

ELECTIVE –II - TCP/IP(18MIT35E)

UNIT-IV: File Transfer Protocol (FTP): Connections – Communication Command Processing – file Transfer – User interface – Anonymous FTP. Simple Mail Transfer Protocol (SMTP): User Agent (UA) – Addresses – delayed Delivery – Aliases – Mail transfer agent (MTA) – Commands and Responses – Mail Transfer Phases – Multipurpose Internet Mail Extensions (MIME) – Mail Delivery – Mail Access Protocols.

Text Book :

1.Behrouz A. Forouzan, “TCP/IP Protocol Suite”, Tata Mcgraw-Hill Publishing Company, Second edition.

Reference Books:

1.W. Richard Stevens, “TCP/IP Illustrated: The Protocols”, Vol.1, Pearson Education.

2. Comer , ” Inter networking with TCP/IP : Principles ,protocols & Architecture”, Vol.1,fourth Edition, Pearson Education.

Prepared by

*Dr.M.Soranamageswari*₁

4.1 File Transfer Protocol (FTP)

- 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.
- **OBJECTIVES**
- *The chapter has several objectives:*
- To discuss FTP and two connections used in this protocol: control connection and data connection.
- To discuss six classes of commands sent by the client to establish communication with the server.
- To explain three types of file transfer transferred by FTP.

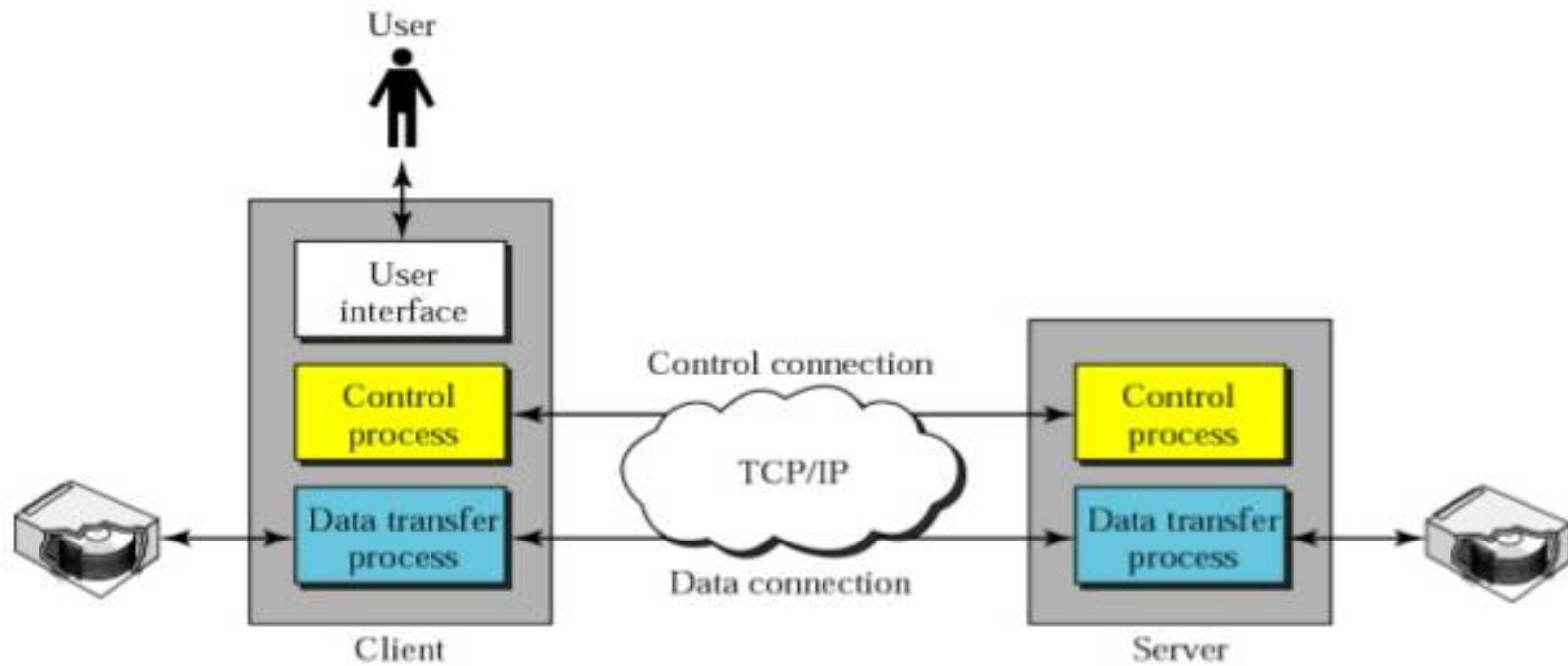
Cont...

- To show some user-friendly commands used by some FTP interfaces.
- To discuss anonymous FTP and its application.
- To discuss how file transfer can be done using a secure channel.
- To discuss TFTP as a simple file transfer protocol without the complexities and sophistication of FTP.
- To discuss five types of TFTP messages and their applications.
- To discuss the sorcerer's apprentice bug related to TFTP's flow- and error-control mechanisms.
- To show how TFTP can be used in conjunction with DHCP to initialize devices by downloading configuration files.

Cont...

- File Transfer Protocol (FTP) is the standard mechanism 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 text and data.
- Two systems may have different directory structures.
- All of these problems have been solved by FTP in a very simple and elegant approach.

Figure : *FTP*



Cont...

- **Figure 1 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.
- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.

Cont...

- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transferred.
- It open search time commands that involve transferring files are used, and it closes when the file is transferred.

4.2 Connections

- The two FTP connections, control and data, use different strategies and different port numbers.
- **A) Control Connection**
- The control connection is created in the same way as other application programs described so far.
- There are two steps:
- **1. The server issues a passive open on the well-known port 21 and waits for a client.**
- **2. The client uses an ephemeral port and issues an active open.**

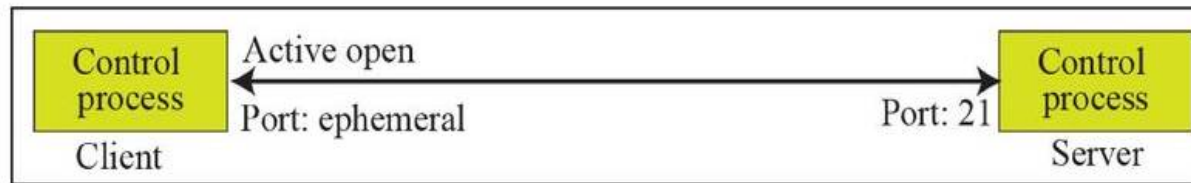
Cont...

- The connection remains open during the entire process.
- The service type, used by the IP protocol, is minimize delay because this is an interactive connection between a user (human) and a server.
- The user types commands and expects to receive responses without significant delay.
- Figure 2 shows the initial connection between the server and the client.

Figure : *Opening the control connection*



a. First, passive open by server



b. Later, active open by client

Cont...

- **B) Data Connection:**
- The data connection uses the well-known port 20 at the server site.
- However, the creation of a data connection is different from what we have seen so far.
- **The following shows how FTP creates a data connection:**
- 1. The client, not the server, 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.

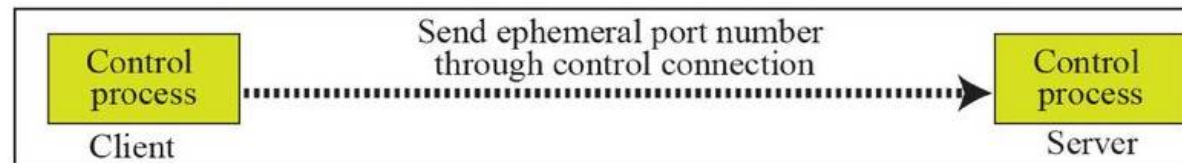
Cont...

- 2. The client sends this port number to the server using the PORT command (we will discuss this command shortly).
- 3. The server receives the port number and issues an active open using the well known port 20 and the received ephemeral port number.
- The steps for creating the initial data connection are shown in Figure 3.
- Later we will see that these steps are changed if the PASV command is used.

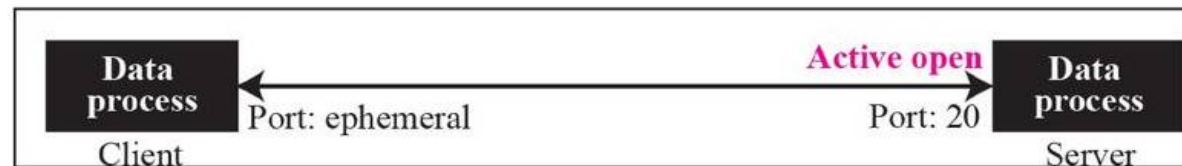
Figure : *Creating the data connection*



a. First, passive open by client



b. Second, sending of ephemeral port



c. Third, active open by server

4.3 Communication

- The FTP client and server, which run on different computers, must communicate with each other.
- These two computers may use different operating systems, different character sets, different file structures, and different file formats.
- FTP must make this heterogeneity compatible.
- FTP has two different approaches, one for the control connection and one for the data connection.
- We will study each approach separately.

Communication over Control Connection

- FTP uses the same approach as TELNET or SMTP to communicate across the control connection.
- It uses the NVT ASCII character set (see Figure 4).
- 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 command or response is only one short line so we need not worry about file format or file

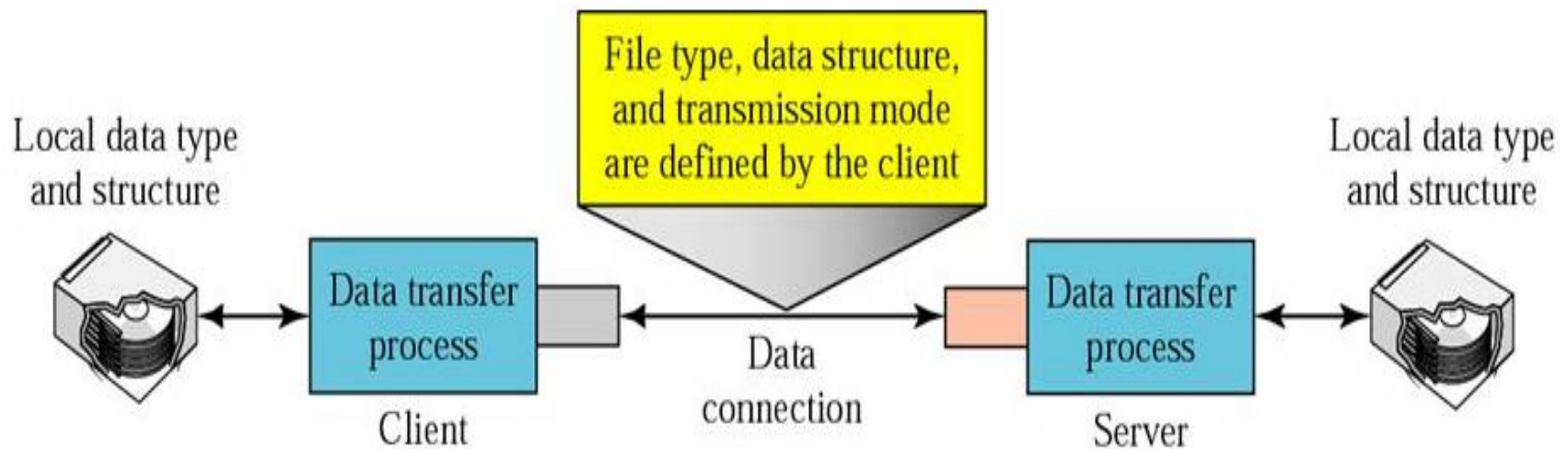
Figure : *Using the control connection*



Communication over Data Connection

- The purpose and implementation of the data connection are different from that 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** (see Figure 5).

Figure : *Using the data connection*



File Type

- **FTP can transfer one of the following file types across the data connection:**
- **ASCII file:** This is the default format for transferring text files. Each character is encoded using NVT ASCII. The sender transforms the file from its own representation into NVT ASCII characters and the receiver transforms the NVT ASCII characters to its own representation.
- **EBCDIC file:** If one or both ends of the connection use EBCDIC encoding, the file can be transferred using EBCDIC encoding.
- **Image file.** This is the default format for transferring binary files. The file is sent as continuous streams of bits without any interpretation or encoding. used to transfer binary files such as compiled programs.

Cont...

- If the file is encoded in ASCII or EBCDIC, another attribute must be added to define the printability of the file.
- **a. Nonprint:** This is the default format for transferring a text file. The file contains no vertical specifications for printing. This means that the file cannot be printed without further processing because there are no characters to be interpreted for vertical movement of the print head. This format is used for files that will be stored and processed later.
- **b. TELNET:** In this format the file contains NVT ASCII vertical characters such as CR (carriage return), LF (line feed), NL (new line), and VT (vertical tab). The file is printable after transfer.

Data Structure

- FTP can transfer a file across the data connection using one of the following interpretations about the structure of the data:
- File structure (default). The file has no structure. It is a continuous stream of bytes.
- Record structure. The file is divided into records. This can be used only with text files.
- 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.

Transmission Mode

- FTP can transfer a file across the data connection using one of
- the following three transmission modes:
 - **Stream mode:** This is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes. TCP is responsible for chopping data into segments of appropriate size.
 - **Block mode:** Data can be delivered from FTP to TCP in blocks. In this case, 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.

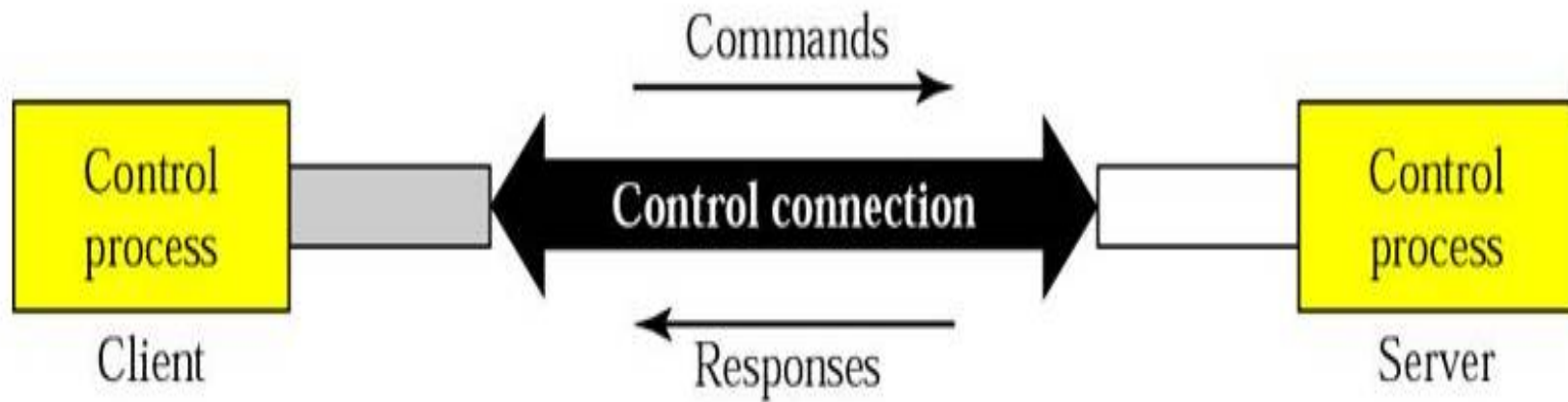
Cont...

- **Compressed mode:** If the file is big, the data can be compressed. The compression method normally used is run-length encoding. In this method, consecutive appearances of a data unit are replaced by one occurrence and the number of repetitions. In a text file, this is usually spaces (blanks). In a binary file, null characters are usually compressed.

4.4 Command Processing

- FTP uses the control connection to establish a communication between the client control process and the server control process.
- During this communication, the commands are sent from the client to the server and the responses are sent from the server to the client (see Figure 6).

Figure : *Command processing*



Commands

- 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.
- commands are divided into six groups:
- 1.access commands, 2.file management commands, 3.data formatting commands, 4.port defining commands,5. file transferring commands, and 6.miscellaneous commands.
- **Access commands:** These commands let the user access the remote system.
- Table 1 lists common commands in this group.

Table : *Access commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
USER	User id	User information
PASS	User password	Password
ACCT	Account to be charged	Account information
REIN		Reinitialize
QUIT		Log out of the system
ABOR		Abort the previous command

Cont...

- **File management commands:** These commands let the user access the file system on the remote computer.
- They allow the user to navigate through the directory structure, create new directories, delete files, and so on.
- Table 2 gives common commands in this group.

Table : *File management commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
CWD	Directory name	Change to another directory
CDUP		Change to the parent directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
NLIST	Directory name	List the names of subdirectories or files without other attributes
MKD	Directory name	Create a new directory
PWD		Display name of current directory
RMD	Directory name	Delete a directory
RNFR	File name (old file name)	Identify a file to be renamed
RNTO	File name (new file name)	Rename the file
SMNT	File system name	Mount a file system

Cont...

- **Data formatting commands:** These commands let the user define the data structure, file type, and transmission mode.
- The defined format is then used by the file transfer commands.
- Table 3 shows common commands in this group.

Table : *Data formatting commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
TYPE	A (ASCII), E (EBCDIC), I (Image), N (Nonprint), or T (TELNET)	Define the file type and if necessary the print format
STRU	F (File), R (Record), or P (Page)	Define the organization of the data
MODE	S (Stream), B (Block), or C (Compressed)	Define the transmission mode

Cont...

- **□ Port defining commands:**
- These commands define the port number for the data connection on the client site.
- There are two methods to do this. In the first method, using the PORT command, the client can choose an ephemeral port number and send it to the server using a passive open.
- The server uses that port number and creates an active open.
- In the second method, using the PASV command, the client
- just asks the server to first choose a port number.
- The server does a passive open on that port and sends the port number in the response.

Table : *Port defining commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
PORT	6-digit identifier	Client chooses a port
PASV		Server chooses a port

Cont...

- **File transfer commands :**
- These commands actually let the user transfer files.
- Table 5 lists common commands in this group.

Table : *File transfer commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
RETR	File name(s)	Retrieve files; file(s) are transferred from server to client
STOR	File name(s)	Store files; file(s) are transferred from client to server
APPE	File name(s)	Similar to STOR, but if file exists, data must be appended to it
STOU	File name(s)	Same as STOR, but file name will be unique in the directory
ALLO	File name(s)	Allocate storage space for files at the server
REST	File name(s)	Position file marker at a specified data point
STAT	File name(s)	Return status of files

Cont...

- **Miscellaneous commands:**
- These commands deliver information to the FTP user at the client site.
- Table 21.6 shows common commands in this group.

Table : *Miscellaneous commands*

TABLE 1.10 MISCELLANEOUS COMMANDS

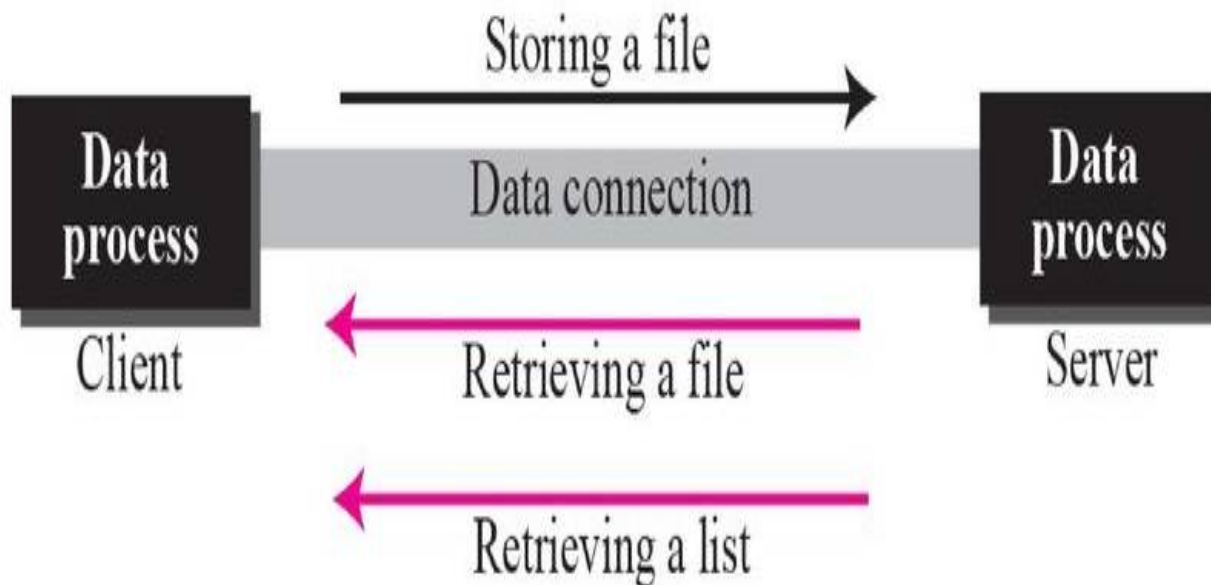
<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
HELP		Ask information about the server
NOOP		Check if server is alive
SITE	Commands	Specify the site-specific commands
SYST		Ask about operating system used by the server

4.5 File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection.
- However, we should remember that file transfer in FTP means one of three things (see Figure 7).
- A file is to be copied from the server to the client (download).
- 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 (upload). This is called *storing a file*. It is done under the supervision of the *STOR* command.

Cont...

- □ 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.
- Note that FTP treats a list of directory or file names as a file.
- It is sent over the data connection.



File transfer Example

- The following shows an actual FTP session that parallels Example .
- 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 black background show data transfer.

4.6 User interface

- The normal user does not use TELNET commands as defined above.
- Usually, the operating system (UNIX, for example) defines an interface with user-friendly commands.
- An example of such a set of commands can be found in Table 20.6.
- Note that the interface is responsible for translating the user-friendly commands to the previously defined commands in the protocol.

Examples of interface commands

<i>Command</i>	<i>Meaning</i>	<i>Command</i>	<i>Meaning</i>
open	Connect to a remote computer	set	Set the operating parameters
close	Close the connection	status	Display the status information
display	Show the operating parameters	send	Send special characters
mode	Change to line or character mode	quit	Exit TELNET

4.7 Anonymous FTP

- To use FTP, a user needs an account (user name) and a password on the remote server.
- Some sites have a set of files available for public access.
- To access these files, a user does not need to have an account or password. Instead, the user can use *anonymous* as the user name and *guest as the password*.
- User access to the system is very limited. Some sites allow anonymous users only a subset of commands.
- For example, most sites allow the user to copy some files, but do not allow navigation through the directories.

Anonymous FTP Example

- We show an example of anonymous FTP. We assume that some public data are available at
- **\$ ftp internic.net**
- **Connected to internic.net**
- **220 Server ready**
- **Name: anonymous**
- **331 Guest login OK, send “guest” as password**
- **Password: guest**
- **ftp > pwd**

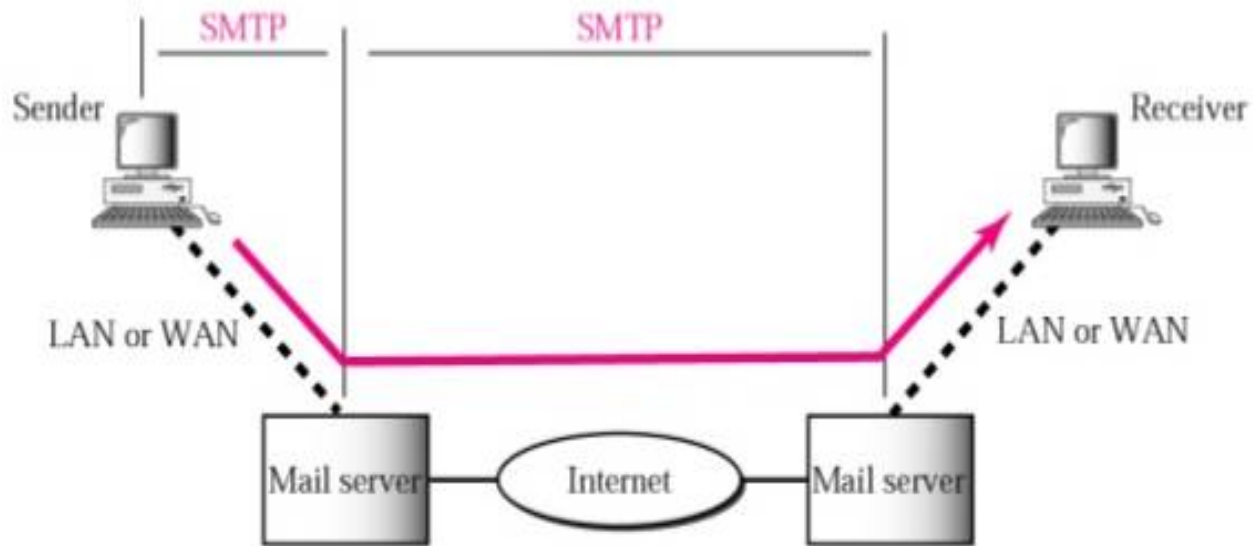
Cont...

- **257 '/' is current directory**
- **ftp > ls**
- **200 OK**
- **150 Opening ASCII mode**
- **bin**
- **...**
- **...**
- **...**
- **ftp > close**
- **221 Goodbye**

4.7 Simple Mail Transfer Protocol (SMTP)

- 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.

Figure : *SMTP range*



Cont...

- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.
- As we will see shortly, another protocol is needed between the mail server and the receiver.
- SMTP simply defines how commands and responses must be sent back and forth.
- Each network is free to choose a software package for implementation.
- We will discuss the mechanism of mail transfer by SMTP in the remainder of the section.

Cont...

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server (see Figure 1).
- Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

4.8 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 local mailboxes on the user computers.

User Agent Types

- **There are two types of user agents: command-driven and GUI-based.**
- Command-driven user agents belong to the early days of electronic mail.
- They are still present as the underlying user agents in servers.
- A command-driven user agent normally accepts a one character command from the keyboard to perform its task.
- For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients.

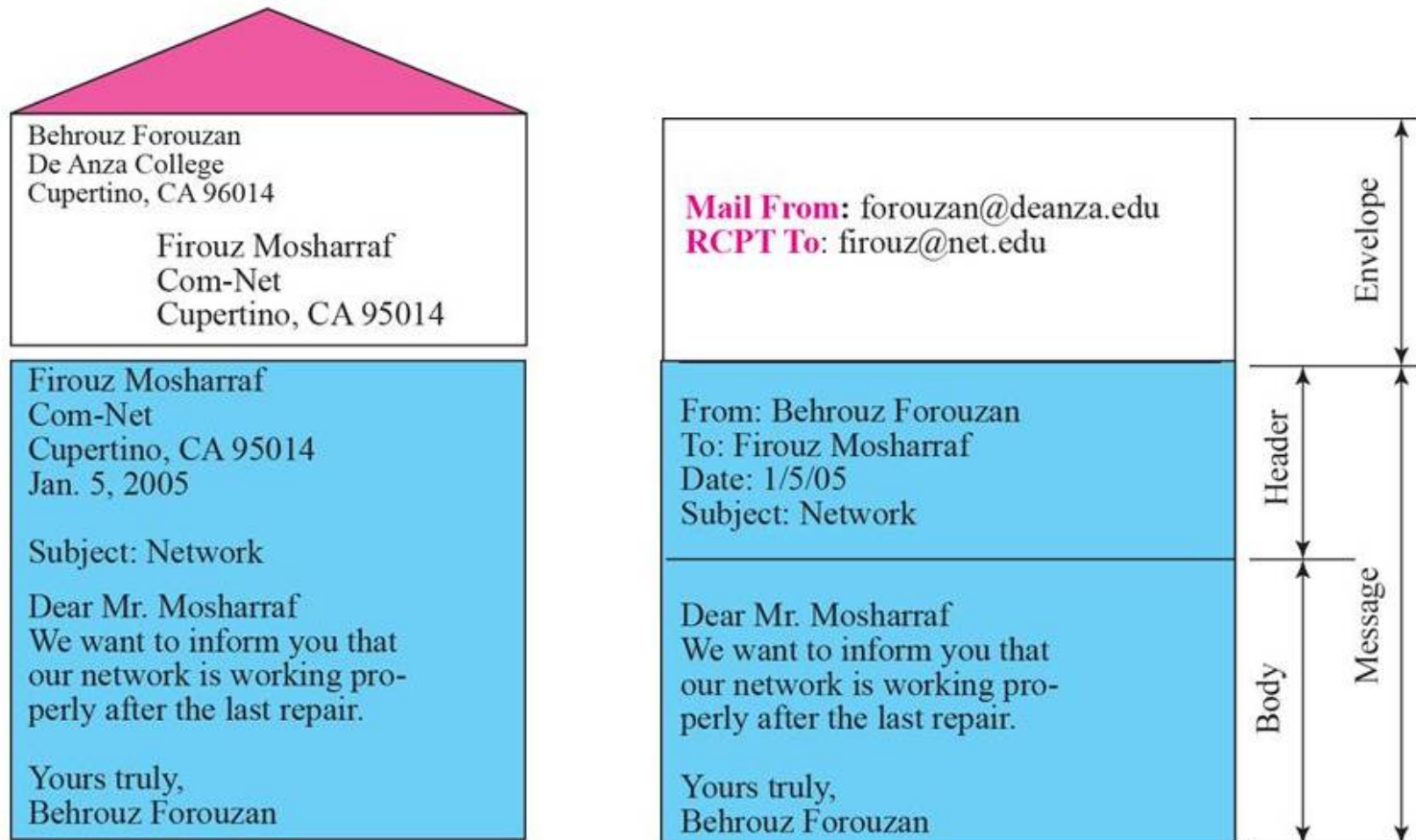
Cont...

- 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.

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 (see Figure 2).
- **Envelope**
- The envelope usually contains the sender address, the receiver address, and other information.

Figure : *Format of an e-mail*



Cont...

Message

- 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.

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.

4.9 Addresses

- **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 3).

Local Part

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

Figure : *E-mail address*



Cont...

- **Domain Name**
- 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 (for example, the name of the organization).

Cont...

- **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.
- If there is no mailing list for the alias, the name itself is the receiving address and a single message is delivered to the mail transfer entity.

4.10 Delayed Delivery

- Most spam and nearly all e-mail borne virii are delivered directly to your server by way of specialized SMTP client software, optimized for sending out large amounts of mail in a very short time. Such clients are commonly known as RATWARE
- Certain *Mail Transport Agents* automatically treat such SMTP protocol violations as *synchronization errors*, and immediately drop the incoming connection. If you happen to be using such an MTA, you may already see a lot of entries to this effect in your log files. In fact, chances are that if you perform any time-consuming checks (such as DNS checks) prior to presenting the initial SMTP banner, such errors will occur frequently, as ratware clients simply do not take the time to wait for your server to come alive

4.11 Aliases

- Email aliases can be created on a mail server. The mail server simply forwards email messages addressed to an email alias on to another, the specified email address. An email alias may be used to create a simple replacement for a long or difficult-to-remember email address.
- It can also be used to create a generic email address such as `webmaster@example.com` and `info@example.com`.

4.12 Mail transfer agent (MTA)

- Within the Internet email system, a **message transfer agent** or **mail transfer agent (MTA)** or **mail relay** is software that transfers electronic mail messages from one computer to another using SMTP. The terms **mail server**, **mail exchanger**, and **MX host** are also used in some contexts.
- A message transfer agent receives mail from either another MTA, a mail submission agent (MSA), or a mail user agent (MUA). The transmission details are specified by the Simple Mail Transfer Protocol (SMTP).

4.13 Commands and Responses

Commands

- Commands are sent from the client to the server. The format of a command is shown below:

Keyword: argument(s)

- It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands listed in Table 1 and described in more detail below.



Table :*Commands*

<i>Keyword</i>	<i>Argument(s)</i>	<i>Keyword</i>	<i>Argument(s)</i>
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
VERFY	Name of recipient	SMAL FROM	Intended recipient

Cont...

- **HELO.** This command is used by the client to identify itself. The argument is the domain name of the client host. The format is **HELO: challenger.atc.fhda.edu**
- **MAIL FROM.** This command is used by the client to identify the sender of the message. The argument is the e-mail address of the sender (local part plus the domain name). The format is **MAILFROMforouzan@challenger.atc.fhda.edu**
- **RCPT TO.** This command is used by the client to identify the intended recipient of the message. The argument is the e-mail address of the recipient. If there are multiple recipients, the command is repeated. The format is
RCPT TO: betsy@mcgraw-hill.com

Cont...

- **DATA.** This command is used to send the actual message. All lines that follow the DATA command are treated as the mail message. The message is terminated by a line containing just one period. The format is **DATA**
 - **This is the message to be sent to the McGraw-Hill Company.**
- **QUIT.** This command terminates the message. The format is **QUIT**
- **RSET.** This command aborts the current mail transaction. The stored information about the sender and recipient is deleted. The connection will be reset. **RSET**

Cont...

- **VRFY.** This command is used to verify the address of the recipient, which is sent as the argument. The sender can ask the receiver to confirm that a name identifies a valid recipient. Its format is **VRFY: betsy@mcgraw-hill.com**
- **NOOP.** This command is used by the client to check the status of the recipient. It requires an answer from the recipient. Its format is **NOOP**
- **TURN.** This command lets the sender and the recipient switch positions, whereby the sender becomes the recipient and vice versa. However, most SMTP implementations today do not support this feature. The format is **TURN**

Cont...

- **EXPN.** This command asks the receiving host to expand the mailing list sent as the arguments and to return the mailbox addresses of the recipients that comprise the list. The format is **EXPN: x y z**
- **HELP.** This command asks the recipient to send information about the command sent as the argument. The format is **HELP: mail**
- **SEND FROM.** This command specifies that the mail is to be delivered to the terminal of the recipient, and not the mailbox. If the recipient is not logged in, the mail is bounced back. The argument is the address of the sender. The format is **SEND FROM: forouzan@fhda.atc.**

Cont...

- **SMOL FROM.** This command specifies that the mail is to be delivered to the terminal or the mailbox of the recipient. This means that if the recipient is logged in, the mail is delivered only to the terminal. If the recipient is not logged in, the mail is delivered to the mailbox. The argument is the address of the sender. The format is

SMOL FROM: forouzan@fhda.atc.edu

- **SMAL FROM.** This command specifies that the mail is to be delivered to the terminal and the mailbox of the recipient. This means that if the recipient is logged in, the mail is delivered to the terminal and the mailbox. If the recipient is not logged in, the mail is delivered only to the mailbox. The argument is the address of the sender. The format is

SMAL FROM: forouzan@fhda.atc.edu

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 2 lists some of the responses.

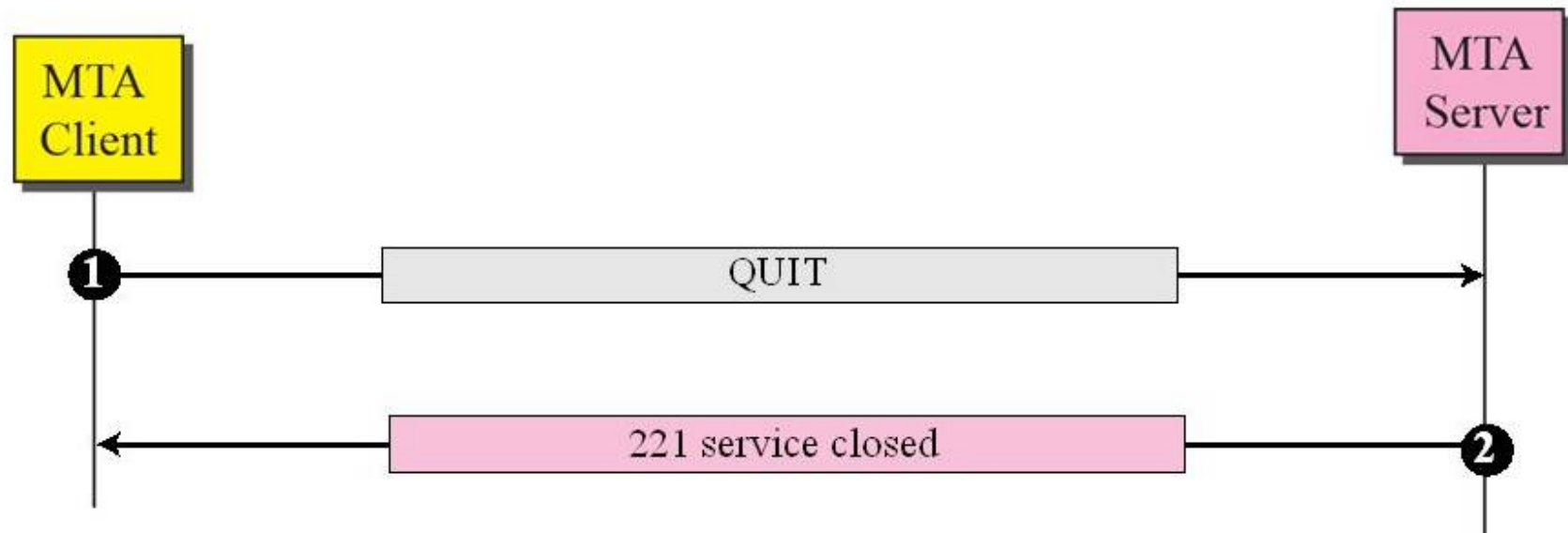
Table : *Responses*

<i>Code</i>	<i>Description</i>
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

4.13 Mail Transfer Phases

- The process of transferring a mail message occurs in three phases:
 - A) connection establishment,
 - B) mail transfer, and
 - C) connection termination.
-
- **A) 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, which are illustrated in Figure 5.

Figure 5 : *Connection establishment*



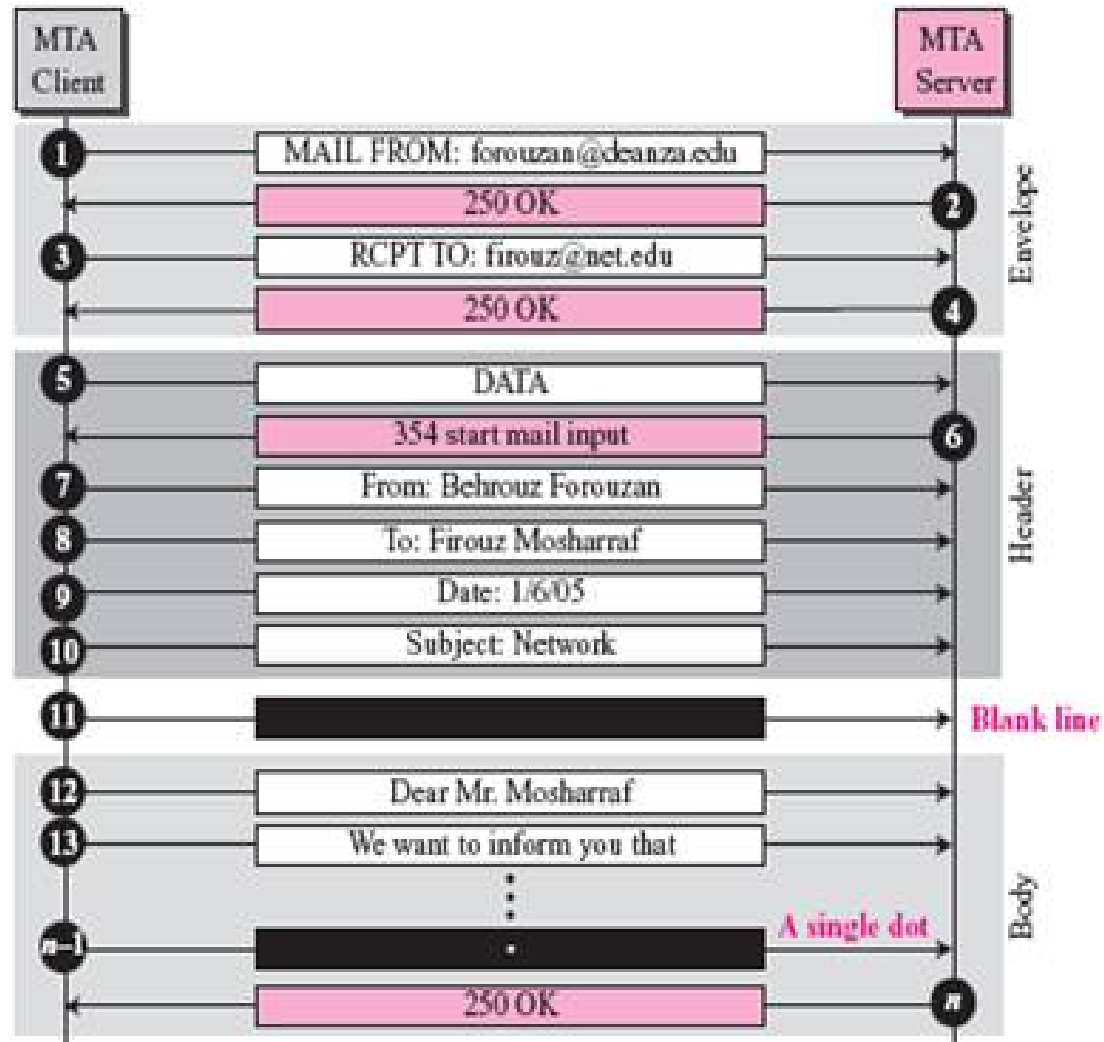
Cont...

- 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. Remember that during TCP connection establishment, the sender and receiver know each other through their IP addresses.
- 3. The server responds with code 250 (request command completed) or some other code depending on the situation.

B) *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 (see Figure 6).
- **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.

Figure : *Message transfer*



Cont...

- 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.

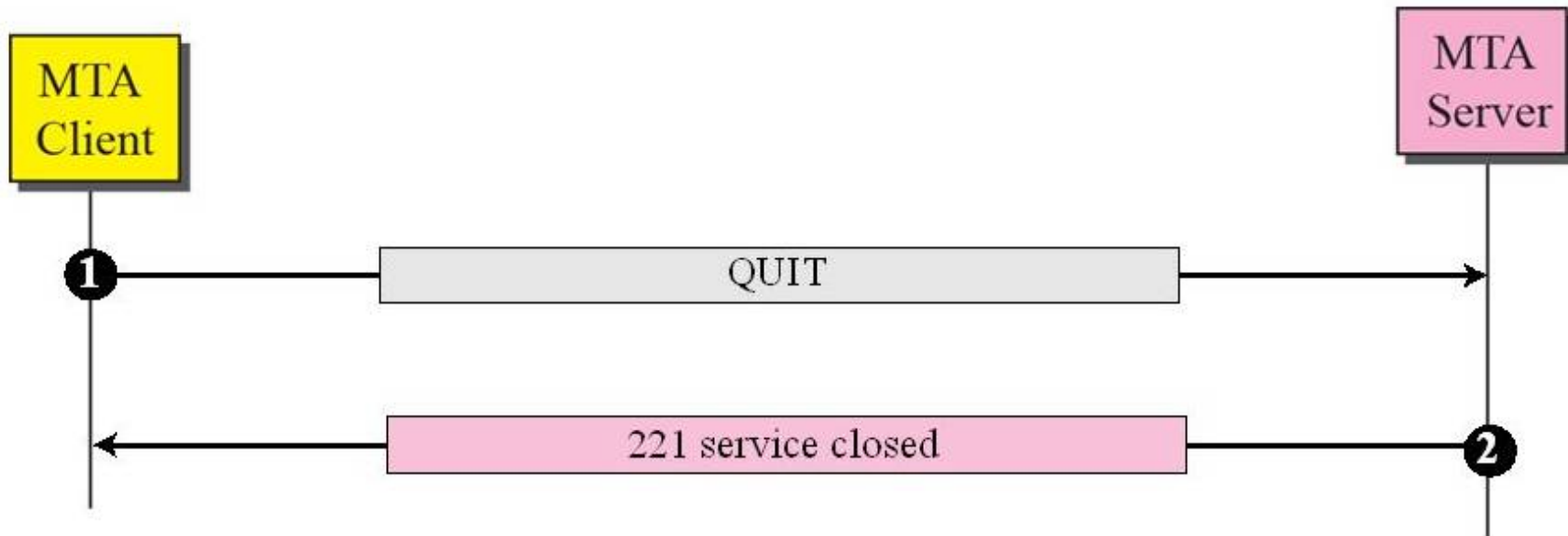
Cont...

- 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.

C) Connection Termination

- After the message is transferred successfully, the client terminates the connection.
- This phase involves two steps (see Figure 7).
 - 1. The client sends the QUIT command.
 - 2. The server responds with code 221 or some other appropriate code. After the connection termination phase, the TCP connection must be closed.

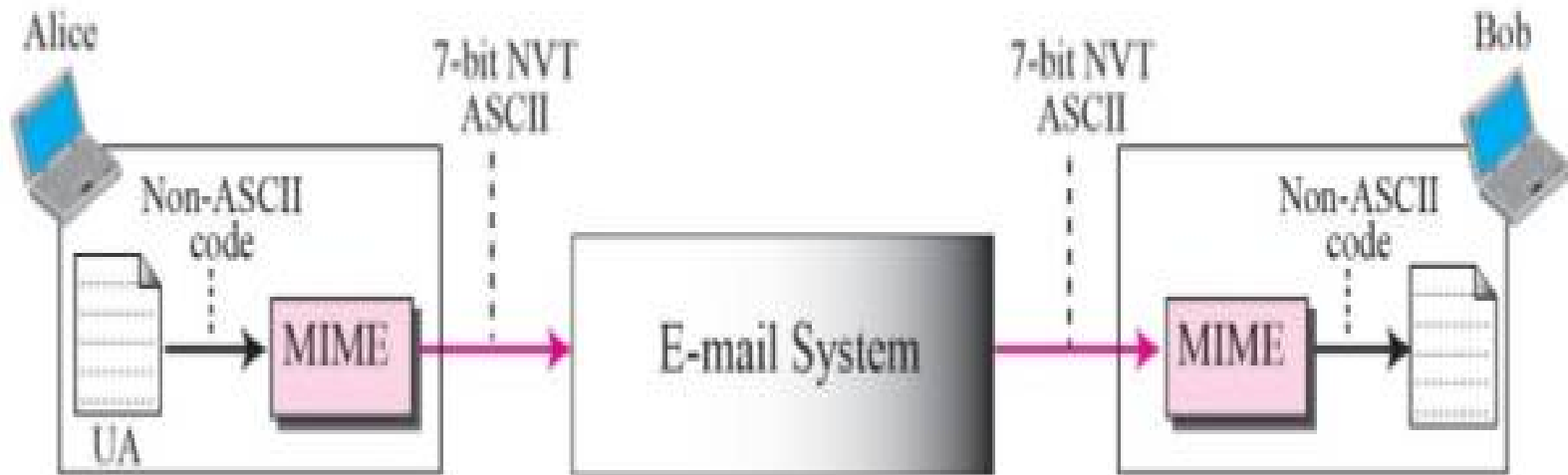
Figure : *Connection termination*



4.14 Multipurpose Internet Mail Extensions (MIME)

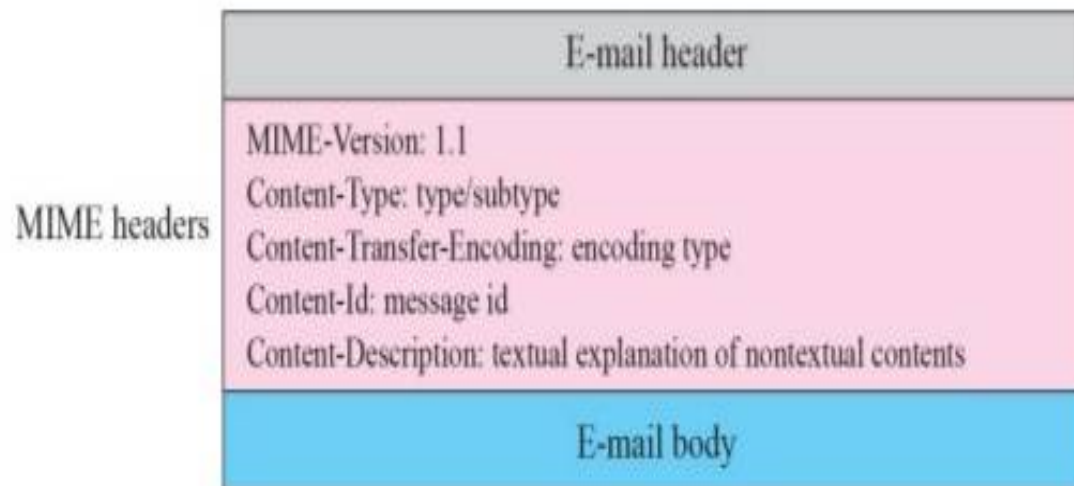
- **Multipurpose Internet Mail Extensions (MIME)** is a supplementary protocol that allows non-ASCII data to be sent through e-mail.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through
- the Internet.
- The message at the receiving site is transformed back to the original data.
- We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa, as shown in Figure 23.15.

Figure : *MIME*



MIME Headers

- MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:
- **1. MIME-Version**
- **2. Content-Type**
- **3. Content-Transfer-Encoding**
- **4. Content-Id**
- **5. Content-Description**



Cont...

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. Depending on the subtype, the header may contain other parameters. MIME allows seven different types of data, listed in Table 2.

Data Types and Subtypes in MIME

Table 23.3 *Data Types and Subtypes in MIME*

<i>Type</i>	<i>Subtype</i>	<i>Description</i>
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)

Cont....

- **Text.** The original message is in 7-bit ASCII format and no transformation by MIME is needed. There are two subtypes currently used, plain and HTML.
- **Multipart.** The body contains multiple, independent parts. The multipart header needs to define the boundary between each part.
- A parameter is used for this purpose.
- The parameter is a string token that comes before each part; it is on a separate line by itself and is preceded by two hyphens.
- The body is terminated using the boundary token, again preceded by two hyphens and the terminated with two hyphens.

Content-Transfer-Encoding

- This header defines the method used to encode the messages into 0s and 1s for transport:

Content-Transfer-Encoding: <type>

- The five types of encoding methods are listed in Table 3.
- **7bit.** This is 7-bit NVT ASCII encoding. Although no special transformation is needed, the length of the line should not exceed 1,000 characters.

Table *Content-Transfer-Encoding*

Table 23.4 *Content-Transfer-Encoding*

<i>Type</i>	<i>Description</i>
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

Cont...

- **8bit.** This is 8-bit encoding. Non-ASCII characters can be sent, but the length of the line still should not exceed 1,000 characters. MIME does not do any encoding here; the underlying SMTP protocol must be able to transfer 8-bit non-ASCII characters. It is, therefore, not recommended. Base64 and quoted-printable types are preferable.
- **Binary.** This is 8-bit encoding. Non-ASCII characters can be sent, and the length of the line can exceed 1,000 characters. MIME does not do any encoding here; the underlying SMTP protocol must be able to transfer binary data. It is, therefore, not recommended. Base64 and quoted-printable types are preferable.

Cont...

- **Base64.**
- This is a solution for sending data made of bytes when the highest bit is not necessarily zero.
- Base64 transforms this type of data to printable characters.
- which can then be sent as ASCII characters or any type of character set supported by the underlying mail transfer mechanism.
- Base64 divides the binary data (made of streams of bits) into 24-bit blocks.
- Each block is then divided into four sections, each made of 6 bits (see Figure 9).

Cont...

- **Quoted-printable.** Base64 is a redundant encoding scheme; that is, 24 bits become four characters, and eventually are sent as 32 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.
- 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.
- Figure 10 shows an example.

Figure : *Base64*

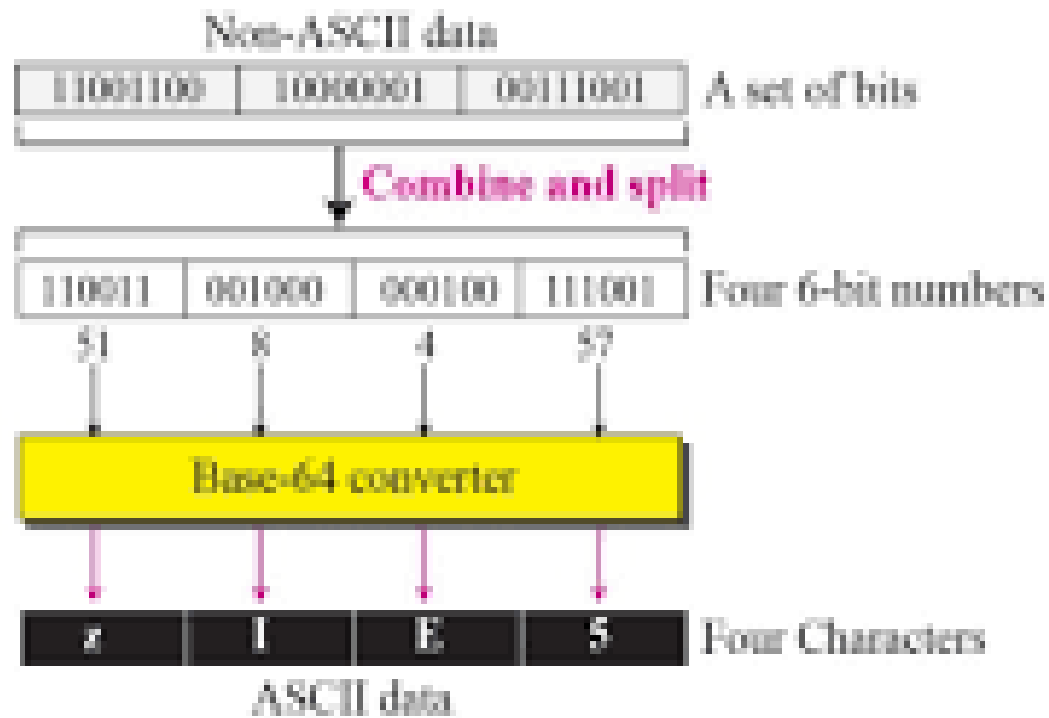
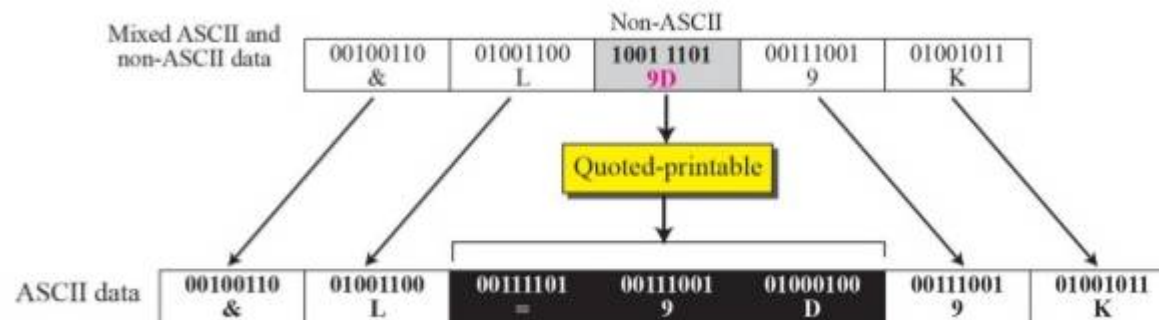


Figure : *Quoted-printable*



4.15 Mail Delivery

- Basically your outgoing messages are handled by a machine called **SMTP server** which takes care of delivering them to the intended recipients. Unfortunately, sometimes things do not work correctly: for instance, think of the well-known, annoying “**Mail Delivery Failure**” notifications from the Mail Delivery System itself.
- If your account list is OK (you have deleted all the wrong addresses and you send communications only to users who have given their permission) but you keep receiving this kind of messages, the issue probably lies right in your own SMTP or **outgoing mail server**.

It may well be that the IP from which you're mailing out is shared with a **spammer**, or it's even been **blacklisted**. Unfortunately that can happen if you're using a common server like the one associated by default to a Gmail or Hotmail account. Solution: opt for a **professional email delivery servicelike** [turboSMTP](#), capable to maximize the capacity of your messages to reach the inboxes. As it's becoming more and more difficult to get emails correctly delivered – because of spam filters and firewalls and other issues – a monitored, highly reliable SMTP server will give you a huge help.

4.16 Mail Access Protocols

- The Simple Mail Transfer Protocol is a communication protocol for electronic mail transmission. As an Internet standard, SMTP was first defined in 1982 by RFC 821, and updated in 2008 by RFC 5321 to Extended SMTP additions, which is the protocol variety in widespread use today.
- Post Office Protocol version 3 (POP3) is a standard mail protocol used to **receive emails** from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline. Note, that when you use POP3 to connect to your email account, messages are downloaded locally and removed from the email server. This means that if you access your account from multiple locations, that may not be the best option.