Application Layer

WWW, HTTP, Electronic Mail, FTP, Telnet

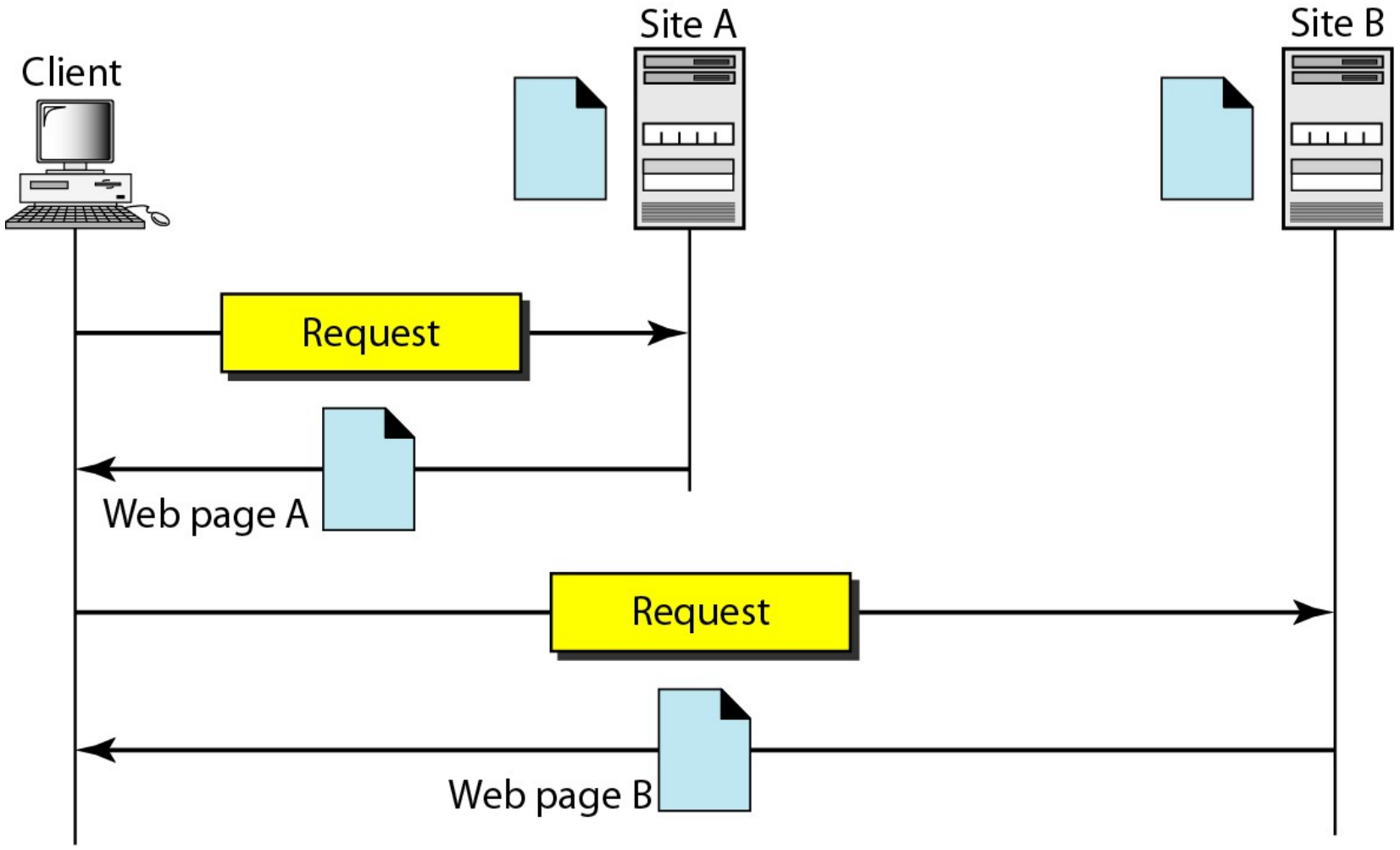
WWW- World Wide Web

- It is a repository of information linked together from points all over the world.
- WWW project was initiated by CERN (European Laboratory for Particle Physics) to create a system to handle distributed resources necessary for scientific research.

Architecture

- The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server.

Architecture of WWW



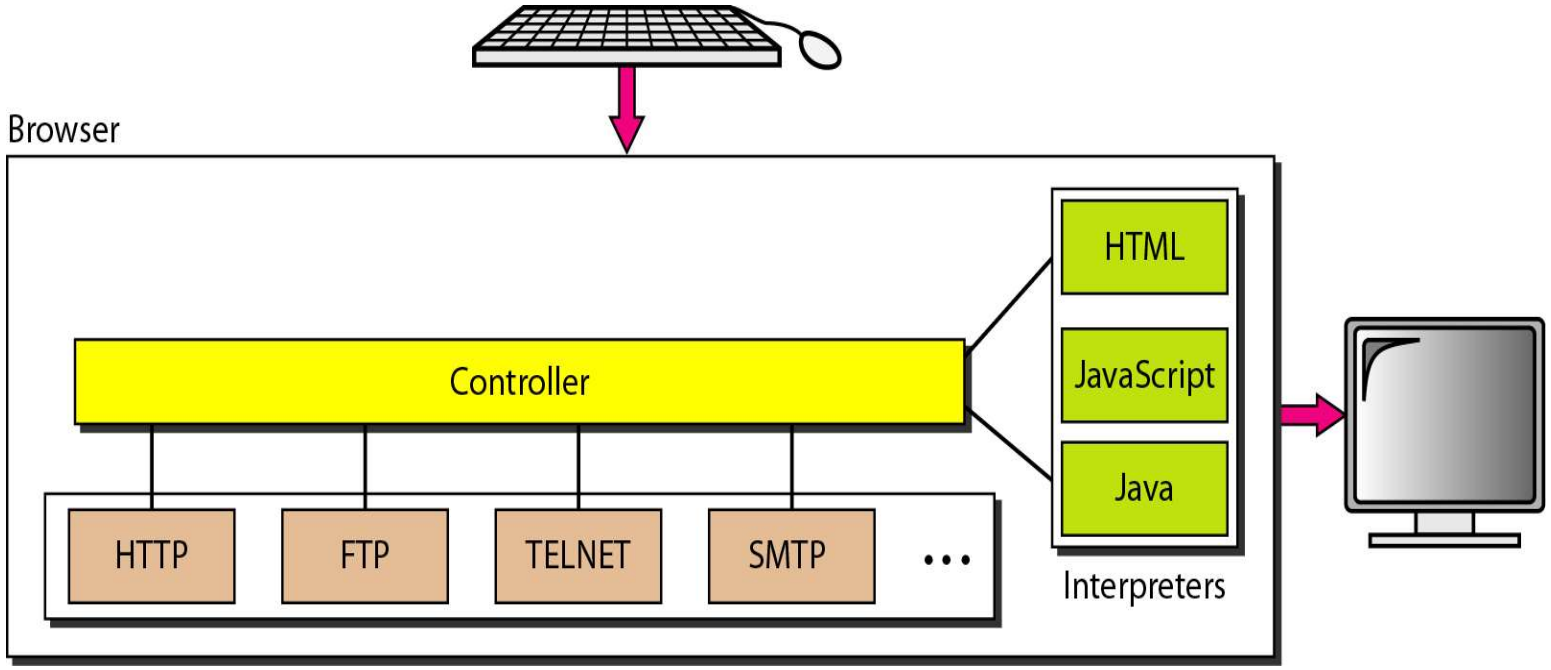
- Each site holds one or more documents, referred to as *Web pages*.
- *Each Web page can* contain a link to other pages in the same site or at other sites
- The client sends the request: Address of the web page - called URL.

Client

- A variety of vendors offer commercial browsers that interpret and display a Web document.
- Each browser usually consists of three parts: a controller, client protocol, and interpreters.

- The controller receives input from the keyboard or the mouse and uses the client programs to access the document.
- After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.
- The client protocol FTP or HTTP.
- The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

Browser

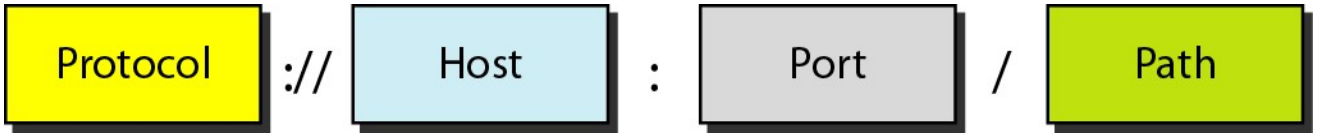

Browser
HTML
JavaScript
Java
Controller
HTTP
FTP
TELNET
SMTP
...
Interpreters

Server

- The Web page is stored at the server.
- Each time a client request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.
- A server can also become more efficient through multithreading or multiprocessing.
- In this case, a server can answer more than one request at a time.

Uniform Resource Locator

- A client that wants to access a Web page needs the address.
- To facilitate the access of documents distributed throughout the world, HTTP uses locators.
- The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet.
- The URL defines four things: **protocol, host computer, port, and path**



- The *protocol is the client/server program used to retrieve the document for example: FTP, HTTP.*
- The host is the computer on which the information is located.
  - Web pages are usually stored in computers, these are given alias names that usually begin with the characters "www".

- The URL can optionally contain the port number of the server.
- If the *port is* included, it is inserted between the host and the path, and it is separated from the host by a colon.
- Path is the pathname of the file where the information is located.

Cookies

- The World Wide Web was originally designed as a stateless entity.
- A client sends a request; a server responds. Their relationship is over.
- Today the Web has other functions:
  - Some websites need to allow access to registered clients only.
  - Websites are being used as electronic stores.
  - Some websites are used as portals: the user selects the Web pages he wants to see.
  - Some websites are just advertising.

Creation and storage of cookies

- The cookies are created at the server and stored at the client.
- When a server receives a request from a client, it stores information about the client in a file or a string.
  - The information may include the domain name of the client.
  - The contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.

- The server includes the cookie in the response that it sends to the client.
- When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

Using cookies

- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server.
- The contents of the cookie are never read by the browser or disclosed to the user.

A cookie is used for the four previously mentioned purposes:

- The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time.
- An electronic store (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it into a cart, a cookie contains information about the item.
- A Web portal uses the cookie in a similar way.
- A cookie is also used by advertising agencies.

Web documents

- The documents in the WWWcan be grouped into three broad categories: static, dynamic, and active.
- The category is based on the time at which the contents of the document are determined.

Static

- Static documents are fixed-content documents that are created and stored in a server.
- The client can get only a copy of the document.
- When a client accesses the document, a copy of the document is sent.
- The user can then use a browsing program to display the document.

HTML

- It is a language for creating Web pages.
- The Web pages are formatted using HTML for interpretation by a browser.
- The basic unit of HTML are tags.
- HTML allows us to embed formatting instructions in the file itself.
- The instructions are included with the text.
- In this way, any browser can read the instructions and format the text accordingly.

- A Web page is made up of two parts: the head and the body.
- The head is the first part of a Web page.
- The head contains the title of the page and other parameters that the browser will use.
- The actual contents of a page are in the body, which includes the text and the tags.
- Additional information for tag is given with attributes.

Dynamic Documents

- A dynamic document is created by a Web server whenever a browser requests the document.
- When a request arrives, the Web server runs an application program or a script that creates the dynamic document.
- The server returns the output of the program or script as a response to the browser that requested the document.
- Example of a dynamic document is the retrieval of the time and date from a server.

Common Gateway Interface(CGI)

- The CGI is a technology that creates and handles dynamic documents.
- CGI is a set of standards that defines how a dynamic document is written, how data are input to the program, and how the output result is used.
- It allows programmers to use any of several languages such as C, C++, Boume Shell, Kom Shell, C Shell, Tcl, or Perl.
- CGI defines a set of rules and terms that the programmer must follow.

- The term **common** in CGI the standard set of rules that is common to any language or platform.
- The term **Gateway:** CGI program can be used to access other resources such as databases, graphical packages, and so on.
- The **interface** : is a set of predefined terms, variables, calls, and so on that can be used in any CGI program.

Input to CGI

- The input from a browser to a server is sent by using a form.
- If the information in a form is small (such as a word), it can be appended to the URL after a question mark. https://172.16.0.1:1003/keepalive?010f0c090d040210
- When the server receives the URL, it uses the part of the URL before the question mark to access the program to be run.
- It interprets the part after the question mark as the input sent by the client.
- It stores this string in a variable.

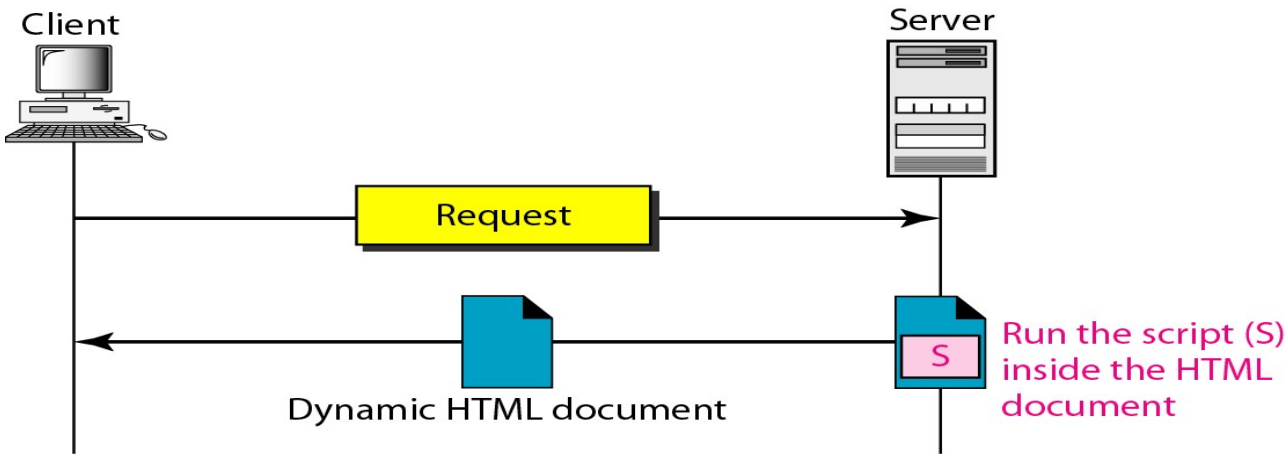
Output

- The whole idea of CGI is to execute a CGI program at the server site and send the output to the client (browser).
- The output is usually plain text or a text with HTML structures; however, the output can be a variety of other things.
- It can be graphics or binary data, a status code, instructions to the browser to cache the result, or instructions to the server to send an existing document instead of the actual output.

- The output of the CGI program always consists of two parts: a header and a body.
- The header is separated by a blank line from the body.
- The header is used by the browser to interpret the body.

Scripting technologies for dynamic documents

- CGI is inefficient to create dynamic web documents.



- A few technologies have been involved in creating dynamic documents using scripts.
- Among the most common are Hypertext Preprocessor (pHP), which uses the Perl language;
- Java Server Pages (JSP), which uses the Java language for scripting; Active
- Server Pages (ASP), a Microsoft product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document.

- Dynamic documents are sometimes referred to as server-site dynamic documents.

Active Documents

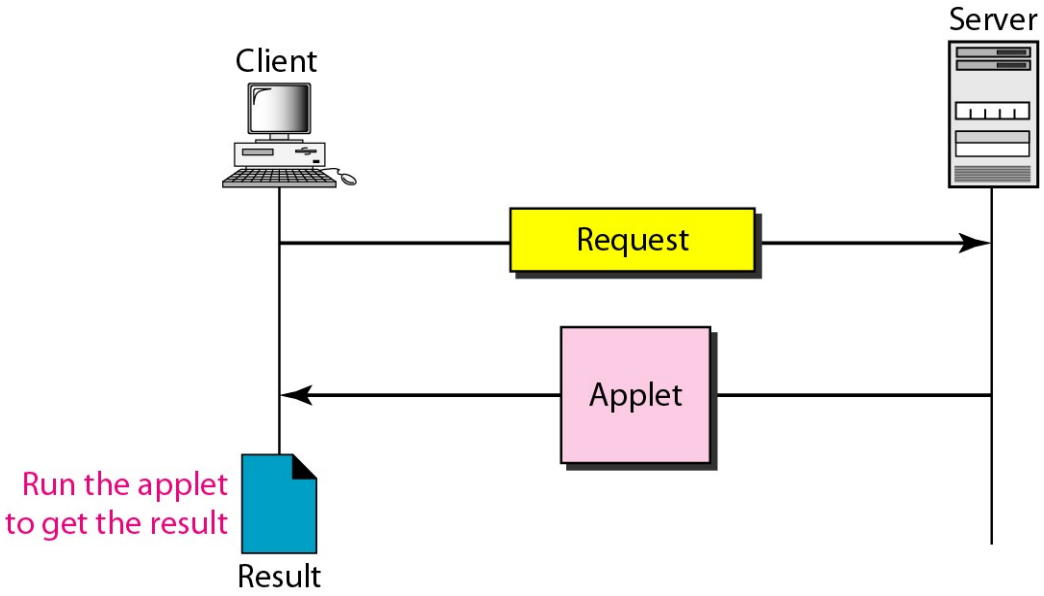
- A program or a script that runs at the client side are called active documents.
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user.

Java Applets

- One way to create an active document is to use Java applets.
- An applet is a program written in Java on the server.
- It is compiled and ready to be run.
- The document is in byte-code (binary) format. The client process (browser) creates an instance of this applet and runs it.

- A Java applet can be run by the browser in two ways.
- In the first method, the browser can directly request the Java applet program in the URL and receive the applet in binary form.
- In the second method, the browser can retrieve and run an HTML file that has embedded the address of the applet as a tag.

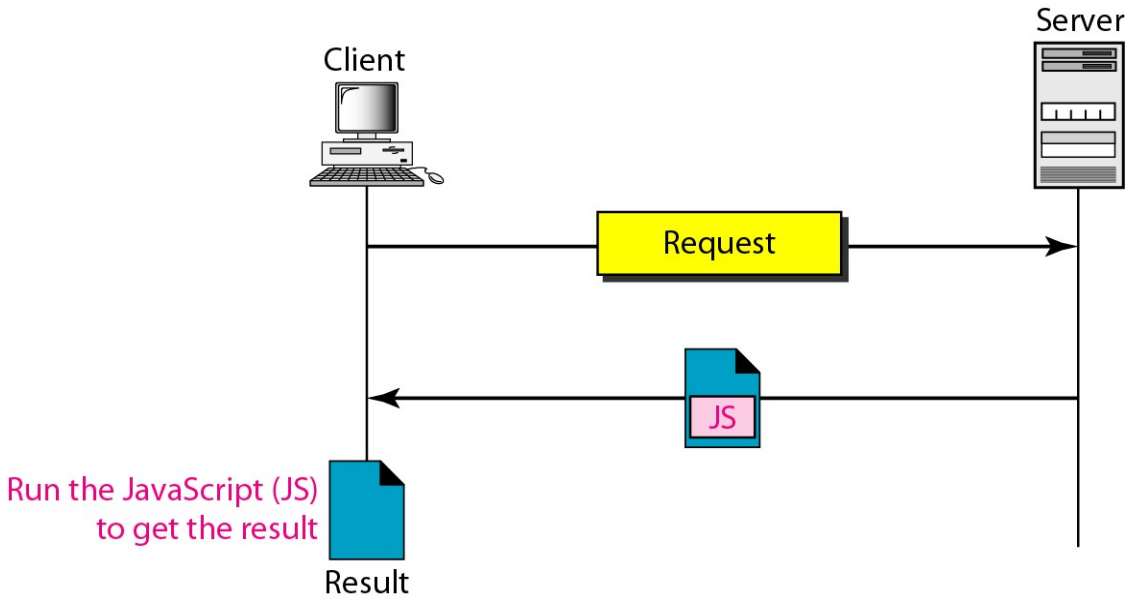
Active document using Java Applets



Java Script

- The idea of scripts in dynamic documents can also be used for active documents.
- If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time.
- The script is in source code (text) and not in binary form.
- The scripting technology used in this case is usually JavaScript.

Active document using client side Script



HTTP: HyperText Transfer Protocol

- It is a protocol used mainly to access data on the World Wide Web.
- HTTP functions as a combination of FTP and SMTP.
- It is similar to FTP because it transfers files and uses the services of TCP. HTTP uses only one TCP connection.
- HTTP is protocol between client and server, hence the messages look like SMTP messages.

- The format of the messages is controlled by MIME-like headers.
- The HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client.
- The commands from the client to the server are embedded in a request message.
- The contents of the requested file or other information are embedded in a response message.
- HTTP uses the services of TCP on well-known port 80.

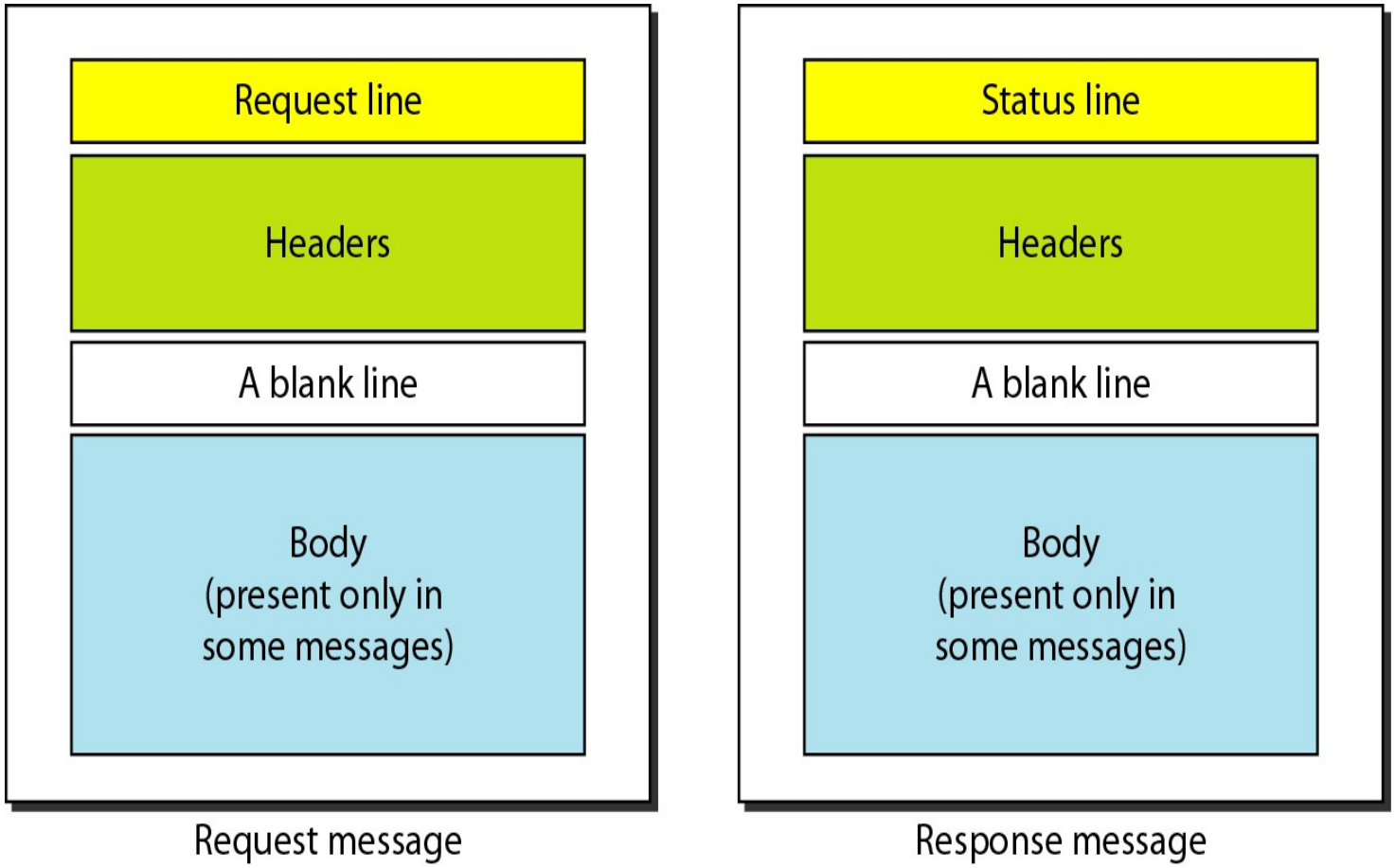
HTTP Transaction

- HTTP uses the services of TCP, HTTP itself is a stateless protocol.
- The client initializes the transaction by sending a request message. The server replies by sending a response.

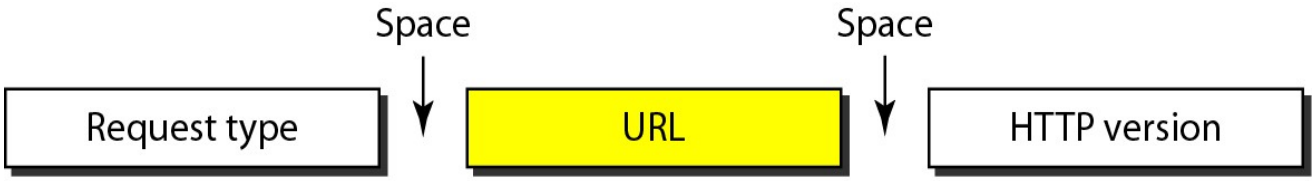
Messages

- Request message: from client to server
- Response message: from server to client
- Both messages have same format: A request/status line, header and body.

Request and Response messages



- The first line in a request message is called a request line.
- The first line in the response message is called the status line.



a. Request line

b. Status line

Request Type:

The request type is categorized into methods:

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

URL:

The address to identify the resource.

Version:

The most current version of HTTP is 1.1.

Status Code:

- Used in response message

Consists of three digits.

- The codes in the 100 range are only informational
- The codes in the 200 range indicate a successful request.
- The codes in the 300 range redirect the client to another URL,
- The codes in the 400 range indicate an error at the client site.
- The codes in the 500 range indicate an error at the server site.

Status Phrase:

- This field is used in the response message. It explains the status code in text form.

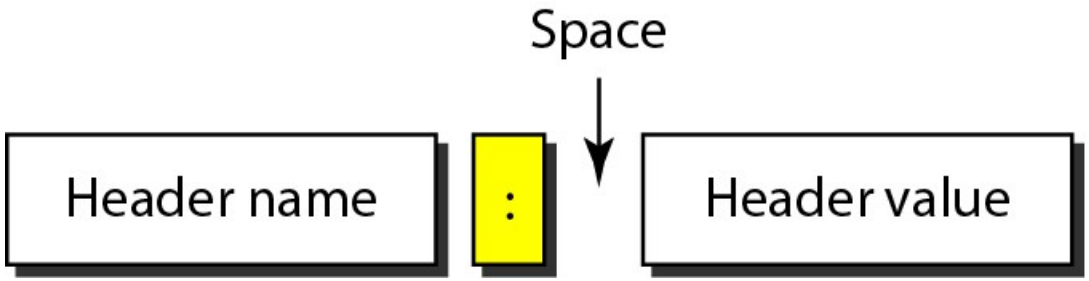
Code	Phrase	Description
100	continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
200	Ok	The request is successful.
201	Created	A new URL is created.
301	Moved Permanently	The requested URL is no longer used by the server.
302	Moved Temporarily	The requested URL has moved temporarily.

Code	Phrase	Description
400	Bad Request	There is a syntax error in the request
401	Unauthorized	The request lacks proper authorization
403	Forbidden	Service Denied
404	Not found	The document is not found
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Header

- The header exchanges additional information between the client and the server.
- For example, the client can request that the document be sent in a special format, or the server can send extra information about the document.

Header Format:



Header Name:

It could be:

- General header, request header, response header, and entity header.
- A request message can contain only general, request, and entity headers.
- A response message, on the other hand, can contain only general, response, and entity headers.

General Header:

- The general header gives general information about the message and can be present in both a request and a response.

Some general headers with their descriptions:

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Request Header:

- The request header can be present only in a request message.
- It specifies the client's configuration and the client's preferred document format.

Header	Description
Accept-encoding	Shows the encoding scheme the client can handle
From	Shows e-mail address of the user
Host	Shows the host and port number of the server
If-modified Since	Sends the document if newer than specified date
User-agent	Identifies the client program

Response header

- The response header can be present only in a response message.
- It specifies the server's configuration and special information about the request.

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Entity header

- The entity header gives information about the body of the document.
- It is mostly present in response messages, also in some request messages, such as POST or PUT methods, that contain a body also use this type of header.

Header	Description
Allow	Lists valid methods that can be used with a URL
Content-Encoding	Specifies Encoding Scheme
Content Length	Specifies length of the document
Content type	Specifies medium type

Body:

The body can be present in a request or response message, it contains the document to be sent or received.

Persistent vs Nonpersistent Connection

- HTTP prior to version 1.1 specified a nonpersistent connection, while a persistent connection is the default in version 1.1.

Nonpersistent Connection

- In a nonpersistent connection, one TCP connection is made for each request/response.

Working:

- The client opens a TCP connection and sends a request.
- The server sends the response and closes the connection.
- The client reads the data until it encounters an end-of-file marker; it then closes the connection.
- For N different pictures in different files, the connection must be opened and closed N times.
- Imposes high overhead on the server: needs N different buffers and requires a slow start procedure each time a connection is opened.

Persistent Connection

- HTTP version 1.1 specifies a persistent connection by default.
- In a persistent connection, the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached.
- The sender usually sends the length of the data with each response.
- For dynamic created documents, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached.

Proxy Server

- HTTP supports proxy servers.
- A proxy server is a computer that keeps copies of responses to recent requests.
- The HTTP client sends a request to the proxy server.
- The proxy server checks its cache.
- If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
- Incoming responses are sent to the proxy server and stored for future requests from other clients.

- The proxy server reduces the load on the original server, decreases traffic, and improves latency.
- To use the proxy server, the client must be configured to access the proxy instead of the target server.

File Transfer

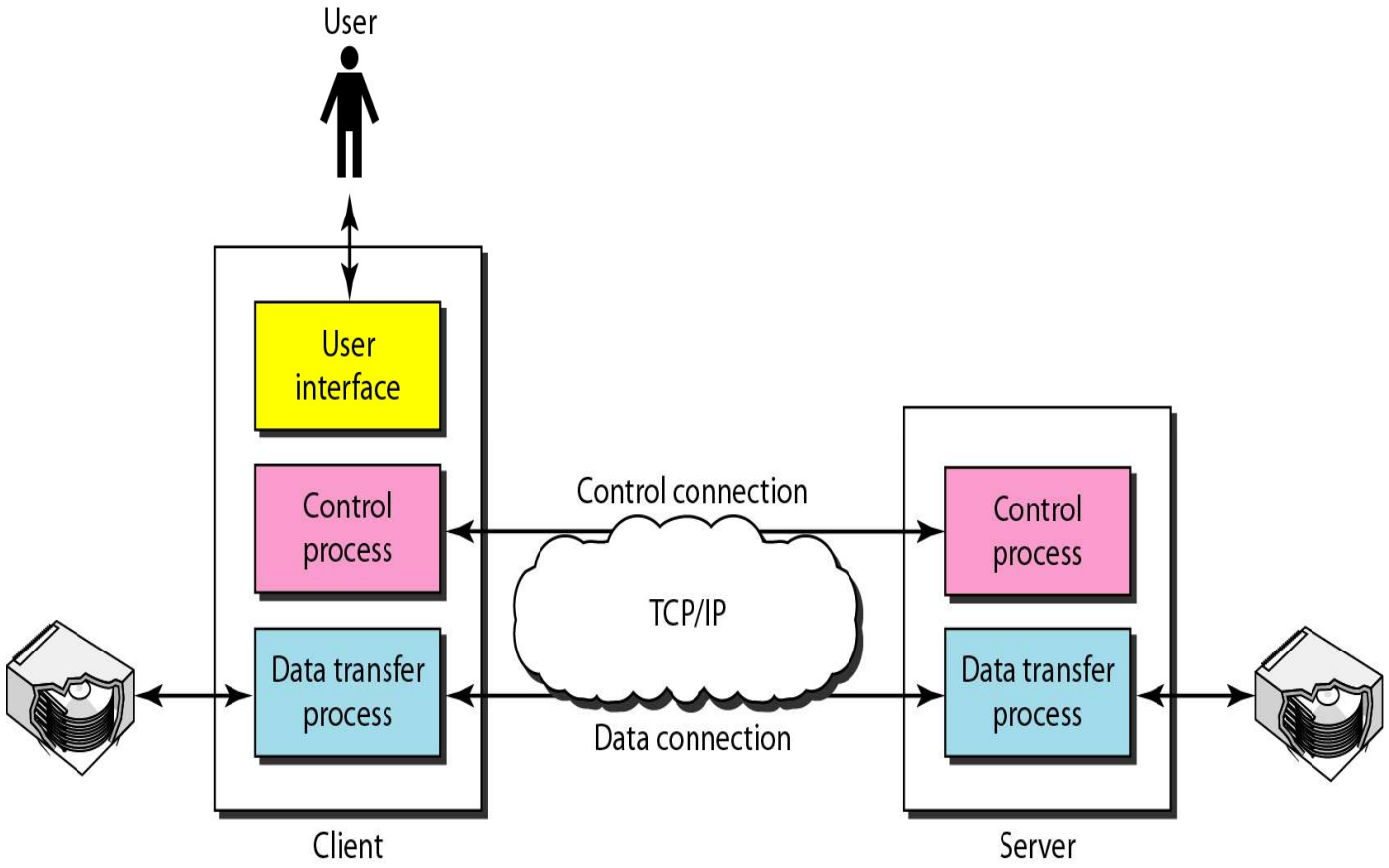
To transfer file from one system to another using File Transfer Protocol(FTP)

FTP

- It is the standard mechanism provided by *TCP/IP for* copying a file from one host to another.
- Complexities associated with file transfer:
  - 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.
- These problems are solved by FTP in very simple and elegant approach

- FTP establishes two connections between the hosts.
- One connection is used for data transfer, the other for control information (commands and responses).
- Separation of commands and data transfer makes FTP more efficient.
- 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.

The FTP protocol

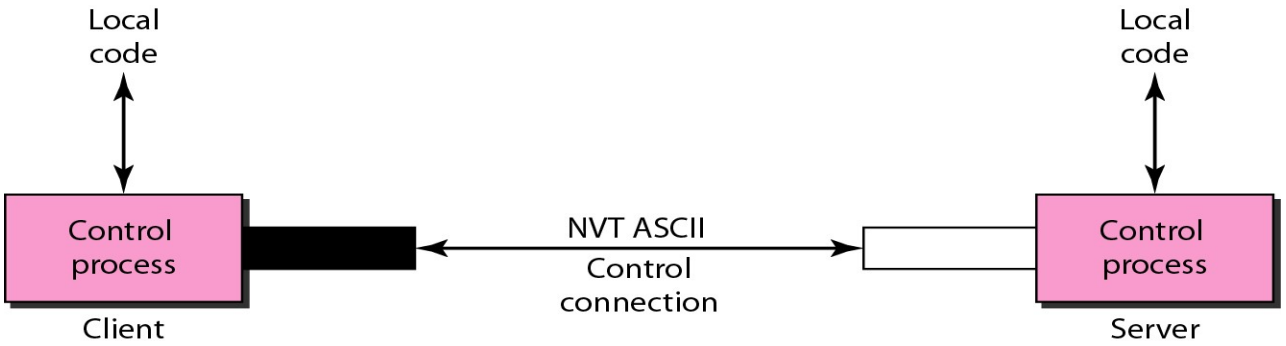


- The client has three components: user interface, client control process, and the client data transfer process.
- The server has two components: the server control process and the server data transfer process.
- The control connection is made between the control processes.
- The data connection is made 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.

Communication over Control Connection

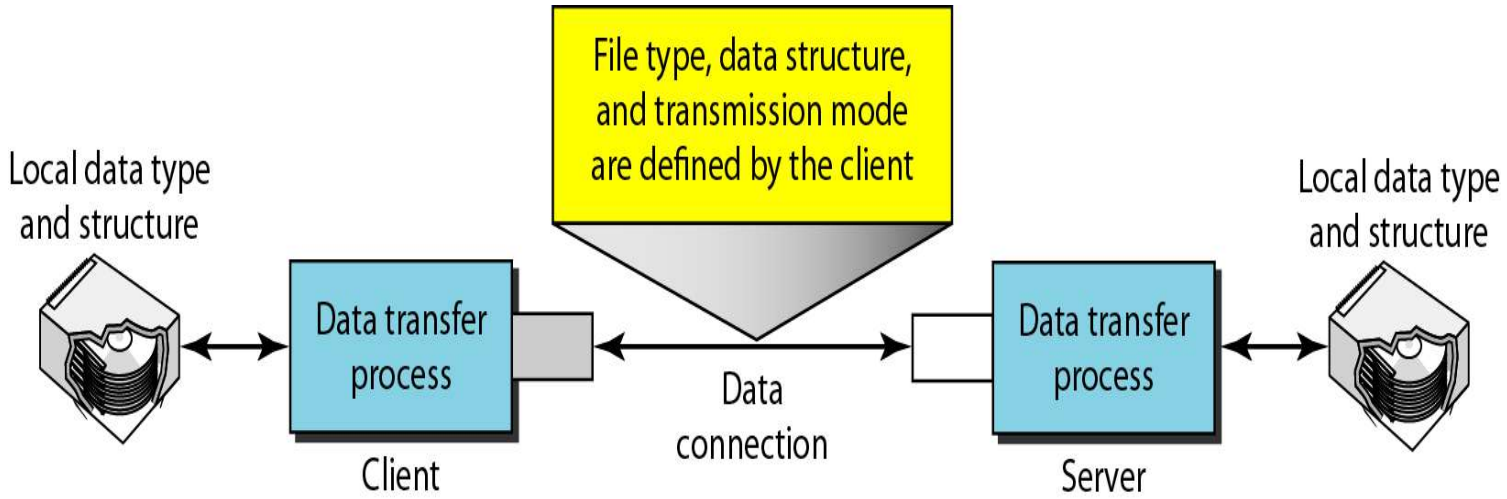
- FTP uses the same approach as SMTP to communicate across the control connection.



Communication over Data 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 heterogeneity problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode.



File Type:

- It could be one of the type- ASCII, EBCDIC or image The ASCII file is the default format for transferring text files.
- If one or both ends of the connection use EBCDIC encoding (the file format used by IBM), the file can be transferred using EBCDIC encoding.
- The image file is the default format for transferring binary files.

Data Structure:

- FTP uses one of interpretations about the structure of the data: file structure, record structure, and page structure.
- File structure: file is a continuous stream of bytes.
- Record structure: file is divided into records. This can be used only with text files.
- Page structure: file is divided into pages, with each page having a page number and a page header.

Transmission Mode:

- FTP can transfer a file across using one of the three transmission modes: stream mode, block mode, and compressed mode.
- The stream mode is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes.
  - End-of-file in this case is the closing of the data connection by the sender.
- In 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 2 bytes define the size of the block in bytes.*

- In the 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.

Electronic Mail

- It is one of the most popular Internet services.
- Initially this application was used to send short text messages, today it is a complex application which allows to send text, audio and video.

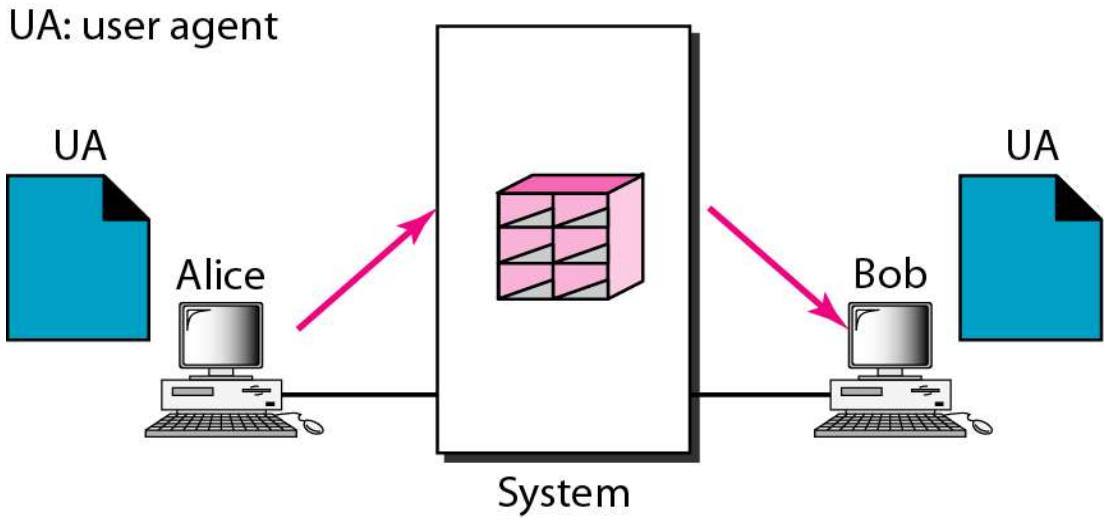
Architecture

- To understand the architecture, let us consider four scenarios

Scenario 1:

- Sender and receiver application programs are connected via 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 a user1 sends message to user2, user1 runs a user agent(UA) program to prepare the message and store in user2's mailbox.

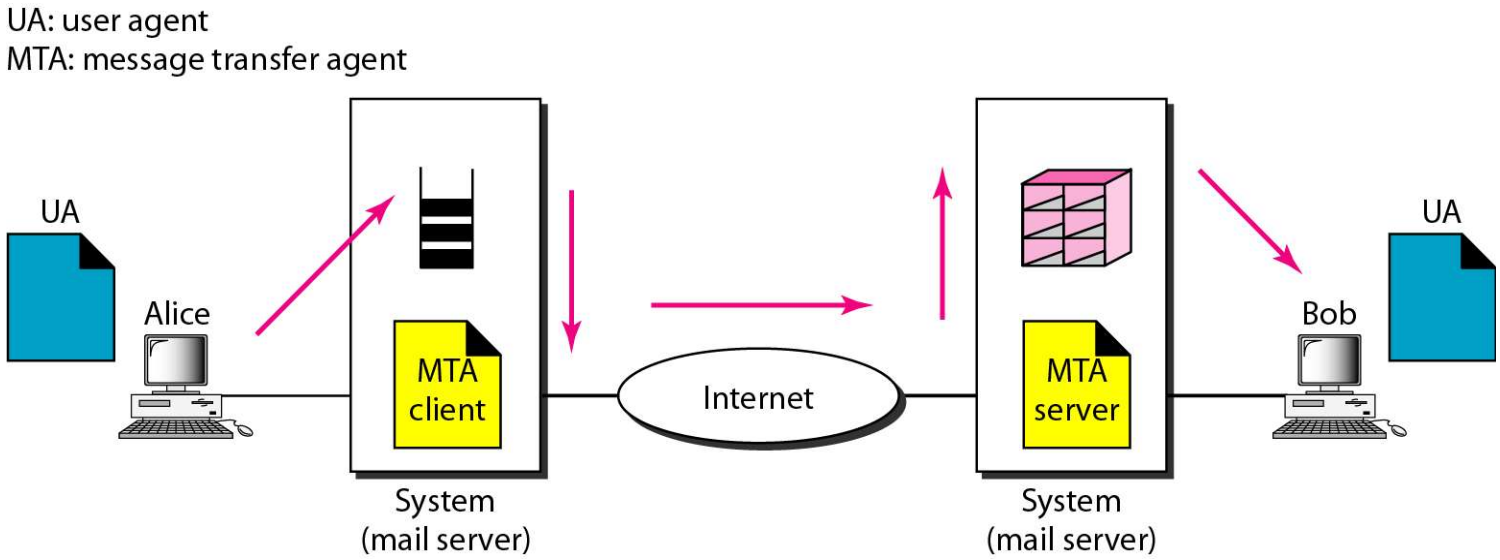
- User2 retrieves and reads the contents of his mailbox at his convenience.
- The message has the sender and recipient mail box address.



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

Scenario 2:

- Sender and receiver application programs are on two different systems.
- The message is to be sent over Internet.
- Here we need UA and Message Transfer Agent(MTA)

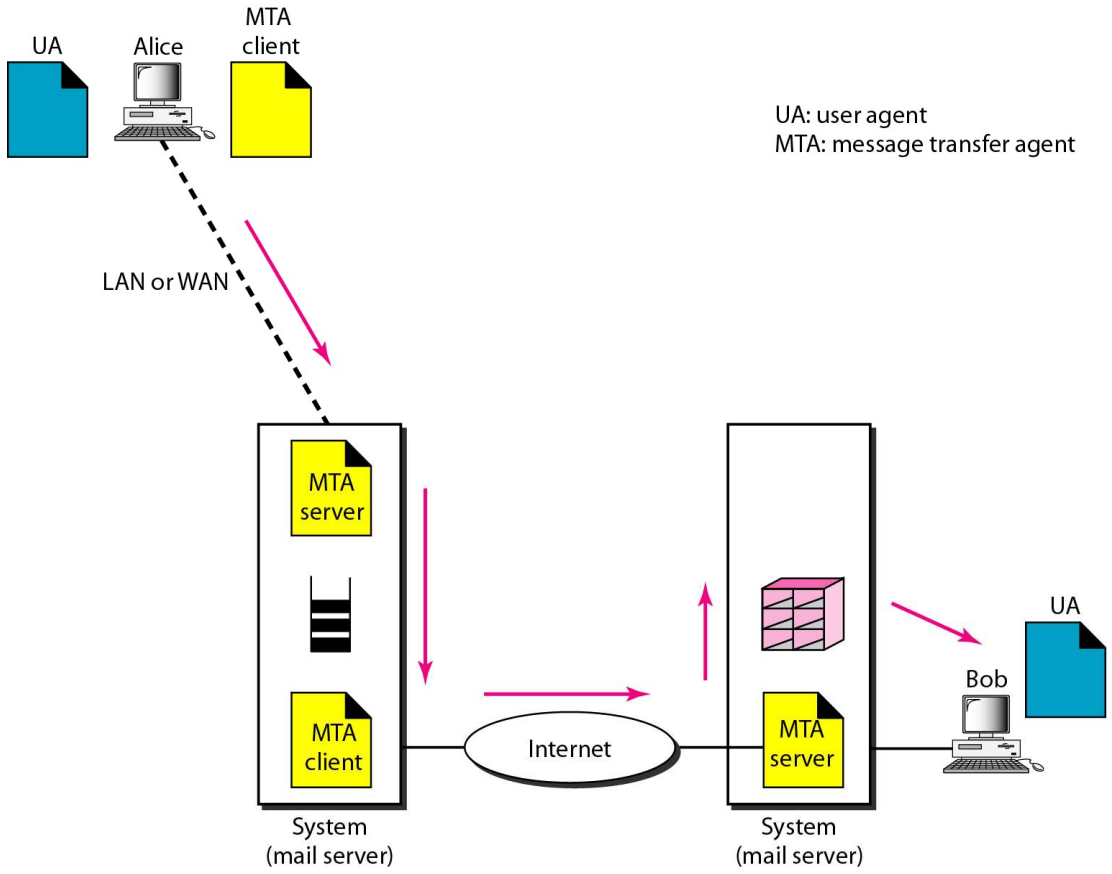


- Two User Agent program: Sends/Retrieve user's message at their own site- this is sometimes called as Mail Server.
- Two Message Transfer Agent(MTA) Programs: one of client and other of server- to transfer message over Internet
- The server needs to run all the time whereas client can be run only when there is message in queue

Third Scenario

- User 1 is connected to a system via point-to-point WAN, or connected via LAN to organizational mail server.
- User 2 is connected to his system.
- 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).

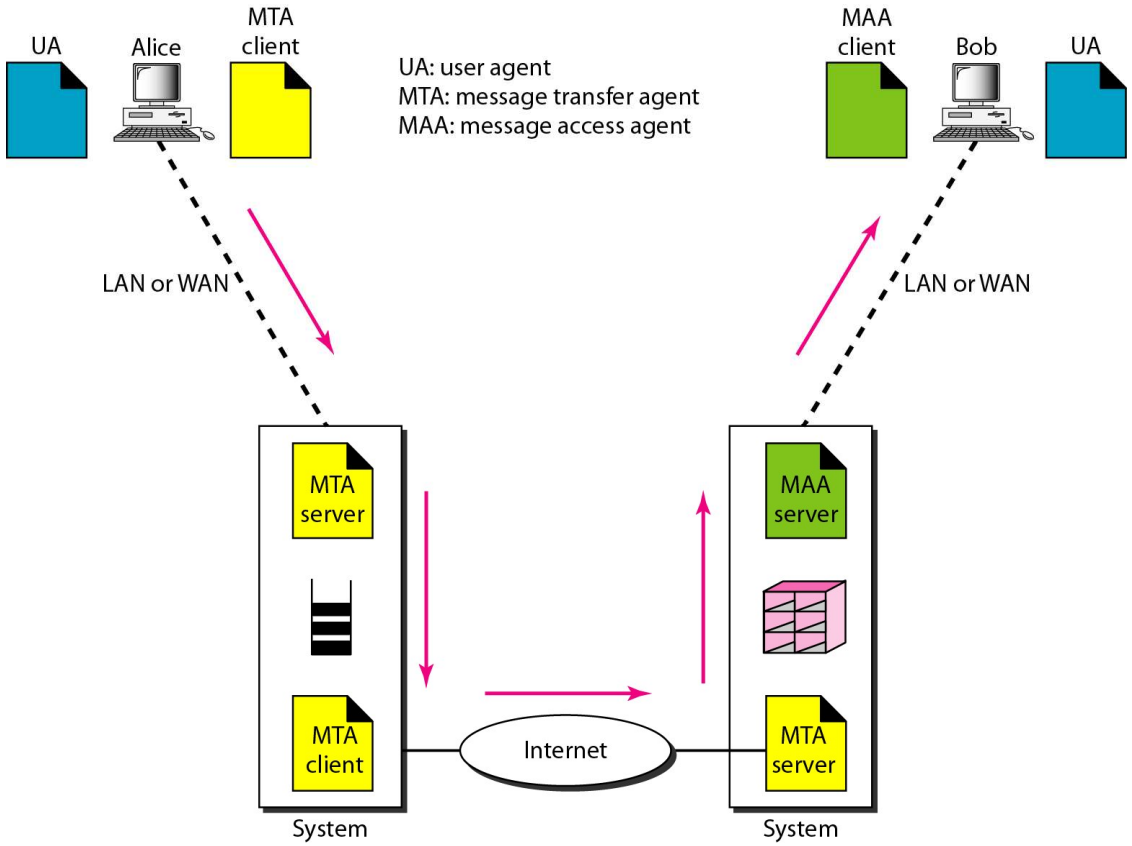
Scenario 3


UA
Alice
MTA client
UA: user agent
MTA: message transfer agent
LAN or WAN
MTA server
MTA client
Internet
MTA server
UA
Bob
System (mail server)
System (mail server)

Scenario 4:

- User1 and User2 are connected to their mail server by a LAN or WAN.
- Another set of client/server agent are needed called as- Message Access Agent(MAA).
- User2 uses MAA client to retrieve message, client then sends a request to MAA server,to transfer all messages
- We need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server).
- This is the most common situation today.

Scenario 4


UA
Alice
MTA client
UA: user agent
MTA: message transfer agent
MAA: message access agent
MAA client
Bob
UA
LAN or WAN
LAN or WAN
MTA server
MTA client
Internet
MAA server
MTA server
System
System

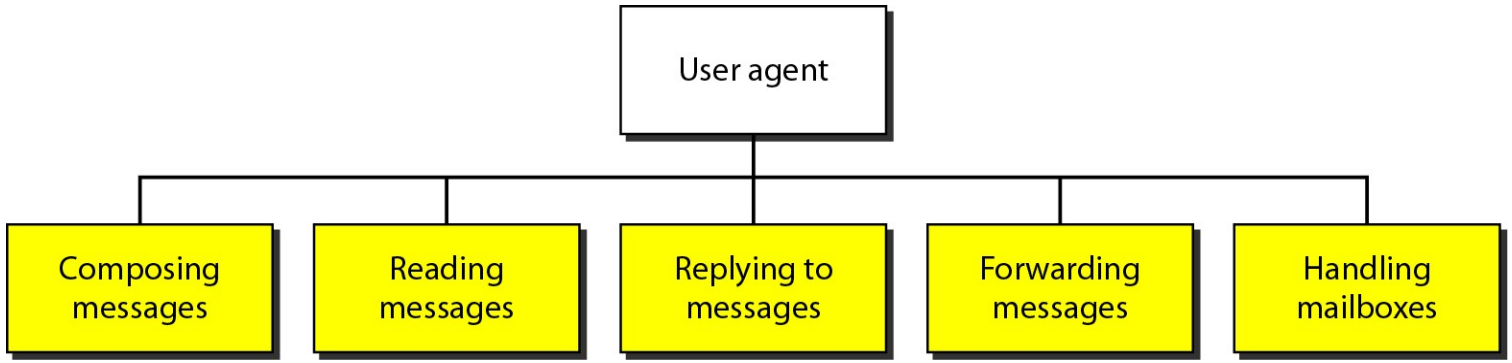
In general a mail transfer system consist of:

- User Agent: It provides service to the user to make the process of sending and receiving a message easier.
- Mail Transfer Agent: SMTP- It actually transfers the messages, client MTA sends a message, server MTA receives messages.
- Mail Access Agent: POP and IMAP- to access the messages locally

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



- Composing Messages: Provide template, perform spell checking, grammar checking
- Reading Messages: read the incoming messages. Display one line summary of the mail in mailbox.
- Replying to messages Replying is defined as sending a message to the sender or recipients of the copy.
- Forwarding Messages: Forwarding is defined as sending the message to a third party, with or without extra comments to third party.

- Handling mailboxes: A user agent normally creates two mailboxes: an inbox and an outbox.
  - Each box has a format.
  - Inbox: received mails
  - Outbox: sent mails.

User Agent Types

There are two types of user agents:

command-driven and GUI-based.

Command driven UA:

- Belong to the early days of electronic mail.
- Accepts a one-character command from the keyboard to perform its task.

Examples: mail, pine, elm

GUI based UA:

- Modem user agents.
- They have graphical components such as icons, menu bars, and windows.

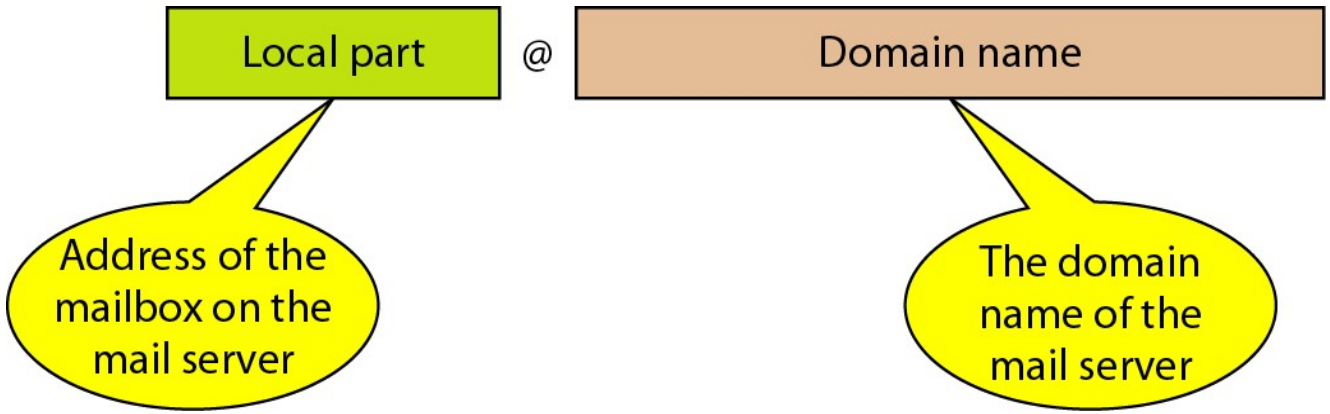
Examples: Eudora, Outlook, and Netscape.

Format of a mail:

- Header: defines the sender, the receiver, the subject of the message.
- Body: Actual information.

Address format:

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

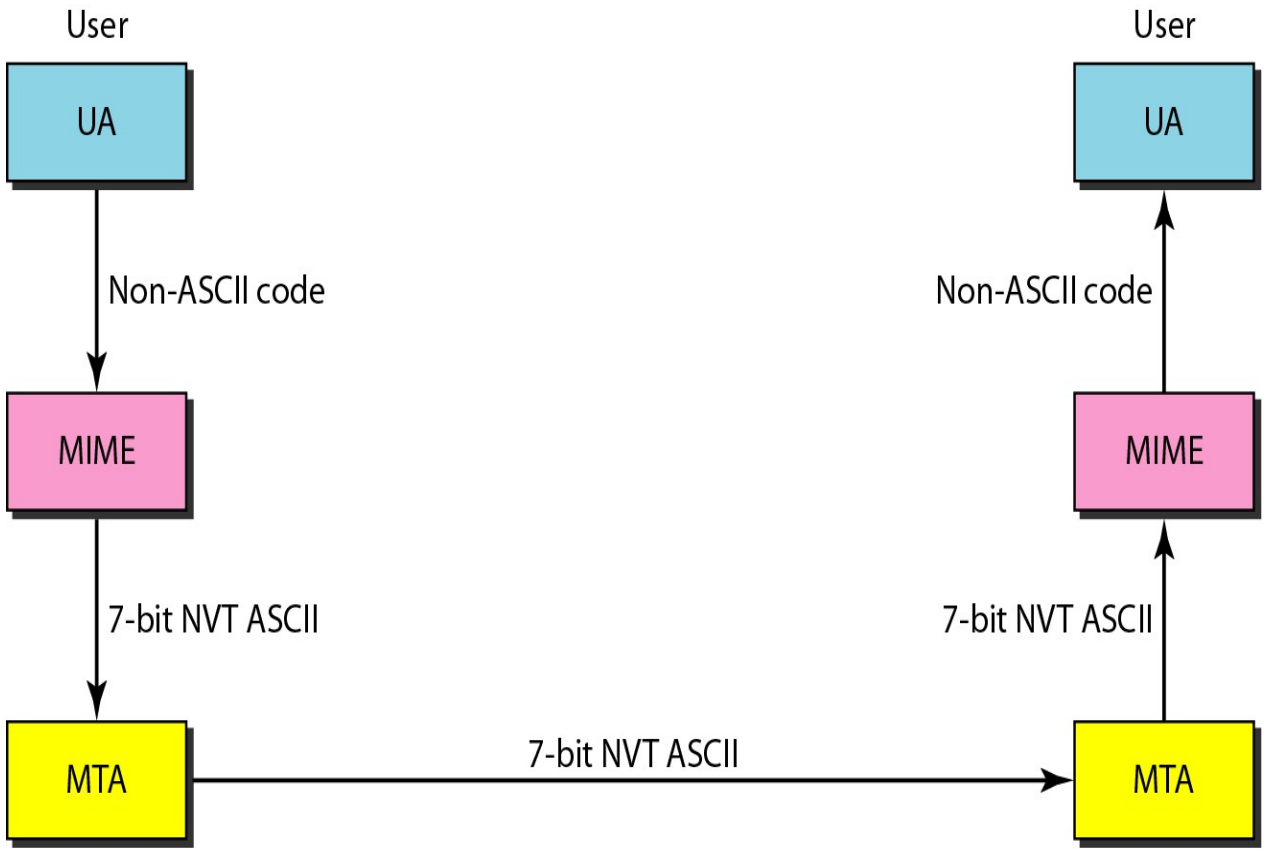


- Local Part: defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.
- Domain Name: The Mail servers or exchangers.

MIME

- Multipurpose Internet Mail Extensions (MIME) -a supplementary protocol.
- To transfer non-ASCII data through e-mail.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet.

MIME


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

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

1. MIME-Version- The current version is 1.1.
2. Content-Type- Type of data used in the body of the message

   Content-Type:<type>/<subtype> parameters

Various types:

Text, Multipart, message, image ,video ,audio, application

3. Content-Transfer-Encoding- To encode message into 0's and 1's.

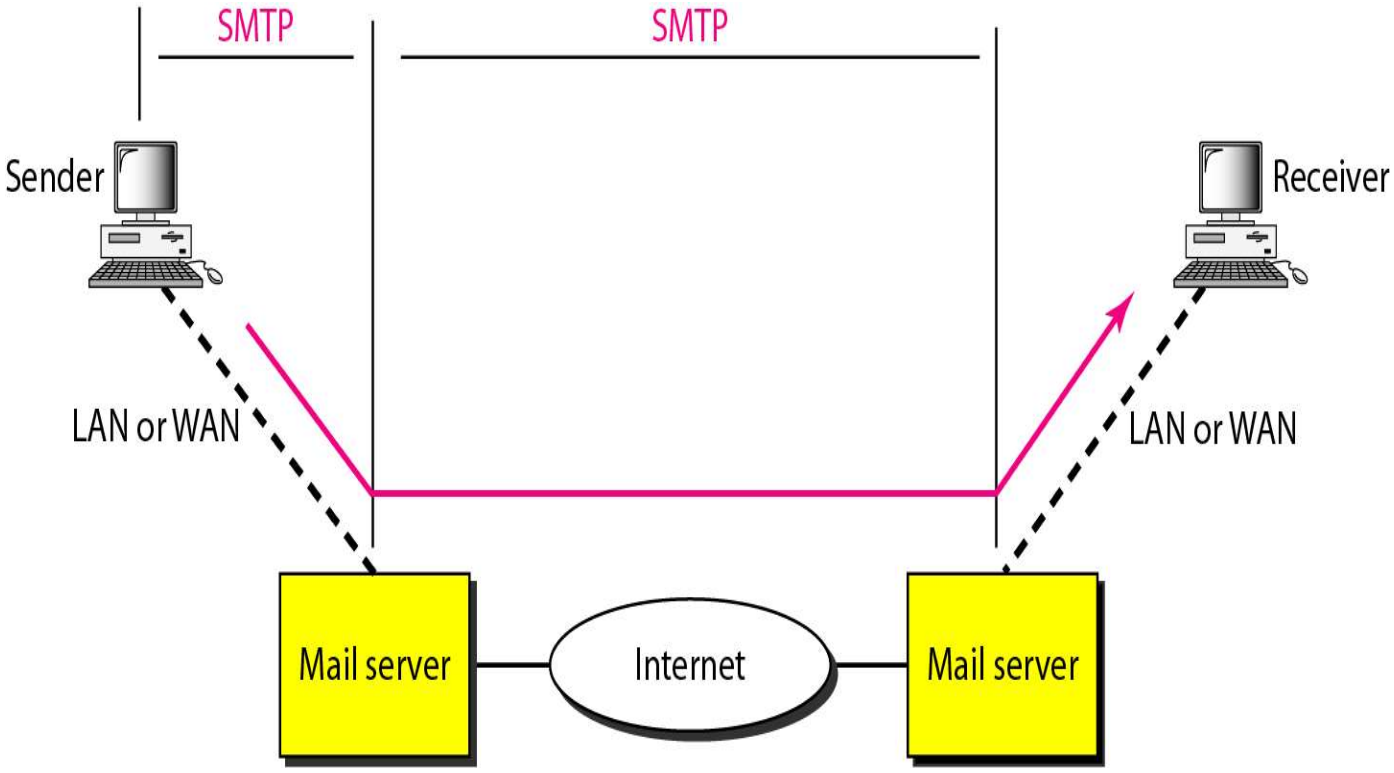
Five types: 7-bit, 8-bit, binary, base-64, quoted printable

4. Content-Id
5. Content-Description: Defines whether the body is image, audio or video

Message Transfer Agent: SMTP

- The actual mail transfer is done through message transfer agents.
- To send mail, a system must have the client MTA, to receive mail, a system must have a server MTA.
- The protocol between client and server is SMTP.
- Two pairs of MTA client/server programs are used in the most common situation.

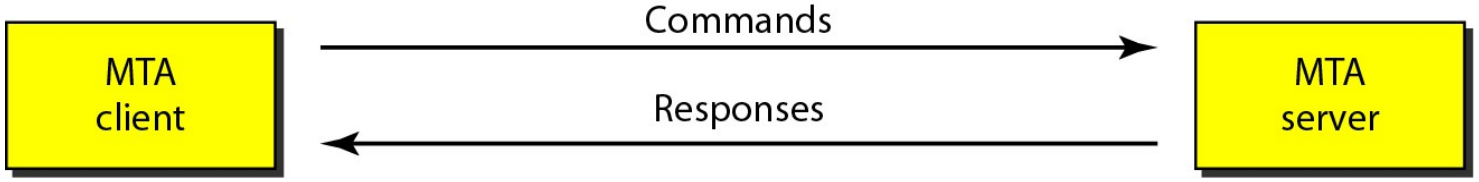
SMTP range


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

- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.
- SMTP simply defines how commands and responses must be sent back and forth.

Commands and Responses

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server



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

Commands:

- Commands are sent from the client to the server.

Format: keyword:argument(s)

- It consists of a keyword followed by zero or more arguments.
- SMTP defines 14 commands.

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

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

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

Mail Transfer Phases

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

$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).

```
================== Connection Establishment ================
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
```

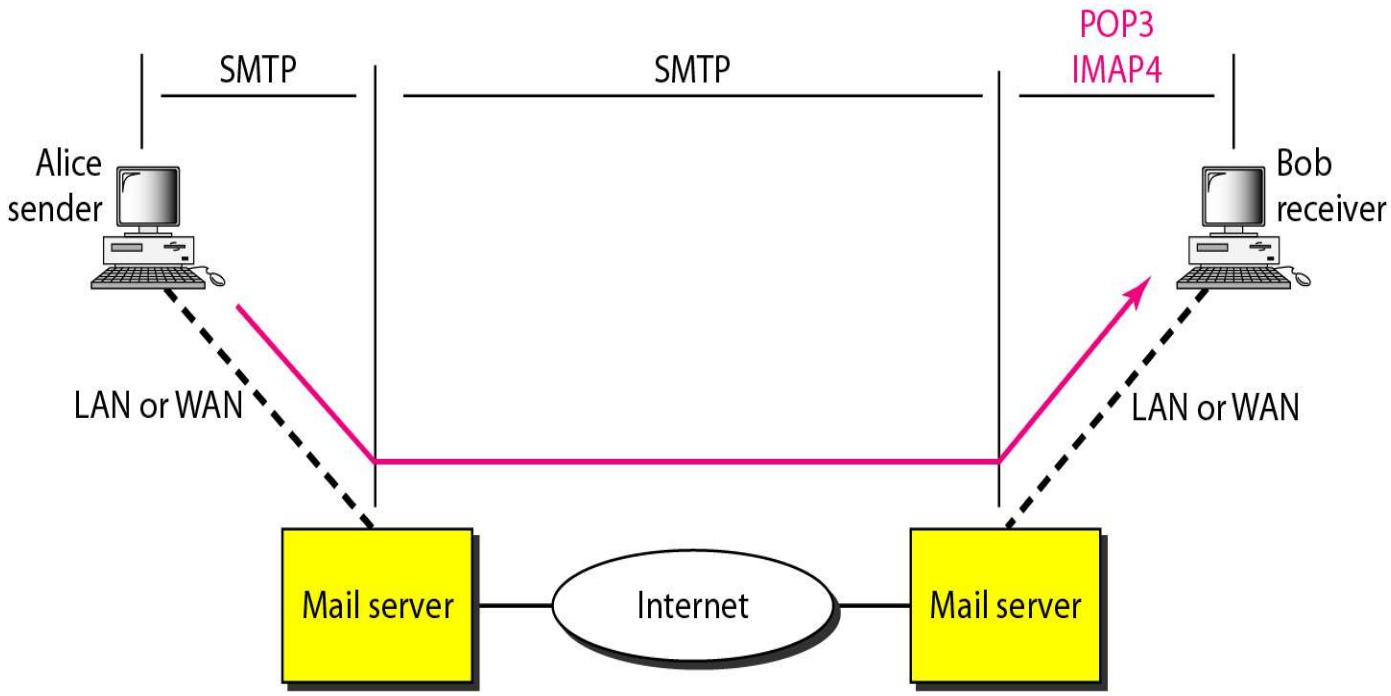
```
==================       Mail Transfer       ================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

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

```
==================== Connection Termination ===============
    250 Message received: adelphia.net@mail.adelphia.net
QUIT
    221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

Message Access Agent: POP and IMAP

- The first and the second stages of mail delivery use SMTP.
- It is not involved in the third stage because SMTP is a *push protocol; it pushes the message from* the client to the server.
- The client needs to pull message from the server, which needs pull protocol.
- This third stage needs message access agent.
- Two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).

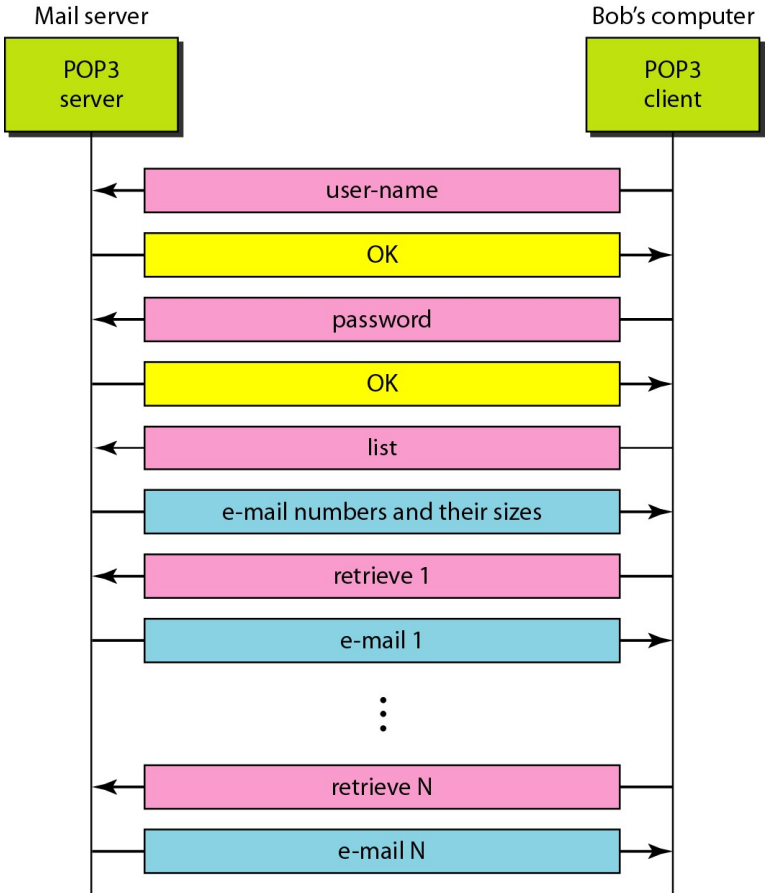
Position of POP3 and IMAP4



POP3

- It is simple and has limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- The protocol starts with the client when the user needs to download e-mail from the mailbox on 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.

Exchange of commands and responses in POP3



- POP3 has two modes:
  - delete mode : mode mail is deleted from the mailbox after each retrieval.
  - keep mode: the mail remains in the mailbox after retrieval.

Deficiencies of POP3:

- User cannot organize the mails on the server.
- Cannot create folders.
- Does not allow user to partially check contents of mail.

IMAP4

- The protocol is more powerful and more complex.

IMAP4 provides the following extra functions:

- A user can check the e-mail header prior to downloading.
- User can search the contents of the e-mail for a specific string of characters.
- A user can partially download e-mail. This is useful for limited bandwidth.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

Web based Mail

- E-mail has become a most common application.
- Websites such as gmail, yahoo, rediff, hotmail etc, provide this services.
- The user using HTTP on the browser logins to the server.
- The transfer of the message from the sending mail server to the receiving mail server is through SMTP.
- In the last phase, instead of POP3 or IMAP4, HTTP is normally used.
- The receiver authenticates himself and uses HTTP to retrieve the message.

Remote Logging - TELNET

- A **general client/server** program that lets a user access any application program on a remote computer.
- Once a user logs in he can use the services available on the remote computer and transfer the results back to the local computer.
- TELNET is a general purpose client/server application protocol

TErminal NETwork.

- It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO).

- TELNET enables the establishment of a connection to a remote system and the local terminal appears to be a terminal at the remote system.

Time Sharing Environment

- TELNET was designed to operate in time sharing environment.
- In such environment, a single host(server) supports multiple users.
- The interaction between a user and the computer occurs through a **terminal**, which is usually a combination of keyboard, monitor, and mouse.
- In such environment, users are part of the system with some right to access resources.

Local login:

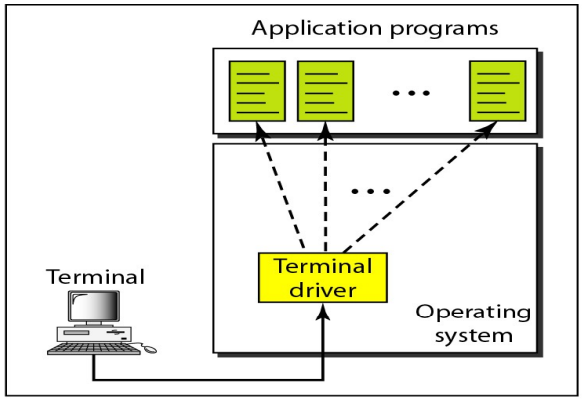
- A terminal driver is responsible for accepting keystrokes from the user and passing them to the OS.
- The OS interprets the combination of characters and invokes the desired application program or utility.

Remote Login:

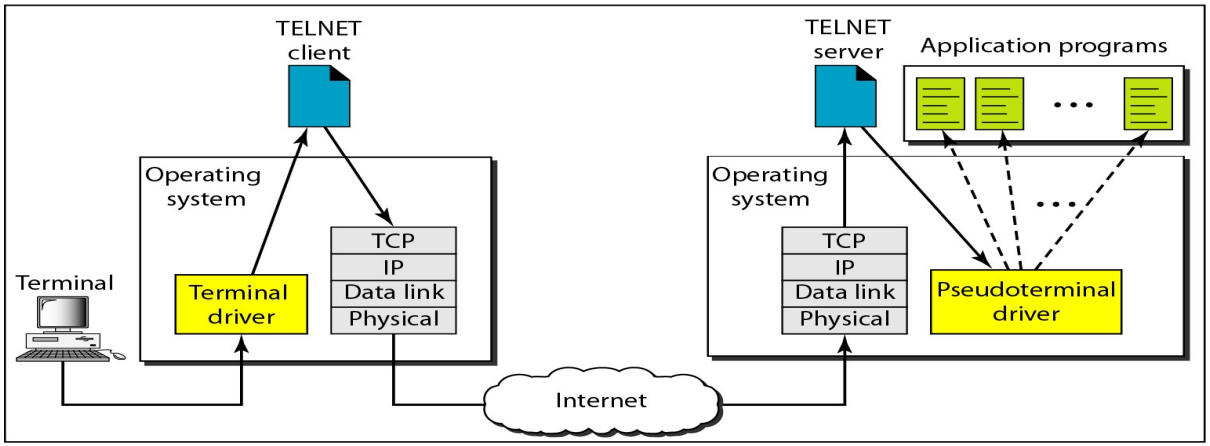
- To access an application program or utility located on a remote machine.
- The user sends the keystrokes to the terminal driver, and the local operating system accepts the characters but does not interpret them.

- The characters are then sent to TELNET Client.
- The TELNET Client 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- TELNET Server
- The TELNET server changes the characters to the corresponding characters understandable by the remote computer.

Local and Remote Log-in



a. Local log-in



b. Remote log-in

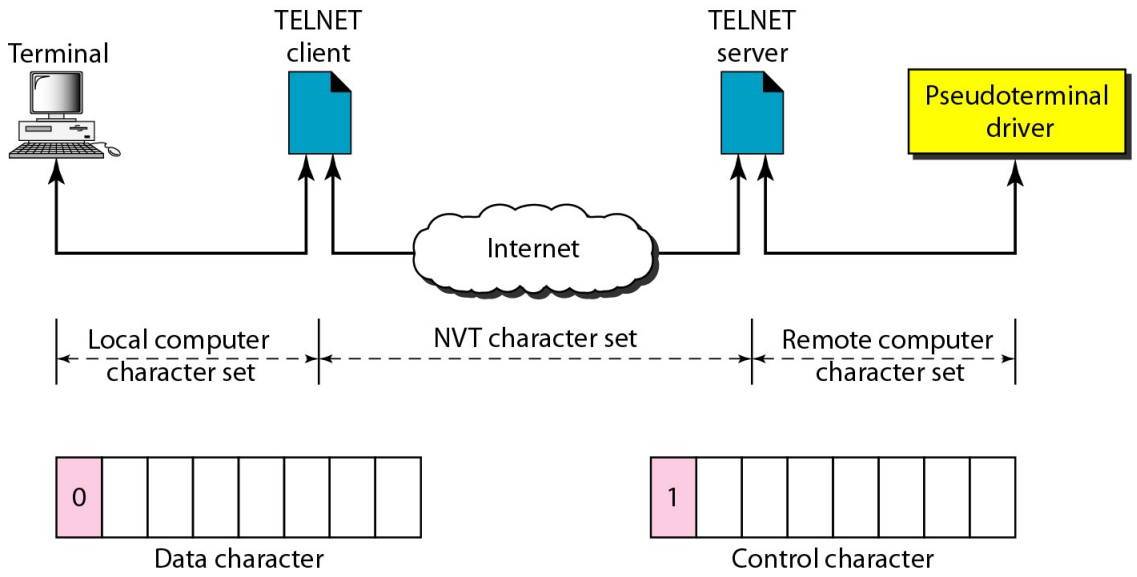
- Since the OS is designed to receive characters from a terminal driver, the TELNET Server passes the characters to **pseudoterminal** driver that pretends like a terminal.
- The pseudoterminal sends characters to OS, the OS then passes characters to appropriate application program.

Network Virtual Terminal

- Accessing a remote computer is complex.
- Every computer and its operating system accept a special combination of characters as tokens.

- For example, the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d.
- Thus with heterogeneous systems, to access any remote computer it would be necessary to have specific terminal emulator.
- A universal interface is needed - 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, the highest-order bit is O.
- 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 l.

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

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.
- To distinguish data from control characters, each sequence of control characters is preceded by a special control character called **interpret as control (lAC).**

- For example: a user wants to display file(file1) on remote server she has to type:

  cat file1.

- Suppose she makes a mistake an sends the command as:

  cat filea1

- The user cannot edit locally; the editing is done at the remote server.

  cat file<backspace>1

- This backspace character is translated into two remote characters (lAC EC), which are embedded in the data and sent to the remote server.

Typed at the remote terminal

Options

- TELNET lets the client and server negotiate options before or during the use of the service.
- Options are extra features available to a user with a more sophisticated terminal.

Option Negotiation

- Option negotiation is required between client and server to use any of the options.
- Four control characters are used for this purpose

TELNET 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.

- Options are agreed by a process of negotiation which results in the client and server having a common view of various extra capabilities
- Either end of a telnet dialogue can enable or disable an option either locally or remotely.
- The initiator sends a 3 byte command of the form

IAC,<type of operation>,<option>

- The response is of the same form.

- Operation is one of

Description	Decimal Code	Action
WILL	251	Sender wants to do something.
DO	252	Sender wants the other end to do something.
WONT	253	Sender doesn't want to do something.
DONT	254	Sender doesn't want the other end to do anything.

- Associated with each of the these there are various possible responses

Sender Sent	Receiver Responds	Implication
WILL	DO	The sender would like to use a certain facility if the receiver can handle it. Option is now in effect
WILL	DONT	Receiver says it cannot support the option. Option is not in effect.
DO	WILL	The sender says it can handle traffic from the sender if the sender wishes to use a certain option. Option is now in effect.
DO	WONT	Receiver says it cannot support the option. Option is not in effect.
WONT	DONT	Option disabled. DONT is only valid response.
DONT	WONT	Option disabled. WONT is only valid response.

- For example if the sender wants the other end to suppress go-ahead it would send the byte sequence
  255(IAC),251(WILL),3
- The final byte of the three byte sequence identifies the required action.

- For some of the negotiable options, values need to be communicated, once support of the option has been agreed.
- This is done using sub-option negotiation.
- Values are communicated via an exchange of value query commands and responses in the following form.
  IAC,SB,<option code number>,1,IAC,SE
- Character set for suboption negotiation

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

- For example if the client wishes to identify the terminal type to the server the following exchange might take place
  Client 255(IAC),251(WILL),24
  Server 255(IAC),253(DO),24
  Server
  255(IAC),250(SB),24,1,255(IAC),240(SE)
  Client
  255(IAC),250(SB),24,0,'V','T','2','2','0',255(IAC),240(SE)

Mode of operation

- TELNET implementations operate in one of three modes: default mode, character mode, or line mode.

Default Mode:

- It is used if no other modes are invoked through option negotiation.
- In this mode, the echoing is done by the client.
- The user types a and the client echoes the character on the screen (or printer) but does not send it until a whole line is completed.

Character Mode

- In the character mode, each character typed is sent by the client to the server.
- The server normally echoes the character back to be displayed on the client screen.
- In this mode the echoing of the character can be delayed if the transmission time is long (such as in a satellite connection).
- It also creates overhead (traffic) for the network because three TCP segments must be sent for each character of data.

Line Mode

- This is new mode to compensate for the deficiencies of the default mode and the character mode.
- In this mode, line editing (echoing, character erasing, line erasing, and so on) is done by the client.
- The client then sends the whole line to the server.