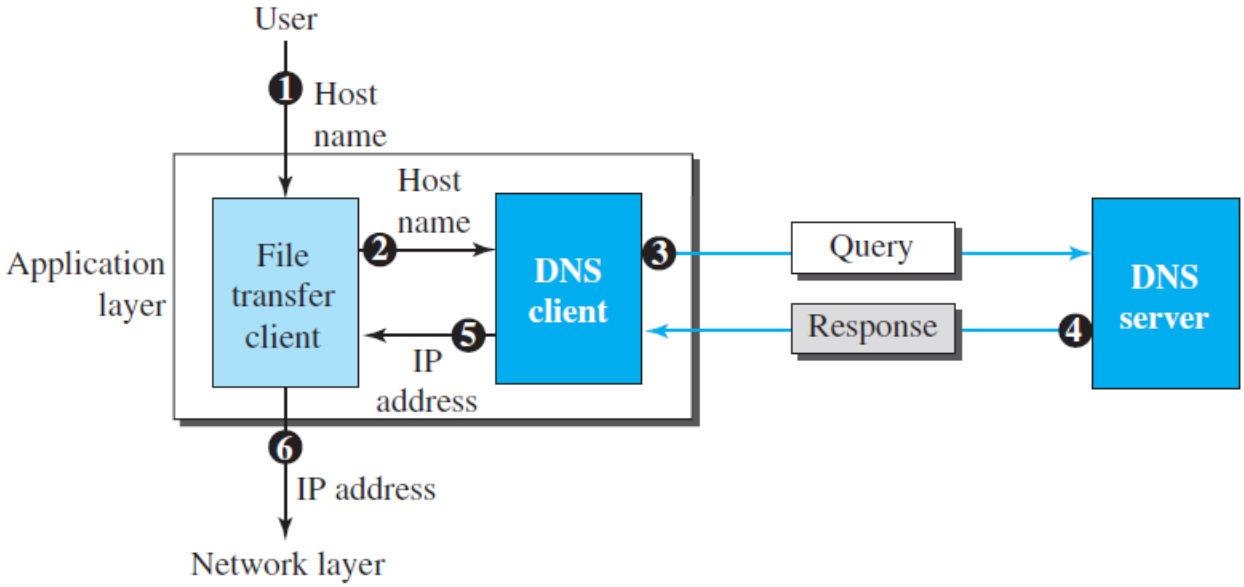
UNIT-V

APPLICATION LAYER

Domain Name System (DNS)

Domain Name System (DNS) is a supporting program that is used by programs such as E- mail. DNS map a name (E mail address or website name) to an IP address or an IP address toa name.



The above figure shows a DNS client/server program can support an E-mail program to find the IP address of an E-mail recipient.

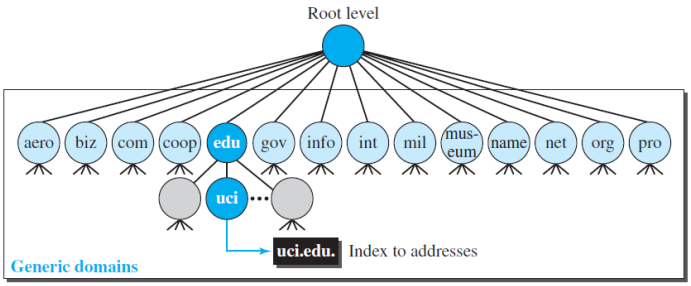
- A user of an E-mail program knows the E-mail address of the recipient but the IP protocolneeds the IP address.
- The DNS client program sends a request to a DNS server to map the E-mail address to the corresponding IP address.
- To identify an entity TCP/IP protocols uses the IP address, which uniquely identifies the connection of a host to the Internet. DNS is designed for the purpose of mapping name to address and address to name.

DNS in the Internet

- DNS is a protocol that can be used in different platforms. In the Internet, the domain
- name space (tree) was originally divided into three different sections: generic domains,
- country domains, and the inverse domains.

Generic Domains

- The **generic domains** define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database.

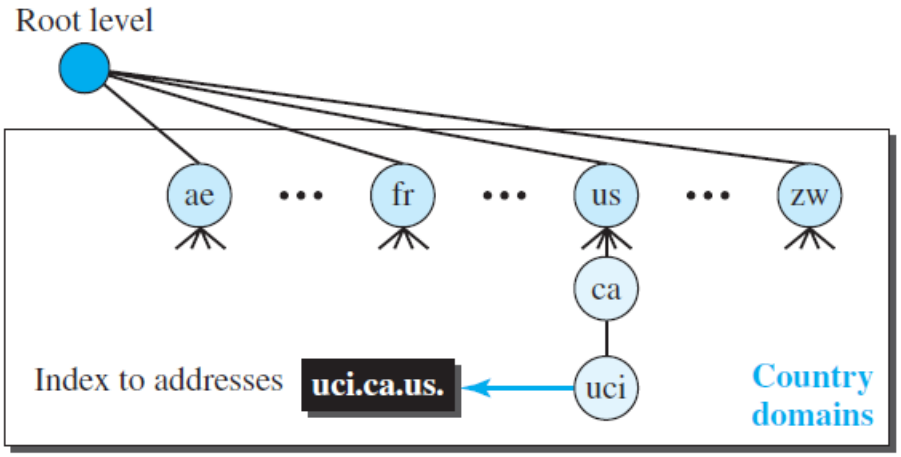


Generic Label Domains:

Label	Description	Label	Description
aero	Airlines and aerospace	**int**	International organizations
biz	Businesses or firms	**mil**	Military groups
com	Commercial organizations	**museum**	Museums
coop	Cooperative organizations	**name**	Personal names (individuals)
edu	Educational institutions	**net**	Network support centers
gov	Government institutions	**org**	Nonprofit organizations
info	Information service providers	**pro**	Professional organizations

Country Domains

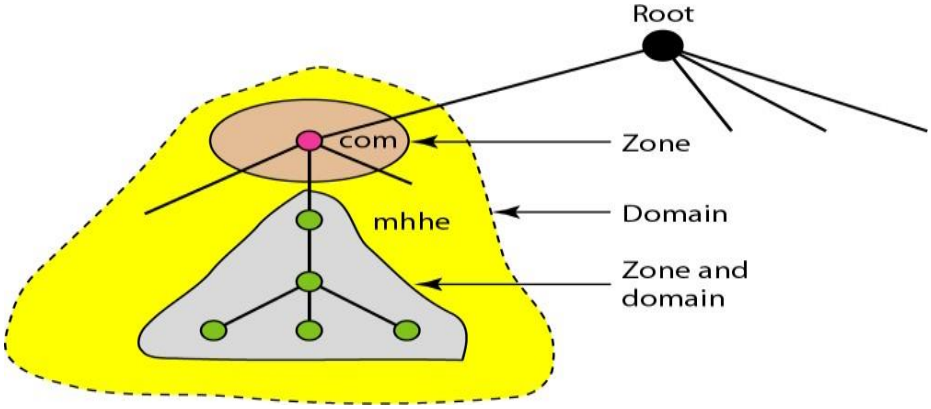
- The **country domains** section uses two-character country abbreviations (e.g., us for United States).
- Second labels can be organizational, or they can be more specific national designations.
- The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).



DISTRIBUTION OF NAME SPACE

- The information contained in the domain name space must be stored and distributed among different computers and in different places.
- Distribution of the information among many computers called DNS servers and space is divided into many domains based on the first level. The root stand alone and create as many domains (subtrees). A domain may itself be divided into sub-domains.
- Each server can be authoritative for either a large or a small domain.

Zone is a contiguous part of the entire tree and it defines what a server is responsible for or server has authority over.



Case 1: When Domain is same as Zone.
If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain.

Case 2: When Domain and Zone are different.
If a server divides its domain into subdomains and delegates part of its authority to other servers, domain and zone refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers.

Root Server: A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.

DNS defines two types of servers: Primary and Secondary servers.

- **Primary Server** is a server that stores a file about the zone for which it is an authority.
- **Secondary Server** is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. These secondary servers are used for crash recovery.
- **Zone transfer:** When the secondary server downloads information from the primary server it is called zone transfer.

NAME ADDRESS RESOLUTION

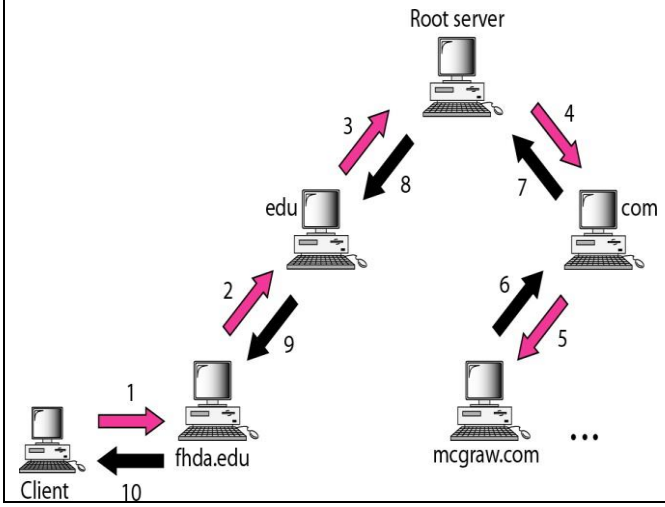
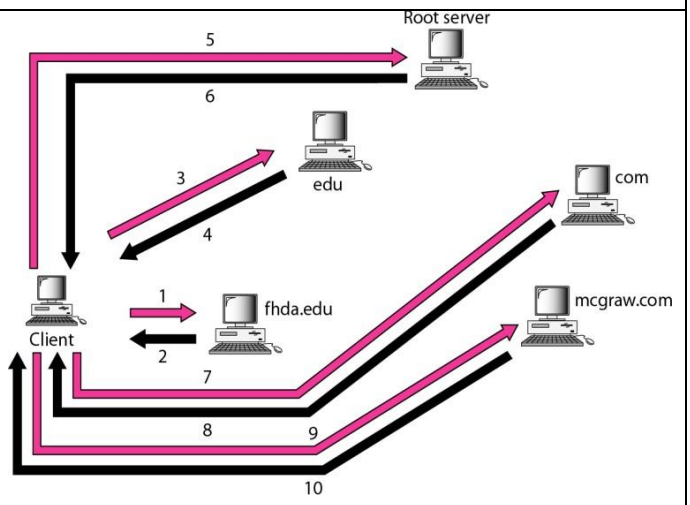
Mapping a name to an address or an address to a name is called Name-Address Resolution. **Resolver:** DNS is designed as a client/server application. A host that needs to map an addressto a name or a name to an address calls a DNS client called a **Resolver**.

Recursive Resolution

- The client (resolver) can ask for a recursive answer from a name server.
- If the server is the authority for the domain name, it checks its database and responds.
- If the server is not the authority, it sends the request to another server (the parent) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server.
- When the query is finally resolved, the response travels back until it finally reaches the requesting client. This process is called **Recursive Resolution**.

Iterative Resolution

- The client repeats the same query to multiple servers. If the server is an authority for the name, it sends the answer.
- If the server is not an authority it returns the IP address of the server to the client, that the server thinks can resolve the query. The client is responsible for repeating the query to this second server. If that server can resolve the problem, it answers the query with the IP address; otherwise, it also returns the IP address of a new server to the client. This process is called Iterative Resolution.

Recursive Resolution	Iterative Resolution
	

Caching

- When a server asks for a mapping from another server and receives the response, it stores thisinformation in its cache memory before sending it to the client.
- If the client asks for the same mapping, it can check its cache memory and returns the result.

Resource Records

- The zone information associated with a server is implemented as a set of *resource records*.
- In other words, a name server stores a database of resource records.

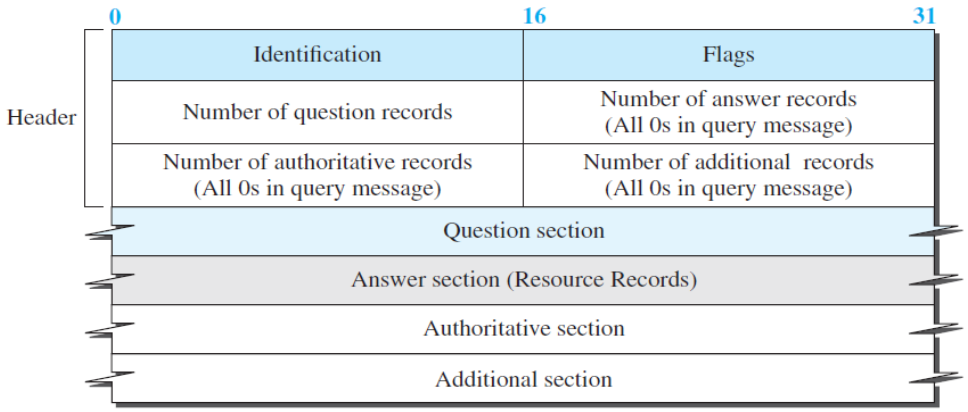
A *resource record* is a 5-tuple structure, as shown below:

- The domain name field is what identifies the resource record.
- The value defines the information kept about the domain name.
- The TTL defines the number of seconds for which the information is valid.
- The class defines the type of network;
- we are only interested in the class IN (Internet).
- The type defines how the value should be interpreted.

Type	*Interpretation of value*
A	A 32-bit IPv4 address (see Chapter 18)
NS	Identifies the authoritative servers for a zone
CNAME	Defines an alias for the official name of a host
SOA	Marks the beginning of a zone
MX	Redirects mail to a mail server
AAAA	An IPv6 address (see Chapter 22)

DNS Messages

To retrieve information about hosts, DNS uses two types of messages: *query* and *response*.



Note:
The query message contains only the question section.
The response message includes the question section,
the answer section, and possibly two other sections.

- The identification field is used by the client to match the response with the query.
- The flag field defines whether the message is a query or response.
- It also includes status of error.
- The next four fields in the header define the number of each record type in the message.
- The question section consists of one or more question records.
- It is present in both query and response messages.
- The answer section consists of one or more resource records.
- It is present only in response messages.
- The authoritative section gives information (domain name) about one or more authoritative servers for the query.
- The additional information section provides additional information that may help the resolver.

Dynamic Domain Name System (DDNS)

In DNS, when there is a change, such as adding a new host, removing a host, or changing anIP address, these change must be made to the DNS master file dynamically (i.e. without manual intervention).

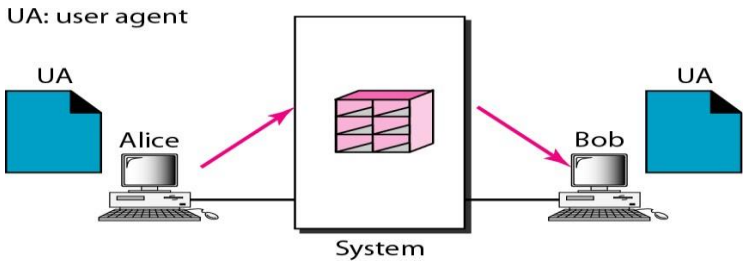
- In DDNS, when a binding between a name and an address is determined the informationis sent usually by DHCP to a primary DNS server.
- The primary server updates the zone. The secondary servers are notified either actively or passively.
- In active notification, the primary server sends a message to the secondary servers about the change in the zone, whereas in passive notification the secondary servers periodically check for any changes, then the secondary server requests information about the entire zone (zone transfer).

ELECTRONIC MAIL (E-Mail)

Electronic mail (E-mail) is one of the most popular Internet services. E-mail allows amessage to include text, audio, and video. There are Four Scenarios of E-mail:

First Scenario

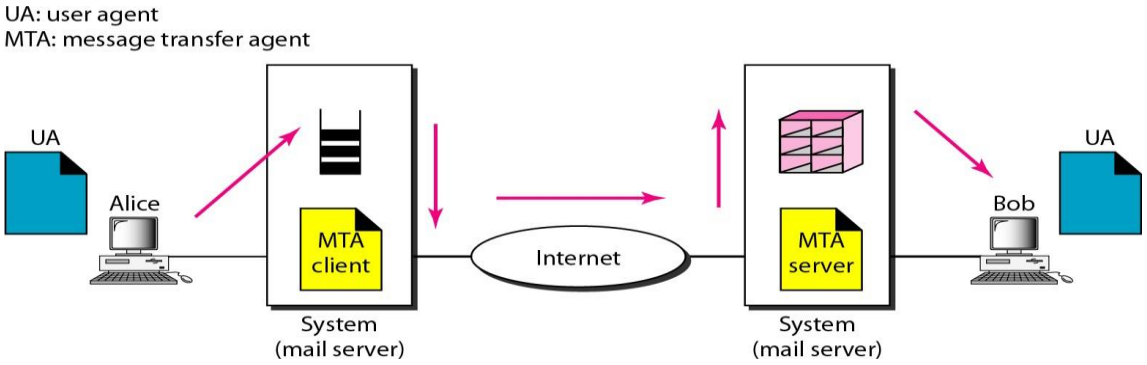
The sender and the receiver of the E-mail are user application programs on the same system. They are directly connected to a shared system.



- When a user Alice needs to send a message to Bob, Alice runs a User Agent (UA) 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 using a User Agent.

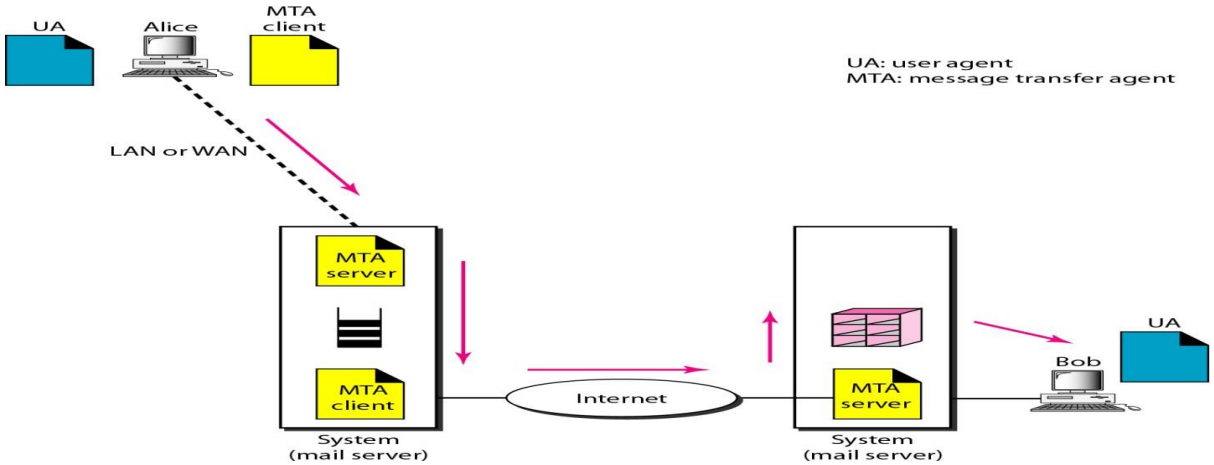
Second Scenario

- In the second scenario, the sender and the receiver of the E-mail are user application programs on two different systems. The message needs to be sent over the Internet.
- We need User Agents (UAs) and Message Transfer Agents (MTA's).



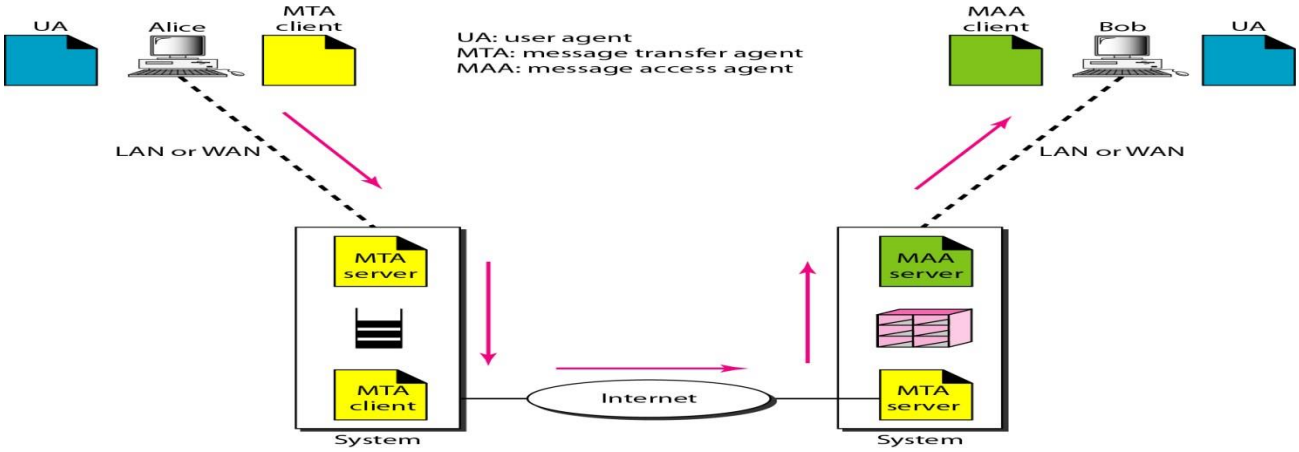
Third Scenario

- In the third scenario, Bob is directly connected to his system (i.e. Mail Server). Alice is separated from her system. Alice is connected to the mail server via WAN or LAN.
- UA of Alice prepares message and sends the message through the LAN or WAN.
- Whenever Alice has a message to send, Alice calls the UA and UA calls the MTA client.
- The MTA client establishes a connection with the MTA server on the system.
- 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. Bob uses his user agent to retrieve the message and reads it. It needs two MTA client and two MTA server programs.



Fourth Scenario

- It is the most common scenario, Alice and Bob both are connected to their mail server by a WAN or a LAN.
- After the message has arrived at Bob's mail server, Bob needs to retrieve it. Now Bob needs another set of client/server agents called Message Access Agents (MAA). Bob usesan MAA client to retrieve his messages.
- The client sends a request to the MAA server and requests the transfer of the messages.



Architecture of E-Mail

There are three major components in the architecture of E-mail:

1. User Agent
2. Message Transfer Agent
3. Message Access Agent

User Agent

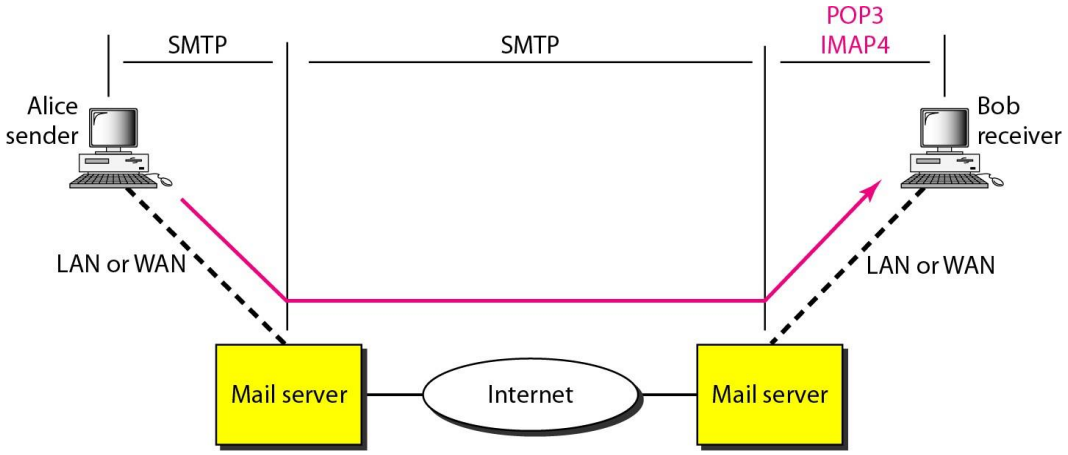
User Agent provides services to the user to make the process of sending and receiving a message easier. Services provided by User agent are:

- **Composing Messages:** A UA helps the user to compose the E-mail message to be sent out.
- **Reading Messages:** The user agent reads the incoming messages.
- **Replying to Messages:** A user agent allows the user to reply to the original sender or to reply to all recipients of the message.
- **Forwarding Messages:** It means sending a message to a third party.
- **Handling Mailboxes:** A user agent normally creates two mailboxes: **Inbox** and **Outbox**. **Inbox** keeps all the received E-mails until they are deleted by the user. **Outbox** keeps all the sent E-mails until the user deletes them.
- **Sending Mail** A user E-mail has an Envelope and a Message. **Envelope** contains the sender and the receiver addresses. **Message** contains the sender, the receiver, the subject of the message, encoding type. Body of the message contains the actual information.
- **Receiving Mail:** If a user has mail, the User Agent informs the user with a notice.
- **Addresses** consists of two parts: a local part and a domain name separated by **@** symbol.
- **Mailing List:** Electronic mail allows one name (an alias) to represent several different E-mail addresses is called a mailing list. Every time a message is to be sent, the system checks the recipient's name against the alias database.

Protocols involved in the working of the Electronic Mail

Electronic mail (E-mail) is one of the most popular Internet services. E-mail allows amessage to include text, audio, and video.

- The actual mail transfer is done through MTA protocol called SMTP. SMTP defines the MTA Client is used to send mail and MTA Server is used to receive a mail.
- SMTP is used two times: Between sender and sender mail server, between sender mail server and receiver mail server.
- SMTP uses commands that are used to send mails from the client to the server. Responses are sent from the server to the client.
- Mail transfer done in 3 phases: Connection Establishment, Mail Transfer, Connection Termination.



Message Access Agent: POP3 and IMAP4POP3 (Post office Protocol version 3)

- Mail access starts with the client when the user needs to download E-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can list and retrieve the mail messages one by one.

POP3 has two modes:

- **Delete Mode** The mail is deleted from the mailbox after each retrieval.
- **Keep Mode** The mail remains in the mailbox after retrieval.

Deficiencies of POP3

- POP3 does not allow the user to organize their mail on the server.
- The user cannot have different folders on the server.
- POP3 does not allow user to partially check the contents of the mail before downloading.

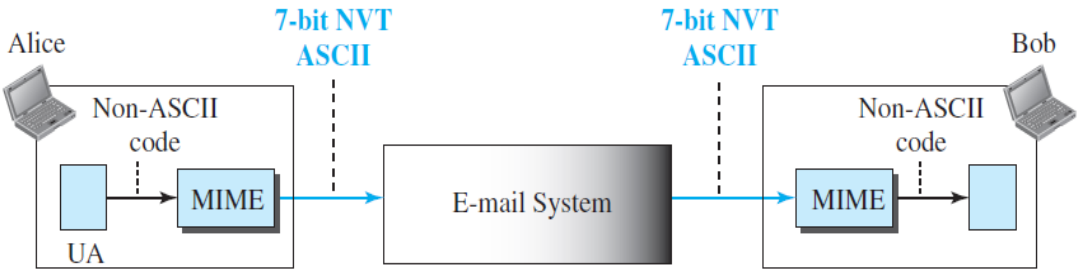
IMAP4 (Internet Mail Access Protocol-version 4)

It is implemented to overcome the deficiencies of POP3. IMAP4 provides the extra functions:

- A user can check the E-mail header prior to downloading.
- A user searches E-mail for a specific string of characters prior to downloading.
- A user can partially download E-mail.
- 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.

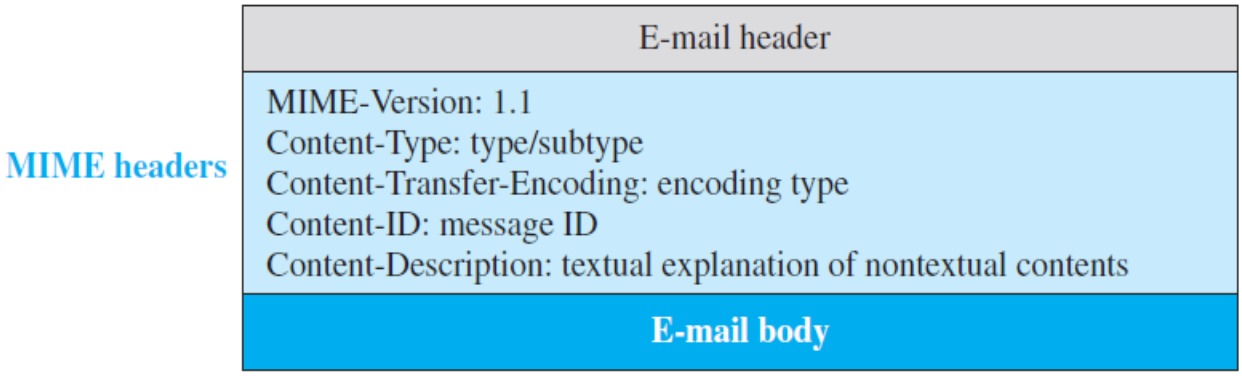
MIME

- MIME is a supplementary protocol that allows non-ASCII data to be sent through E-mail.
- French, German, Hebrew, Russian, Chinese, and Japanese are non-ASCII characters.
- 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.



- The message at the receiving side is transformed back to the original data.

MIME Header



- MIME defines five headers, which can be added to the original e-mail header section to define the transformation parameters.

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.

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
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)

Content-Transfer-Encoding This header defines the method used to encode the messages into 0s and 1s for transport.

Methods for Content-Transfer-Encoding

Type	*Description*
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

The last two encoding methods are interesting. In the Base64 encoding, data, as a string of bits, is first divided into 6-bit chunks.

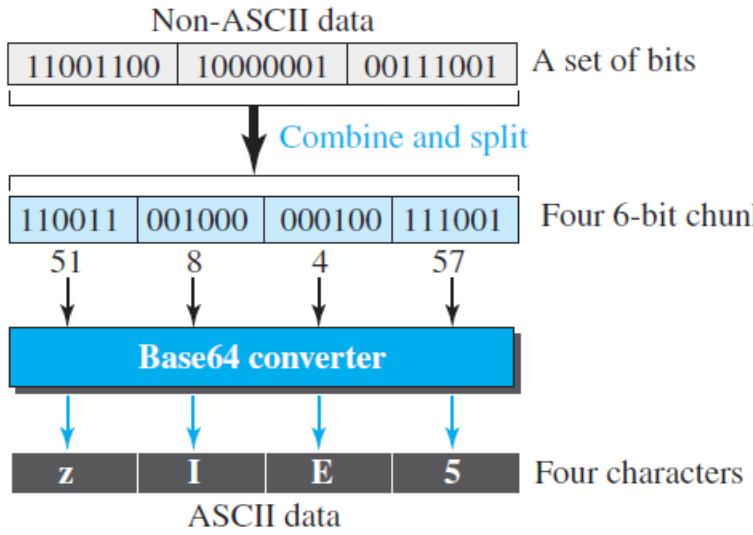


FIG: *Base64 conversion*

Value	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*	*Value*	*Code*
0	**A**	11	**L**	22	**W**	33	**h**	44	**s**	55	**3**
1	**B**	12	**M**	23	**X**	34	**i**	45	**t**	56	**4**
2	**C**	13	**N**	24	**Y**	35	**j**	46	**u**	57	**5**
3	**D**	14	**O**	25	**Z**	36	**k**	47	**v**	58	**6**
4	**E**	15	**P**	26	**a**	37	**l**	48	**w**	59	**7**
5	**F**	16	**Q**	27	**b**	38	**m**	49	**x**	60	**8**
6	**G**	17	**R**	28	**c**	39	**n**	50	**y**	61	**9**
7	**H**	18	**S**	29	**d**	40	**o**	51	**z**	62	+
8	**I**	19	**T**	30	**e**	41	**p**	52	**0**	63	/
9	**J**	20	**U**	31	**f**	42	**q**	53	**1**		
10	**K**	21	**V**	32	**g**	43	**r**	54	**2**		

Table: Base64 Converting table

Each 6-bit section is then converted into an ASCII character
Base64 is a redundant encoding scheme; that is, every six bits become one ASCII character and are sent as eight bits.

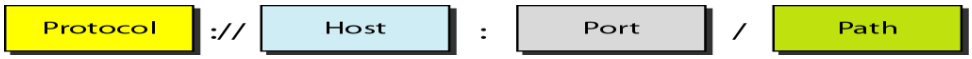
Content-ID This header uniquely identifies the whole message in a multiple message environment.

Content-Description This header defines whether the body is image, audio, or video.

URL (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.

URL defines four things: Protocol, Host computer, Port, and Path.



- **Protocol:** FTP or HTTP.
- **Host:** The host is the computer on which the information is located.
- **Port:** The URL can optionally contain the port number of the server.

Path: It is the pathname of the file where the information is located.

WORLD WIDE WEB

- **World Wide Web** (WWW) is a repository of information linked together from locations allover the world.
- The linking of web pages was achieved using a concept called ***hypertext.***
- Today, the term *hypertext*, coined to mean linked text documents, has been changed to ***hypermedia,*** to show that a web page can be a text document, an image, an audio file, or a video file.
- The purpose of the Web has gone beyond the simple retrieving of linked documents.

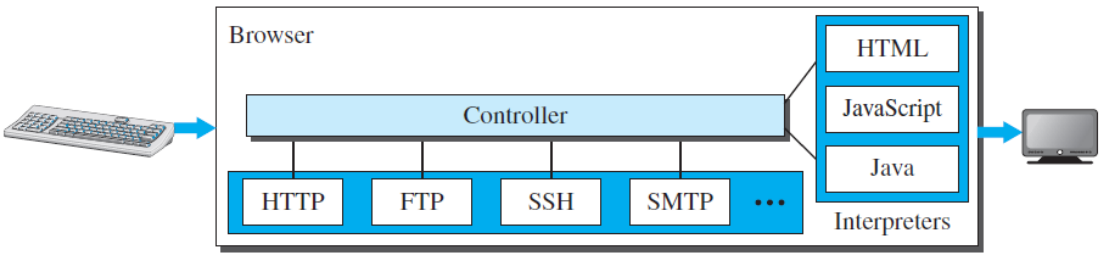
Architecture

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

Architecture of WWW contains four parts: **1. Client 2. Server 3. URL 4. Cookies**

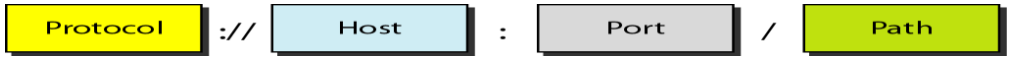
Client: It is a Client is a browser that interprets and displays a Web document. Each browser consists of three parts: **Controller (Keyboard, Mouse)**, **Interpreters (HTML, JAVA, JAVASCRIPT)**, and **Client protocol (HTTP, FTP etc).**

The controller receives input from the keyboard or the mouse and uses the client programs to access the document. Interpreters are used to display the document on the screen.



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.

Uniform Resource Locator (URL): 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. URL defines four things: Protocol, Host computer, Port, and Path.

***Protocol*.** The first identifier is the abbreviation for the client-server program that we need in order to access the web page.
***Host*.** The host identifier can be the IP address of the server or the unique name given to the server.
***Port*.** The port, a 16-bit integer, is normally predefined for the client-server application.
***Path*.** The path identifies the location and the name of the file in the underlying operating system.
To combine these four pieces together, the **uniform resource locator (URL)** has
been designed; it uses three different separators between the four pieces as shown below:

protocol://host/path Used most of the time

protocol://host:port/path Used when port number is needed

Cookies: are used to devise the following functionalities:

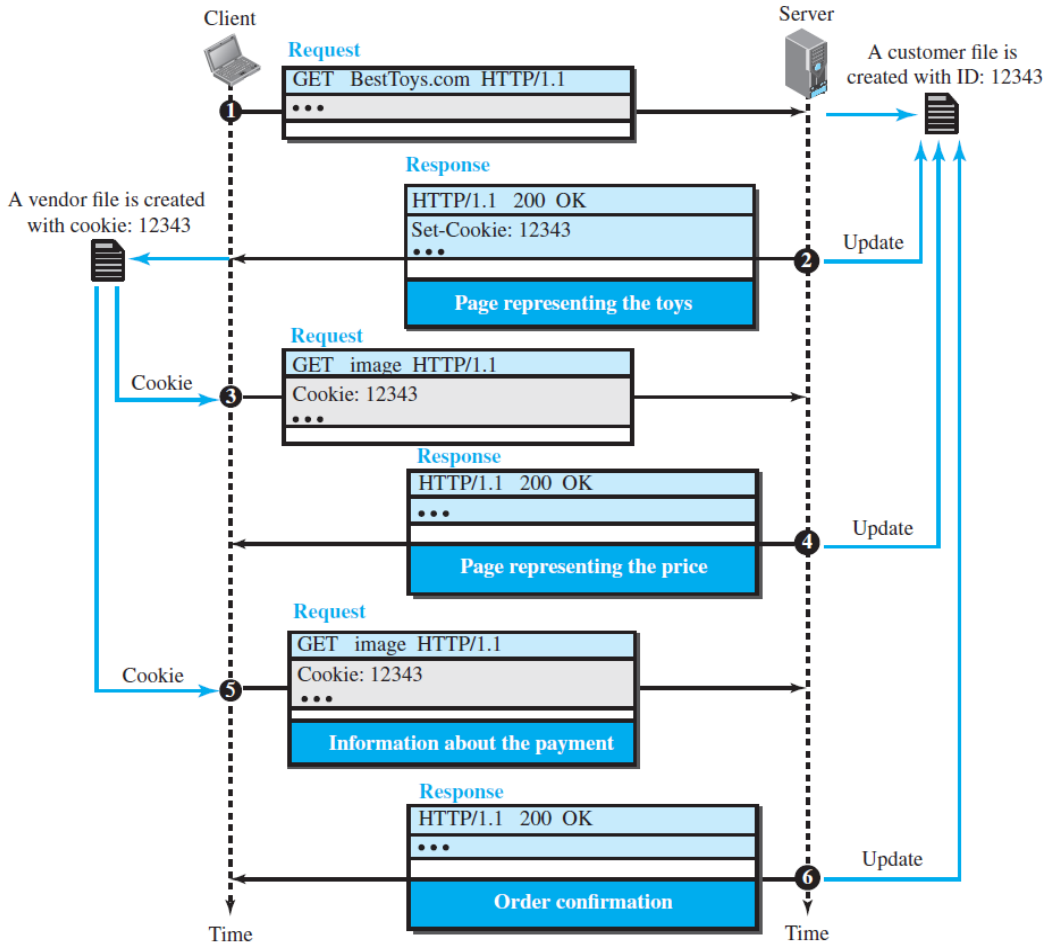
- Some websites need to allow access to registered clients only.
- Websites are being used as electronic stores (such as Flipkart or Amazon) that allow usersto browse through the store, select wanted items, put them in an electronic cart, and pay atthe end with a credit card.
- Some websites are used as portals: the user selects the Web pages he wants to see.
- Some websites are just advertising.

Creating and Storing Cookies

- The creation and storing of cookies depend on the implementation; however, the principle is the same.
- 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 server domain name.

Using Cookies

- An *electronic store* (e-commerce) can use a cookie for its client shoppers.
- The site that restricts access to *registered clients* only sends a cookie to the client when the client registers for the first time.
- A web *portal* uses the cookie in a similar way.
- When a user selects her favourite pages, a cookie is made and sent.
- A cookie is also used by *advertising* agencies.
- An advertising agency can place banner ads on some main website that is often visited by users.



WEB DOCUMENTS

Documents in the WWW can be grouped into three categories:

1. **Static Documents: (HTML)** are fixed-content documents that are created and stored in a server. HTML is a language for creating Web pages.
2. **Dynamic Documents** are created by a Web server whenever a browser requests the document then the Web server runs an application program that creates the dynamic document. The server returns the output of the program as a response to the browser. Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another.
   Example: the retrieval of the time and date from a server is a dynamic document. Dynamic documents are created by using C, C++, Bourne Shell, Korn Shell, C Shell, Tcl, or Perl, PHP, JSP.
3. **Active Documents:** Applications need a program or a script to be run at the client site. These are called active documents. When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client site (browser). Active documents are created by using JAVA (Applets), Javascript,

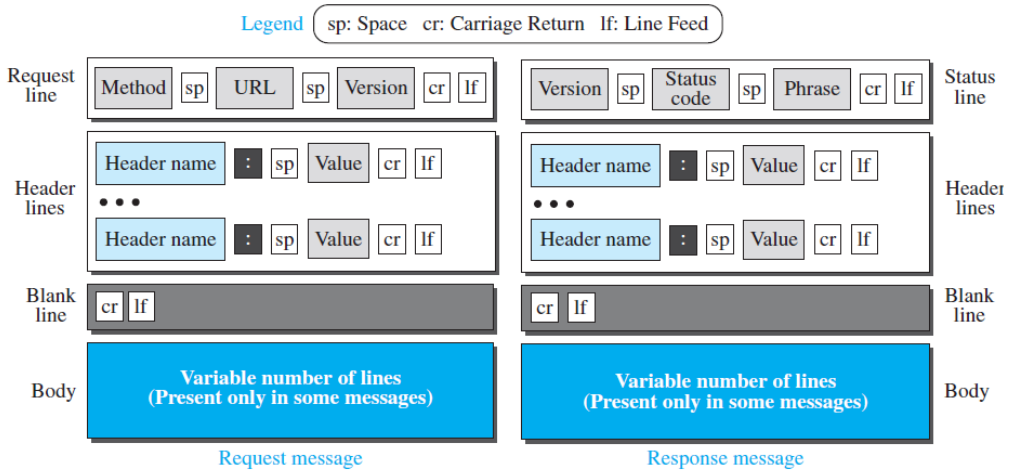
HyperText Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP uses the services of TCP on well-known port 80.

HTTP Transaction

HTTP is a stateless protocol even though it uses TCP services. The client initializes the

transaction by sending a request message consists of a request line, a header, and optional body. The server replies by sending a response consists of a status line, a header, and optionalbody.



Request type is categorized into methods.

Table 26.1 *Methods*

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

URL: By using URL, clients can access the webpage.
Version The most current version of HTTP is 1.1.
Status code is used in the response message. It consists of 3-digits. 100, 200, 300, 400, 500.
Status phrase: It explains the status code in text form, it is used in the response message.

Status Code	Status phrase	Description
400	400- Bad request, 404- Not found,	Error at client side
500	500- Internal server error 503-Service unavailable	Error at server side

Header

The header exchanges additional information between the client and the server. The header can consist of one or more header lines. A Header line can be divided into 4 categories

1. **General header** gives general information about the message such as Date, MIMEversion.
2. **Request header** specifies the client's configuration and the client's preferred document format.

Table 26.2 *Request header names*

Header	*Description*
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server (explained later)
If-Modified-Since	If the file is modified since a specific date

3. **Response header** specifies the server's configuration and special information about the request.

Table 26.3 *Response header names*

Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

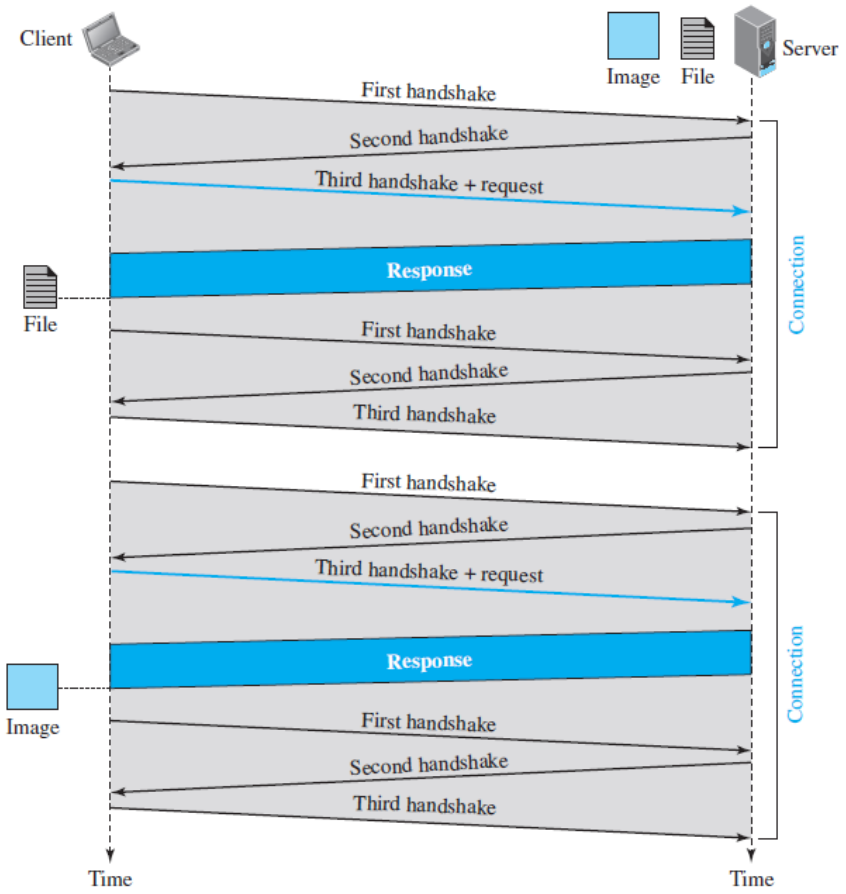
4. **Entity header** gives information about the body of the document.

Body can be present in a request or response message. Body contains the document to be sent or received.

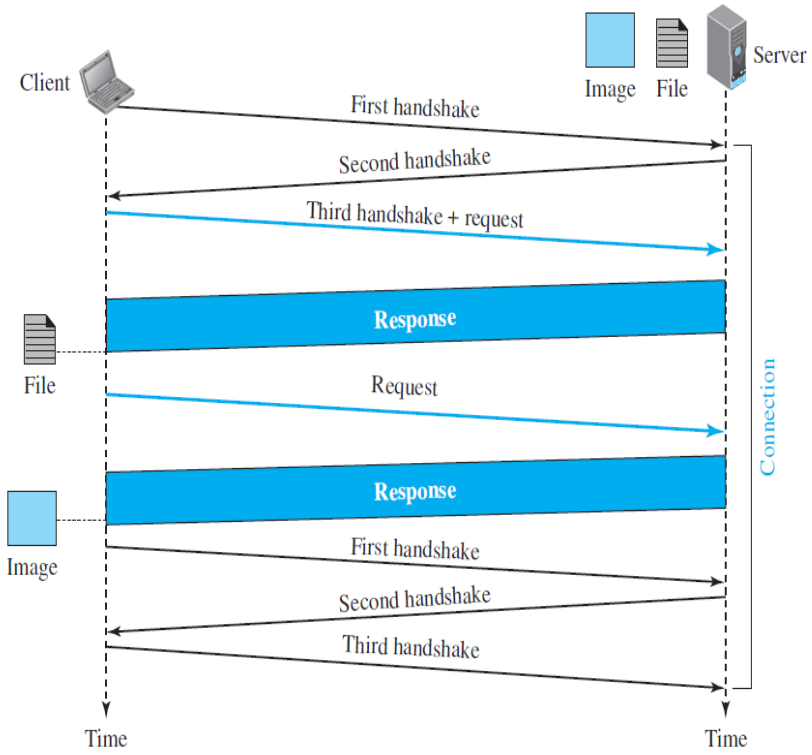
HTTP Connections:

HTTP Connections are categorized in to 2 types: 1. Non persistent 2. Persistent connection.

- **Non-persistent connection:** Versions before 1.1 use Non-persistent method as the default connection. In this connection, one TCP connection is made for eachrequest/response, (i.e.) for N different pictures in different files, the connection must be opened and closed N times. The Non-persistent strategy imposes high overhead on the server because the server needs N different buffers.



- **Persistent Connection** is the default in HTTP version 1.1. In this connection, the server leaves the connection open for more requests after sending a response. Server can close the connection at the request of a client or if a time-out has been reached.



Differences between Persistent and Non-Persistent HTTP Connections

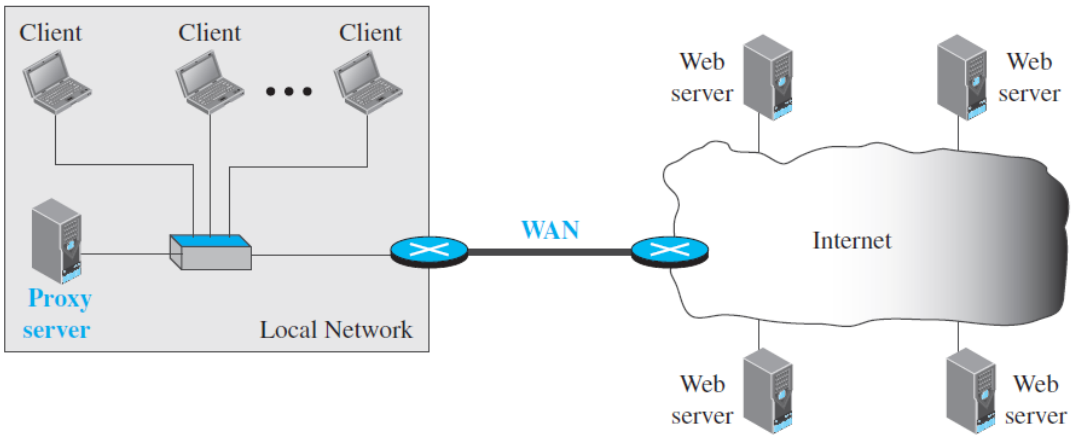
Persistent Connection	Non-Persistent Connection
The Persistent Connection is the second version of the HTTP, and it is also called as HTTP/1.1	The Non-Persistent connection was the first version of HTTP, and it is also called as HTTP/1.0
The Persistent connection will always be in the default mode.	The Non-Persistent connection will always be in the non-default mode.
The Persistent connection uses very less time because all the requests and responses are transferred in a single TCP.	The Non-Persistent connection uses more time when compared to Persistent connection because it uses new TCP for every new request and response.
The Persistent connection requires only one round trip time for all the objects.	The Non-Persistent connection requires two RTT's for every object present in the connection.
The request methods used in the Persistent connection are GET, HEAD, POST, PUT, DELETE, etc.	The request methods used in the non-Persistent connection are HEAD, POST, etc.
For downloading the multiple objects, the Persistent connection only uses a single connection	For downloading the multiple objects, the non-Persistent connection requires multiple connections
The usage of the CPU will be less in the persistent connection because it runs on a single TCP	The usage of the CPU will be more when compared to persistent connection because it runs on the multiple TCP's

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 otherclients.
- 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 thetarget server.



Proxy Server Location

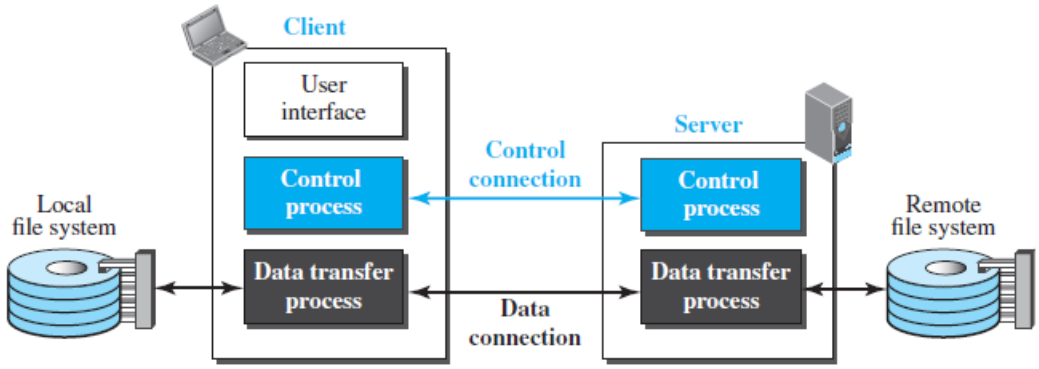
The proxy servers are normally located at the client site. This means that we can have a hierarchy of proxy servers, as shown below:

- A client computer can also be used as a proxy server, in a small capacity, that stores responses to requests often invoked by the client.
- In a company, a proxy server may be installed on the computer LAN to reduce the load going out of and coming into the LAN.
- An ISP with many customers can install a proxy server to reduce the load going out of and coming into the ISP network.

Cache Update

- A proxy server can get the news early in the morning and keep it until the next day.
- Another recommendation is to add some headers to show the last modification time of the information.
- The proxy server can then use the information in this header to guess how long the information would be valid.

FILE TRANSFER PROTOCOL (FTP)



File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another. The client has three components: the user interface, the 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.
- Separation of commands and data transfer makes FTP more efficient.

- The control connection uses very simple rules of communication.
- We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Two Connections

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

Control Connection

- During this control connection, commands are sent from the client to the server and responses are sent from the server to the client.
- Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

Table 26.4 *Some FTP commands*

Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	**F**, **R**, or **P**	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	**A**, **E**, **I**	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User ID	User information
MODE	**S**, **B**, or **C**	Define transmission mode (**S**: stream, **B**: block, or **C**: compressed

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

Table 26.5 *Some responses in FTP*

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

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 the control connection.

The following shows the steps:

- 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.
- Using the PORT command the client sends this port number to the server.
- The server receives the port number and issues an active open using the well known port 20 and the received ephemeral port number.

Security for FTP

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

Advantages of FTP(File Transfer Protocol):

- Speed is one of the advantages of FTP (File Transfer Protocol).
- File sharing also comes in the category of advantages of FTP in this between two machines files can be shared on the network.
- Efficiency is more in FTP.

Disadvantages of FTP (File Transfer Protocol):

- File size limit is the drawback of FTP only 2 GB size files can be transferred.
- Multiple receivers are not supported by the FTP.
- FTP does not encrypt the data this is one of the biggest drawbacks of FTP.
- FTP is unsecured we use login IDs and passwords making it secure but they can be attacked by hackers.

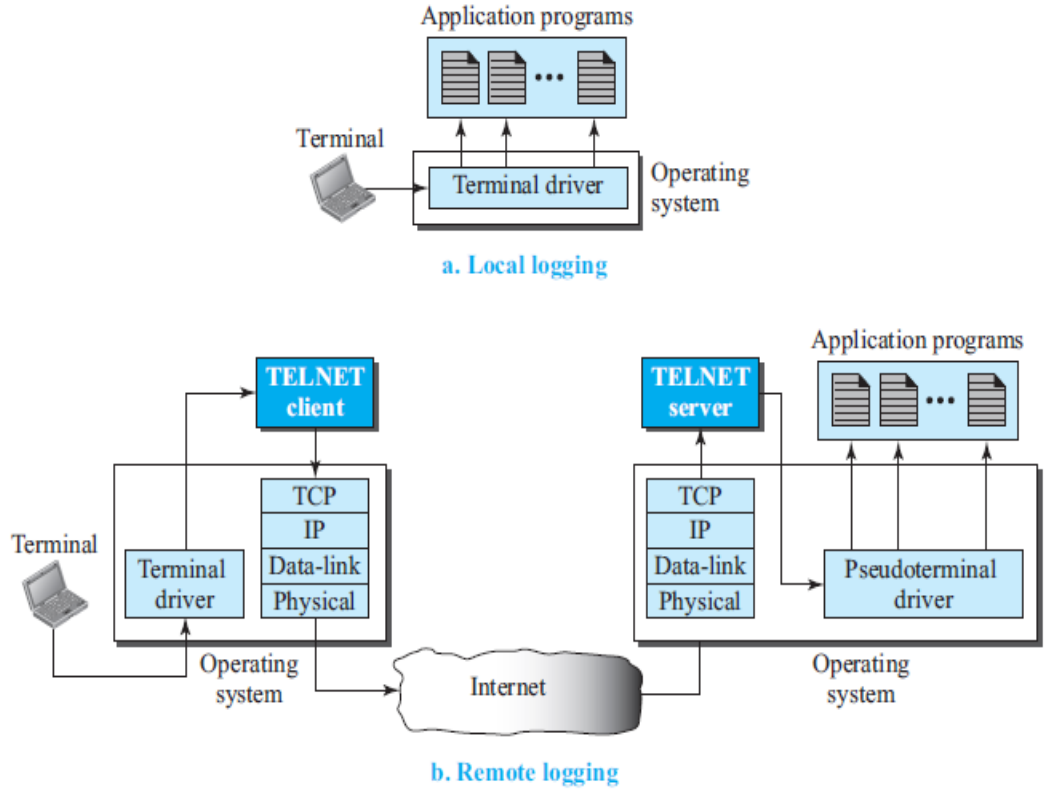
TELNET

- A server program can provide a specific service to its corresponding client program.
- For example, the FTP server is designed to let the FTP client store or retrieve files on the server site.
- One of the original remote logging protocols is **TELNET,** which is an abbreviation for TErminaL NETwork.

Importance of TELNET

- The simple plaintext architecture of TELNET allows us to explain the issues and challenges related to the concept of remote logging, which is also used in SSH when it serves as a remote logging protocol.
- Network administrators often use TELNET for diagnostic and debugging purposes.

Local versus Remote Logging



- When a user logs into a local system, it is called local logging. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system.
- The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility. when a user wants to access an application program or utility located on a remote machine, she performs remote logging.
- 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.