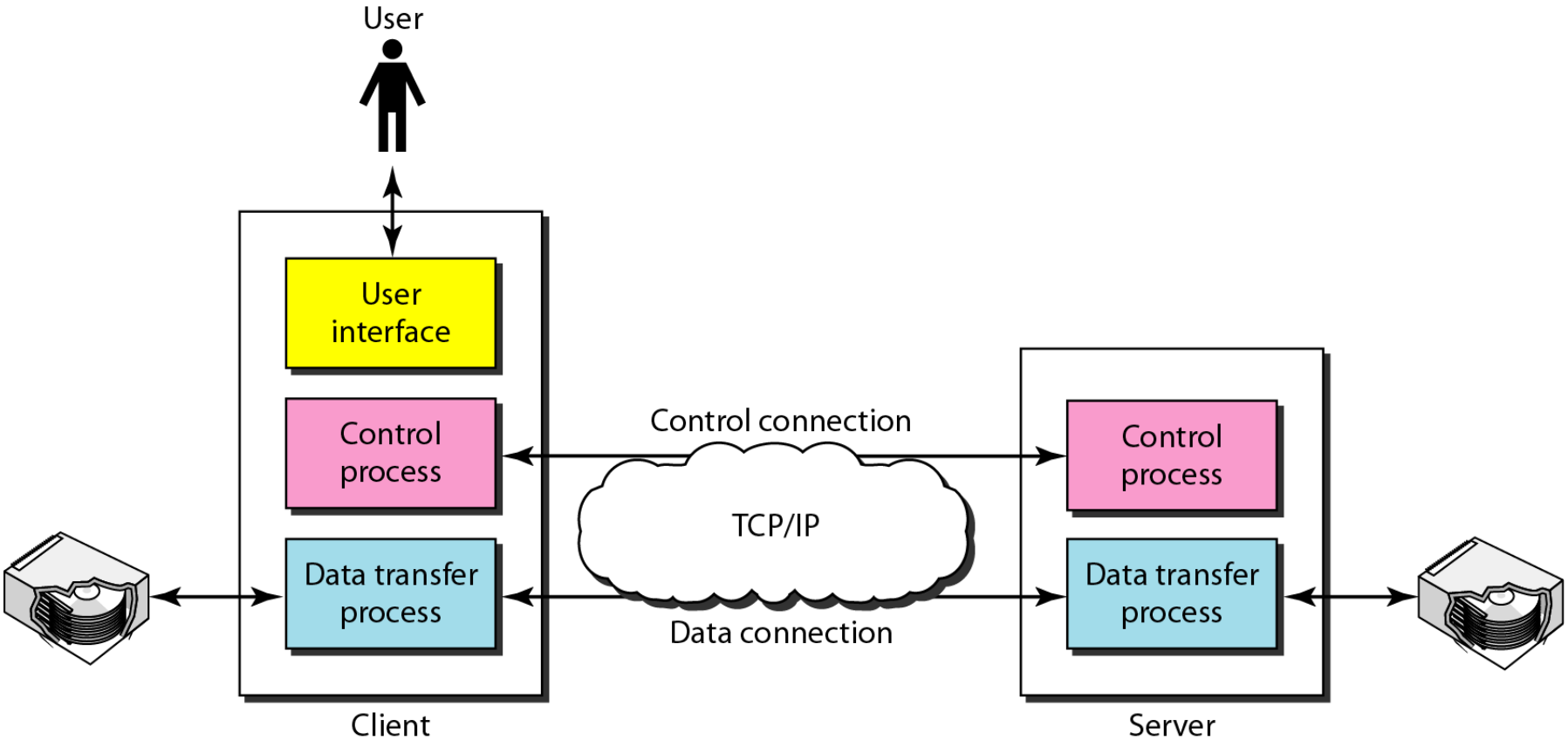
File Transfer Protocol (FTP)

- It is the standard mechanism provided by TCP/IP for copying a file from one host to another.
- It establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses).
- The control connection uses very simple rules of communication.
- The data connection, on the other hand, needs more complex rules due to the variety of data types transferred. However, the difference in complexity is at the FTP level, not TCP.
- For TCP, both connections are treated the same.
- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.

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.
- The control connection is made between the control processes. The data connection is made between the data transfer processes.
- The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file 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.

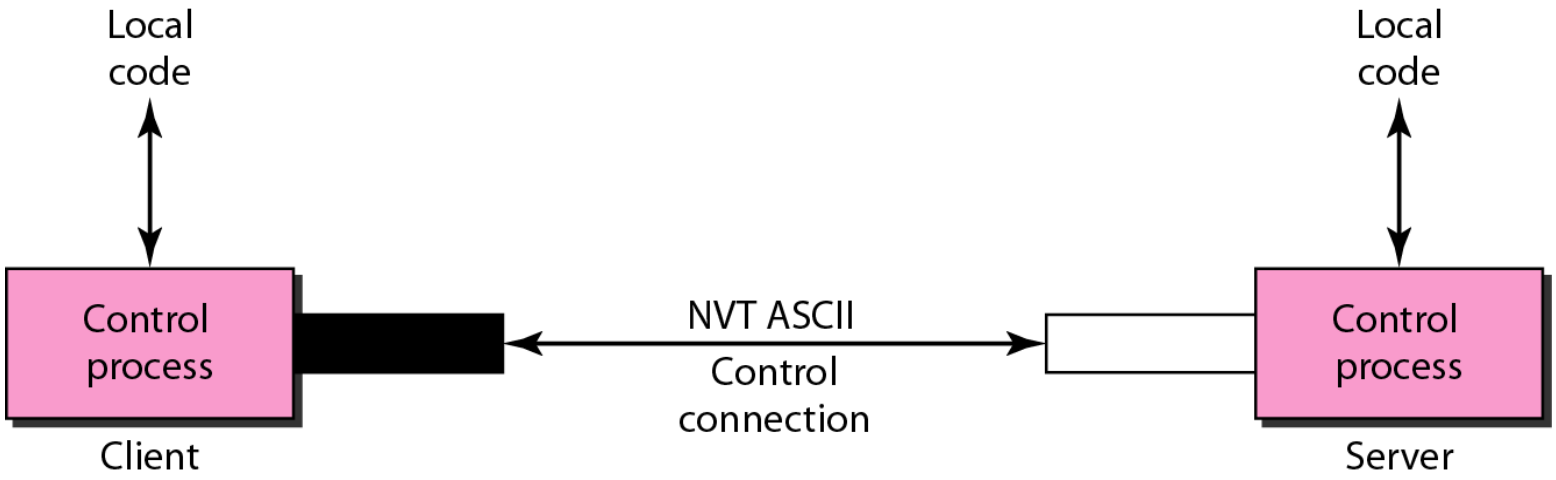
FTP


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

Communication over Control Connection

- FTP uses the same approach as SMTP to communicate across the control connection.
- It uses the 7-bit ASCII character set.
- 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.

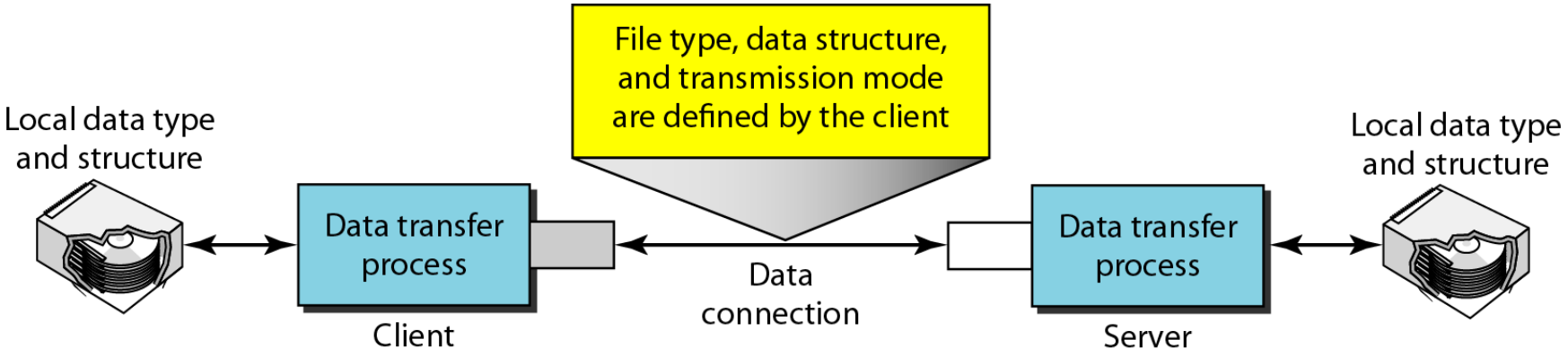
Using the control connection



Communication over Data Connection

- The purpose of the data connection is different from that of the control connection.
- 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:
- A file is to be copied from the server to the client. This is called retrieving a file. It is done under the supervision of the RETR command,
- A file is to be copied from the client to the server. This is called storing a file. It is done under the supervision of the STOR command.
- A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command.
- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.

Using the data connection



- **File Type :**FTP can transfer one of the following file types across the data connection: an ASCII file, EBCDIC file, or image file.
- **Data Structure :**FTP can transfer a file across the data connection by using one of the following interpretations about the structure of the data: file structure, record structure, and page structure.
- **Transmission Mode :**FTP can transfer a file across the data connection by using one of the following three transmission modes: stream mode, block mode, and compressed mode.

REFRENCES

- “ DATA COMMUNICATIONS AND NETWORKING ”, Behrouz A. Forouzan And Sophia Chung Fegan , Fourth Edition , McGraw-Hill