UNIT I

Application Layer

S

er Paradigm : *Communication using TCP and UDP, Peer to Peer Paradigm, Application*
ocols : DNS, FTP, TFTP, HTTP, SMTP, POP, IMAP, MIME, DHCP, TELNET.

S

1.1 Application Layer Paradigm

- The layer where all the applications are found is called Application Layer.
- The application layer enables the user, whether human or software, to access the network.
- It provides user interfaces and support for services such as electronic mail, file access and transfer, access to system resources, surfing the World Wide Web, and network management.
- Application layer needs support protocols, to allow the applications to function. Accordingly, we will look at an important one of these before starting with the applications themselves.
- The application layer provides services to the user. Communication is provided using a logical connection, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages. Fig. 1.1.1 shows the idea behind this logical connection.
- The Fig. 1.1.1 shows the logical connection is establish between two application layers.

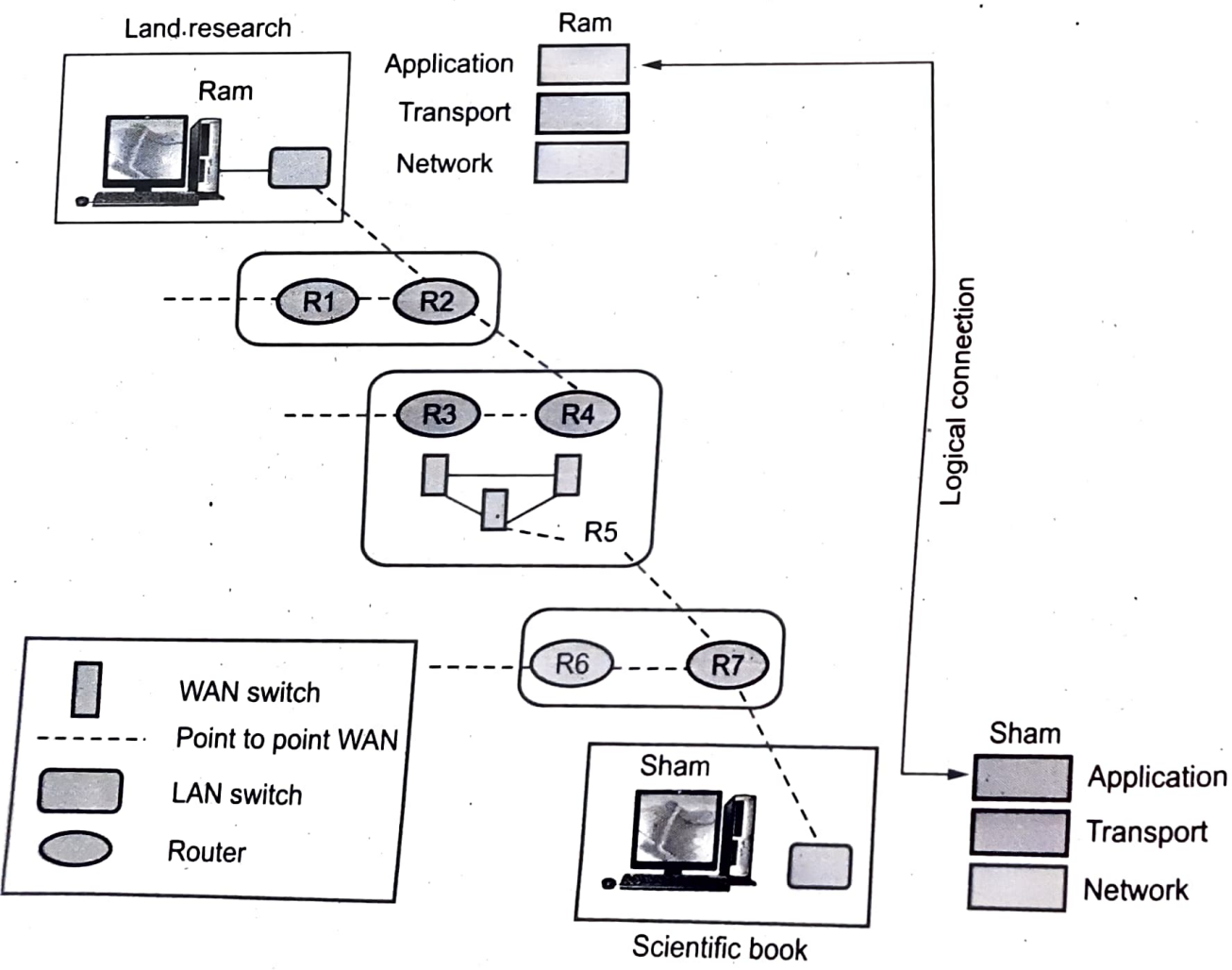

Fig. 1.1.1 Logical connection at the application layer

A scientist working in a research company, Land Research, needs to order a book related to his research from online bookseller, named Scientific Books.

Logical connection takes place between the application layer of a computer at Land research and the application layer of a server at Scientific Books.

We call the first host Ram and second host Sham. The communication at the application layer is logical, not physical. Ram and Sham assume that there is a two way logical channel between them through which they can send and receive messages.

The actual communication, however, takes place through several devices (Ram-R2-R4-R5-R7 and Sham) and several physical channels, as shown in figure.

The service provider is an application program, called the server process.

Server process runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service.

The server process must be running all the time.

The client process starts when the client needs to receive service.

Several traditional services are still using this paradigm, e.g., WWW, HTTP, FTP, SSH, E-mail, and so on.

blems

- The server should be a powerful computer.
- There should be a service provider willing to accept the cost and create a powerful server for a specific service.

2 Client - Server Programming

- In the client/server model, the device requesting the information is called a client and the device responding to the request is called a server.
- Client and server processes are considered to be in the application layer. Data transfer from a client to a server is referred to as an upload and data from a server to a client as a download.
- In a client/server network, the server runs a service or process, sometimes called a server **daemon.** Like most services, daemons typically run in the background and are not under an end user's direct control.
- When a daemon "hears" a request from a client, it exchanges appropriate messages with the client, as required by its protocol and proceeds to send the requested data to the client in the proper format.

1.2.1 Application Programming Inte

- Application programming is a set of instructions t layers (in OS).
- Application programming instructs to open a connection, send and receive data close the connection.
- Application programming is set of instructions of this kind is API.

Interface between a process and network

- Several APIs have been designed for communication.
- Three most common APIs : - 1. Socket interface 2. Transport Layer Interface (TLI 3. STREAM
- A process sends messages into and receives messages from, the network through a software interface called a socket.
- Fig. 1.2.1 shows application process, sockets and underlying transport protocol.

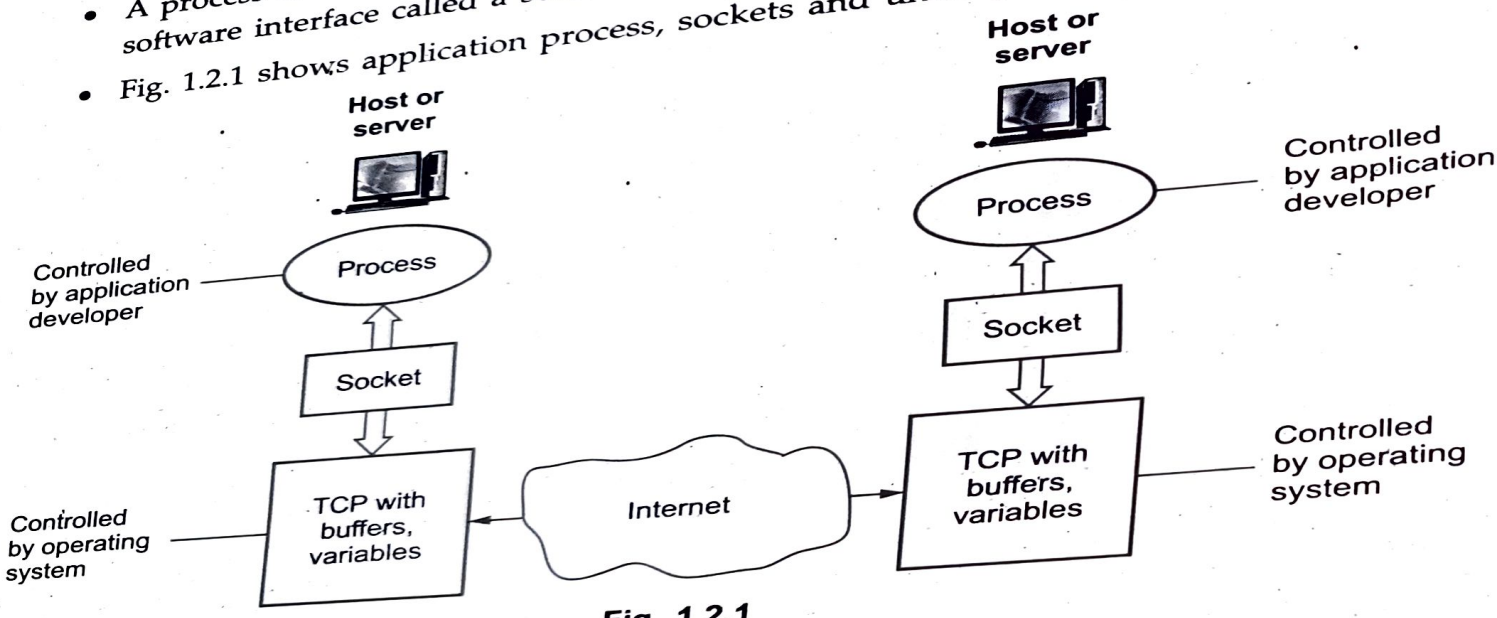

Fig. 1.2.1

Socket interface

- Socket interface started in the early 1980s at UC Berkeley as part of a UNIX environment.
- The socket interface is a set of instructions that provide communication between the application layer and the OS.
- The idea of socket allows us to use the set of all instructions already designed in a programming language for other sources and sinks.
- Socket is not a physical entity like files, keyboard, etc.; it is an abstraction.

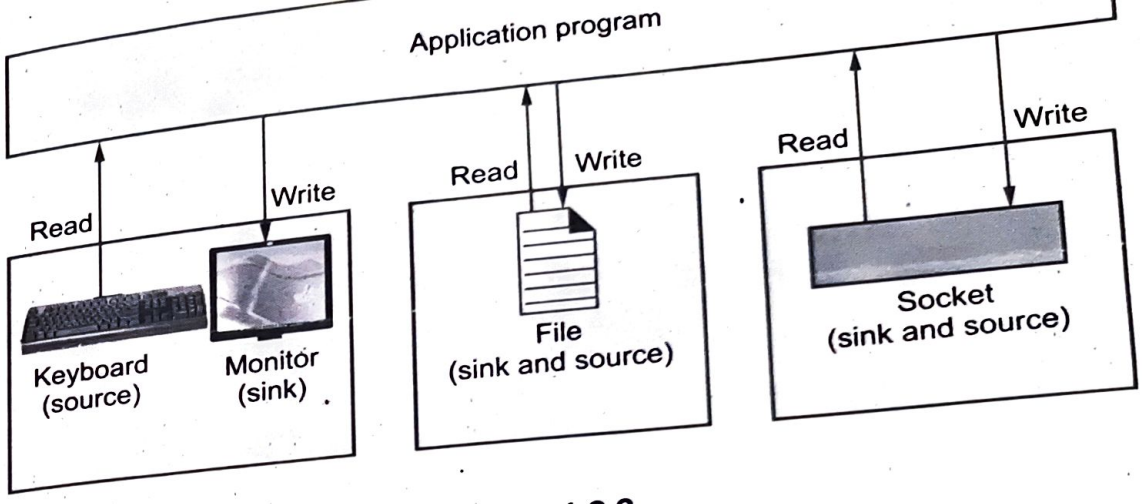

Fig. 1.2.2

- Communication between a client process and a server process is nothing but communication between two sockets.
- Fig. 1.2.3 shows use of sockets in process - to - process communication.

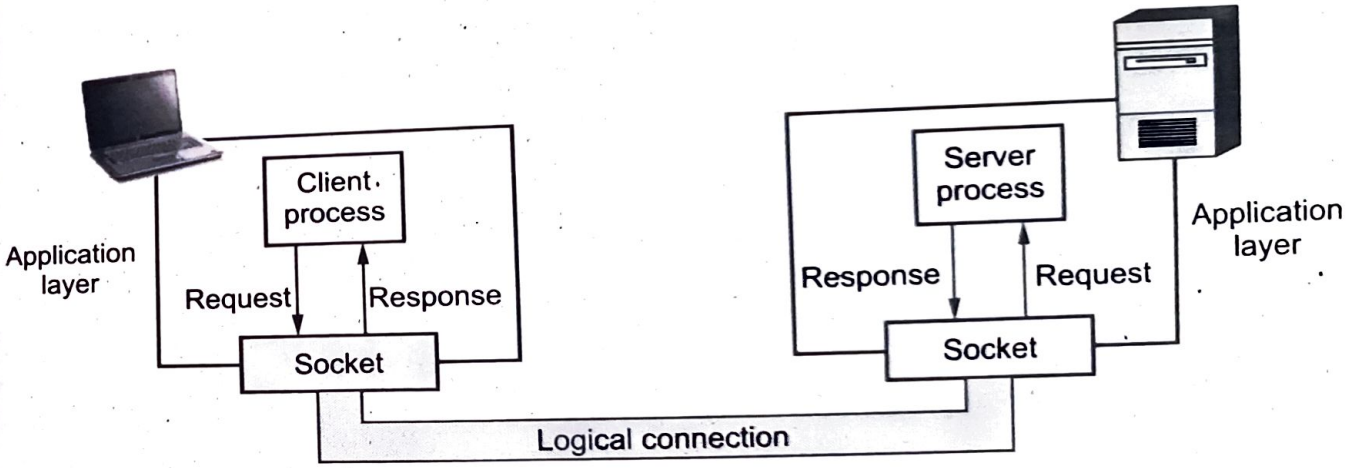

Fig. 1.2.3

- The process - to - process communication need a pair of socket addresses for communication : - A local socket address and a remote socket address.

- A socket address should
  - First define the computer on which a client or a server is running. A computer in the internet is uniquely defined by its IP address
  - Then, we need another identifier to define the specific client or server involved in the communication. An application program can be defined by a port number. Popular applications have been assigned specific port numbers. Few port numbers : Web server = 80, Mail Server = 25. So, socket address = {IP address, port number}

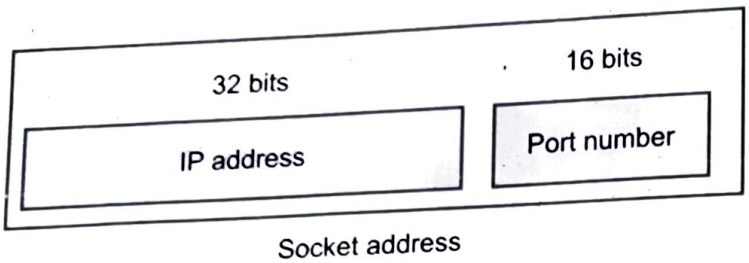

Fig. 1.2.4

1.2.2 Using Services of the Transport Layer

- The choice of the transport layer protocol seriously affects the capability of the application processes.
- Broadly classify the possible transport layer services along four dimensions :
  1. Reliable data transfer
  2. Throughput
  3. Timing
  4. Security
- **Use UDP :**
  a) If it is sending small messages
  b) If the simplicity and speed is more important for the application than reliability
  c) For lightweight transport protocol, providing minimal services
- **Use TCP :**
  a) If it needs to send long messages and require reliability
  b) For providing security it use SSL (Secure Socket Layer)

1.3 Domain Name System (DNS)

- Goal : Assign meaningful high-level names to a large set of machines and handle the mapping of those names to a machine's **IP** address.
- The DNS is a distributed database that resides on multiple machines on the internet and used to convert between names and address and to provide e-mail routing information.
- DNS provides the protocol that allows the client and servers to communicate with each other.
- Domain names are case insensitive so **com** and **COM** mean the same thing.
- The **DNS** protocol runs over **UDP** and uses port 53.
- The **DNS** is specified in **RFC 1034** and **RFC 1035.**
- The **DNS** protocol is the application layer protocol.
- A full domain name is a sequence of labels separated by dots (.).
- The **DNS** name space is hierarchical and it is similar to the unix file system.
- Originally, the internet was small and mapping between names and addresses was accomplished using a centrally-maintained file called *hosts.txt*. To add a name or change an address required contacting the central administrator, updating the table, and distributing it to all the other sites. This solution worked at first because most sites had only a few machines, and the table didn't require frequent changes.
- **The centrally-maintained table suffered from several drawbacks :**
  1. The name space was *flat*, and no two machines could use the same machine name.
  2. As the internet grew, changes to the database took days to weeks to take effect.
  3. The central site became congested with the increase in the number of sites retrieving copies of the current table.
  4. The internet grew at an astonishing rate.
- The Domain Name System (DNS) is a hierarchical, distributed naming system designed to cope with the problem of explosive growth :
  1. It is *hierarchical* because the name space is partitioned into *subdomains*.
  2. It is distributed because management of the name space is delegated to local sites. Local sites have complete control (and responsibility) for their part of the name space. DNS queries are handled by servers called *name servers*.
  3. It does more than just map machine names to internet addresses. For example, it allows a site to associate multiple machines with a single, mailbox name.

- In the DNS, the name space is structured as a tree, with *domain names* referring to nodes in the tree. The tree has a *root*, and a *fully-qualified* domain name is identified by the *components* of the path from the domain name to the root.

Services provided by DNS :

- **Host aliasing :** A host with complicated hostname can have one or more alias names. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.
- **Mail server aliasing :** DNS can be invoked by a mail application to obtain the hostname for a supplied alias hostname as well as the IP address of the host.
- **Load distribution :** DNS is also used to perform load distribution among replicated servers.

1.3.1 Components of DNS

- DNS includes following components
  1. Domain 2. Domain name 3. Name server
  4. Name resolver 5. Name cache 6. Zone

1) For example, vtubooks.com is the site for technical publications. Here com' is the domain.

2) Domain name is defined by the DNS as being the sequence of names and domain. For example, vtubooks.com could be domain name.

3) In name server, software (program) that maps names to addresses. It does this by mapping domain names to IP addresses.

4) Name resolver is a software that functions as a client interacting with a name server.

5) Name cache is the storage used by the name resolver to store information frequently used.

6) Zone is a contiguous part of a domain.

1.3.2 DNS in the Internet

- DNS is divided into three different sections in the internet i.e. Generic domain Country domain and Inverse domain.
- Fig. 1.3.1 shows the DNS in the internet.

Generic Domains

- Each node in the tree defines a domain, which is an index to the domain name space database.

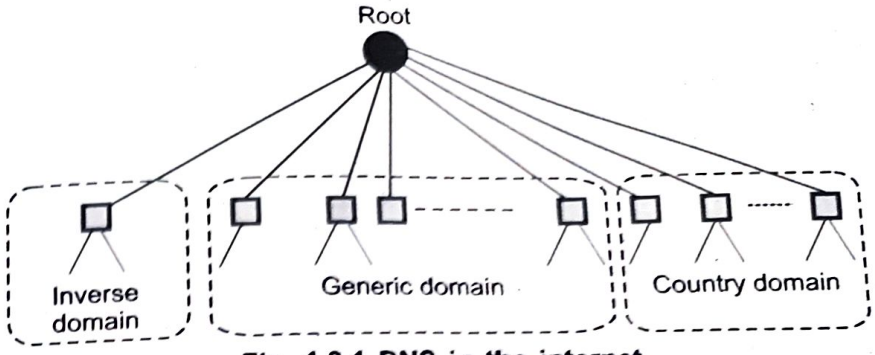

Fig. 1.3.1 DNS in the internet

- Generic domain labels are as follows

Sr. No.	Label	Description
1.	com	Commercial organization
2.	edu	Educational organization
3.	gov	Government Institutions
4.	int	International organizations
5.	mil	Military group
6.	net	Network support centers
7.	org	Nonprofit organization

- Fig. 1.3.2 shows the generic domains

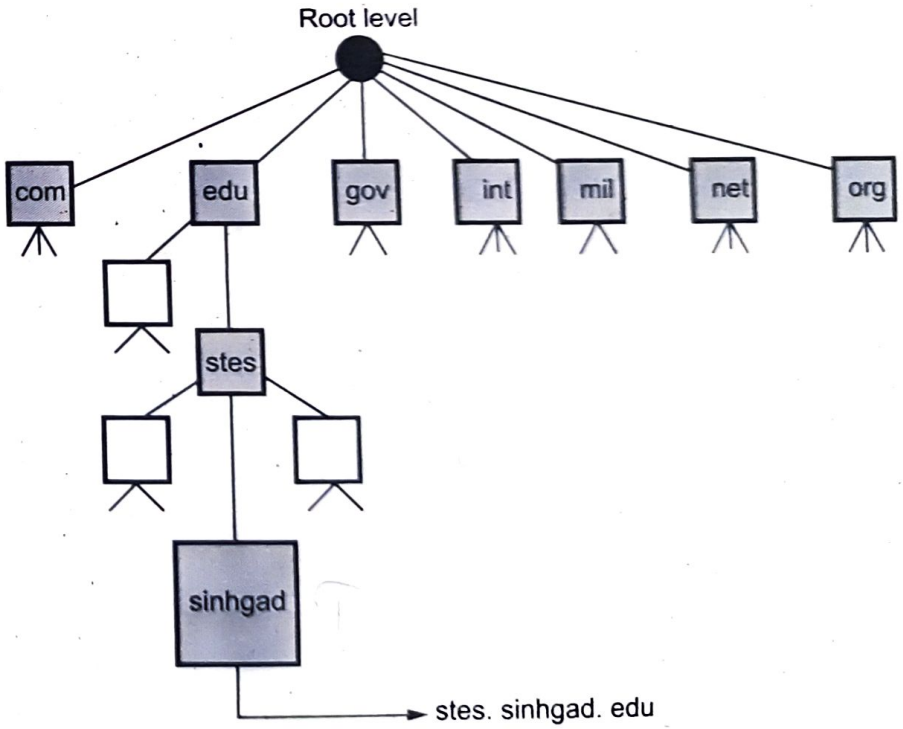

Fig. 1.3.2 Generic domains

Country Domains

- It uses two character country abbreviations at first level. Second level labels can be more specific, national destinations. For India, the country domain is in.
- Fig. 1.3.3 shows country domains.

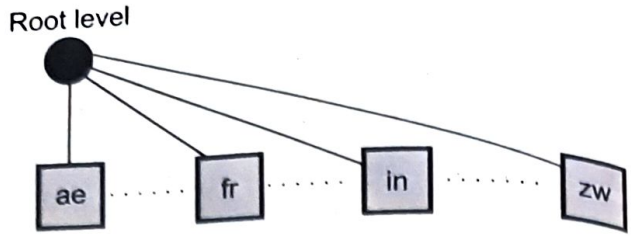

Fig. 1.3.3 Country domains

Inverse Domain

- Used to map an address to a name.
- Example : When a client send a request to the server for doing a particular task, server finds the list of authorized client. The list contains only IP address of the client.
- Server send a query to the inverse DNS server and ask for a mapping of address to name for authorized client list.
- The above query is called an inverse or pointer query.
- The pointer query is handled by the first level node called arpa. The second level is also one single node named in-addr. The rest of the domain defines IP addresses.

Fig. 1.3.4 shows inverse domain.

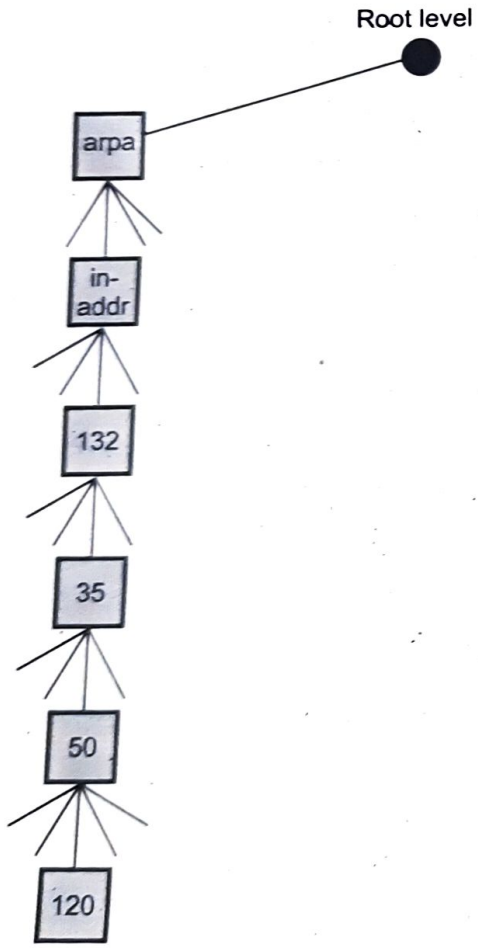

Fig. 1.3.4 Inverse domain

1.3.3 Name Spaces

- Name spaces are of two types : Flat name spaces and hierarchical names.
- The name assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

i) Flat name spaces :

- The original set of machines on the Internet used flat namespaces.
- These namespaces consisted of sequence of characters with no further structure.
- A name is assigned to an address.
- **Advantage :**
  1. Names were convenient and short.
- **Disadvantages :**
  1. Flat name spaces cannot generalize to large sets of machines because of the single set of identifiers.
  2. Single central name authority was overloaded.
  3. Frequent name-address binding changes were costly and cumbersome.

ii) Hierarchical names

- The partitioning of a namespace must be defined in such a way that it :
  - Supports efficient name mapping.
  - Guarantees autonomous control of name assignment.
- Hierarchical namespaces provides a simple yet flexible naming structure.
- The namespace is partitioned at the top level.
- Authority for names in each partition are passed to each designated agent.

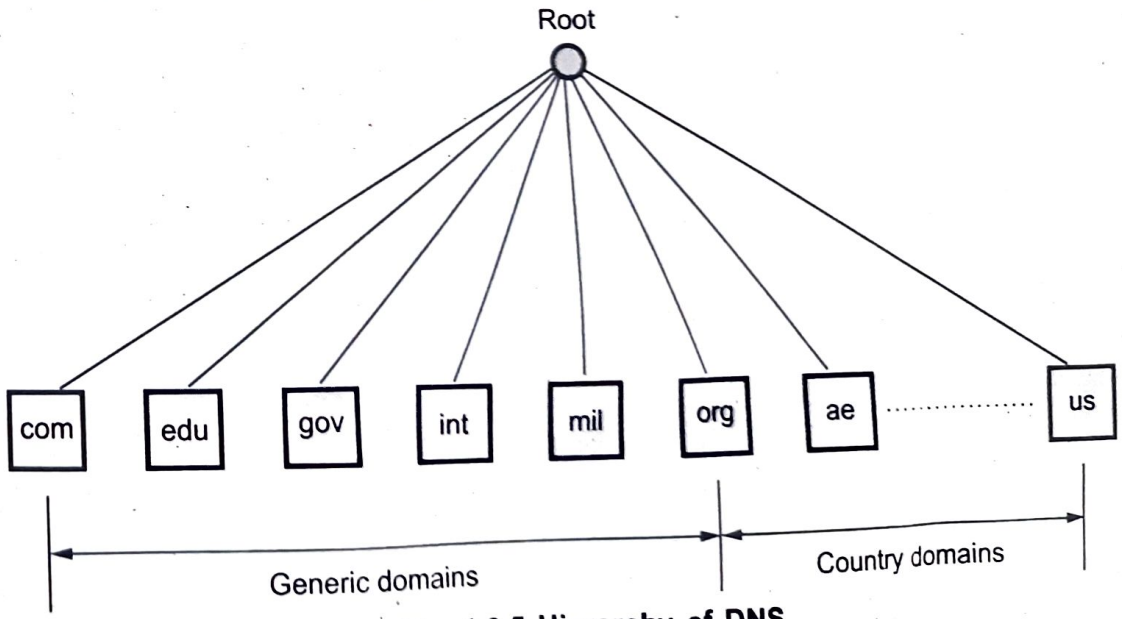

Fig. 1.3.5 Hierarchy of DNS

- The names are designed in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels.
- The top level domains are divided into three areas :
  1. Arpa is a special domain used for the address-to-name mappings.
  2. The 3 character domains are called the generic domains.
  3. The 2 character domains are based on the counter codes found in ISO 3166. These are called the country domains.
- Fig. 1.3.5 shows the hierarchy of DNS.

1.3.4 Domain Name Space

- In DNS, names are defined in an inverted tree structure with the root at the top. The tree can have only 128 levels : Level 0 to Level 127.

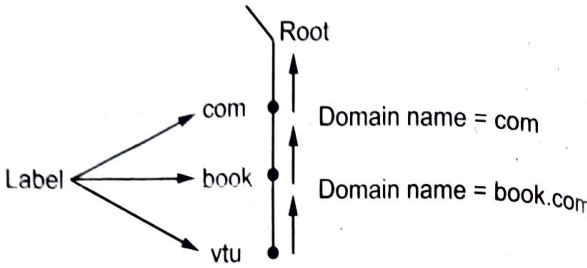

Fig. 1.3.6 Domain names and label

- Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string i.e. empty string.
- Each node in the tree has a domain name, a full domain name is a sequence o labels separated by dots(.). Fig. 1.3.6 shows the domain names and labels.
- In fully qualified domain name, label is terminated by a null string. Full Qualified Domain Name (FQDN) contains the full name of host. All labels are par of FQDN.
- Partially Qulified Domain Name (PQDN) : In this label is not terminated by a null string. It always start from node. A domain name does not include all the level between the host and the root node. For example, vtu.book.com.

Hierarchy of Name Servers

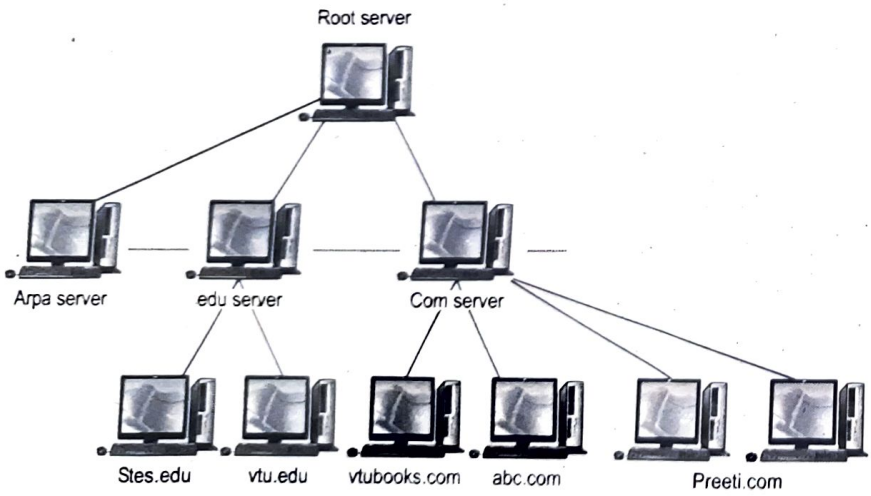

Fig. 1.3.7 Hierarchy of name server

- To distribute the information among many computers, DNS servers are used. Creates many domains as there are first level nodes. Fig. 1.3.7 shows hierarchy of name servers.
- **Zone** : Server have some authority and also responsible for operation. Server creates database, which is called zone file. Server maintain all the information about node of that domain.
- Fig. 1.3.8 shows domain with zone.
- Domain and zone are same if server accepts responsibillity for a domain and does not divide the domain into **subdomain.**
- Domain and zone are different, if a server divides its domain into subdomains and delegates part of its authority to other server.

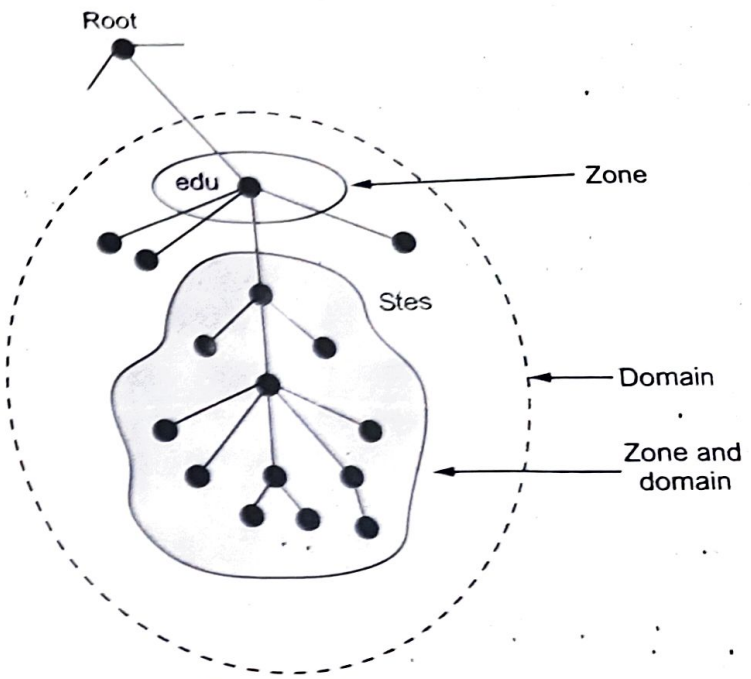

Fig. 1.3.8 Domain and zone

- **Root server :** If zone consists of the full tree then that zone server is called root server. Root server do not maintain any information about domains.
- DNS uses two types of servers :
  1. Primary server 2. Secondary server
- **Primary server :** This server keeps a file about the zone for which it is responsible and have authority. It performs operation on zone file like create, update and maintaining.
- **Secondary server :** It loads all information from the primary server. Secondary server can not perform any operation on zone file.

1.3.5 Recursive and Interative Resolution

- DNS is designed as a client server application. A host that needs to map an address to a name or a name to an address calls a DNS client named a **resolver.**

Working :

- Name resolving must also include the type of answer desired (specifying the protocol family is optional).
- The DNS partitions the entire set of names by class (for mapping to multiple protocol suites).

- Naming items is required since one cannot distinguish the names of subdoma from the names of individual objects or their types.

Mapping Domain Names to Addresses :

a) The DNS also includes an efficient, reliable, general purpose, distributed syst for mapping names to addresses using an independent co-operative system cal name servers.

b) Names Servers - are server programs that translate names-to-addres (maps DN => IP addresses) and usually executes on a dedicated processor.

c) Name Resolvers - client software that uses one or more name servers in gettin mapped name.

d) Domain name servers are arranged in a conceptual tree structure that correspo to the naming hierarchy.

Recursive Resolution

- A client request complete translation.
- If the server is authority for the domain name, it checks its database and respon
- If the server is not authority, it sends the request to another server and waits the response.
- When the query is finally resolved, the response travel back until it finally reac the requesting client. This is called recursive resolution.
- Fig. 1.3.9 shows the recursive resolution.

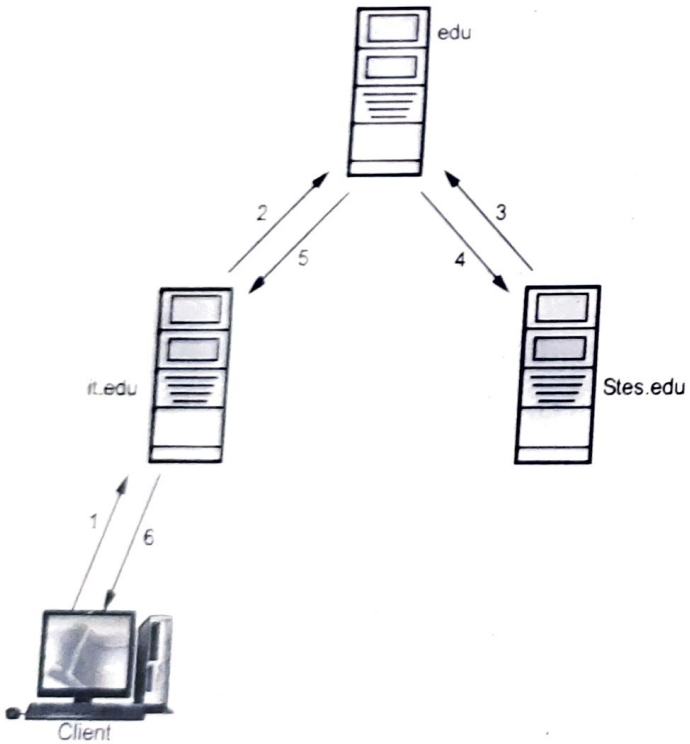

Fig. 1.3.9 Recursive resolution

Iterative Resolution

- Only a single resolution is made and returned (not recursive).
- Client must now explicitly contact different name servers if further resolution is needed.
- If the server is an authority for the name, it sends the answer. If it is not, it returns the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. This process is called iterative resolution because the client repeats the same query to multiple servers.
- Fig. 1.3.10 shows iterative resolution.

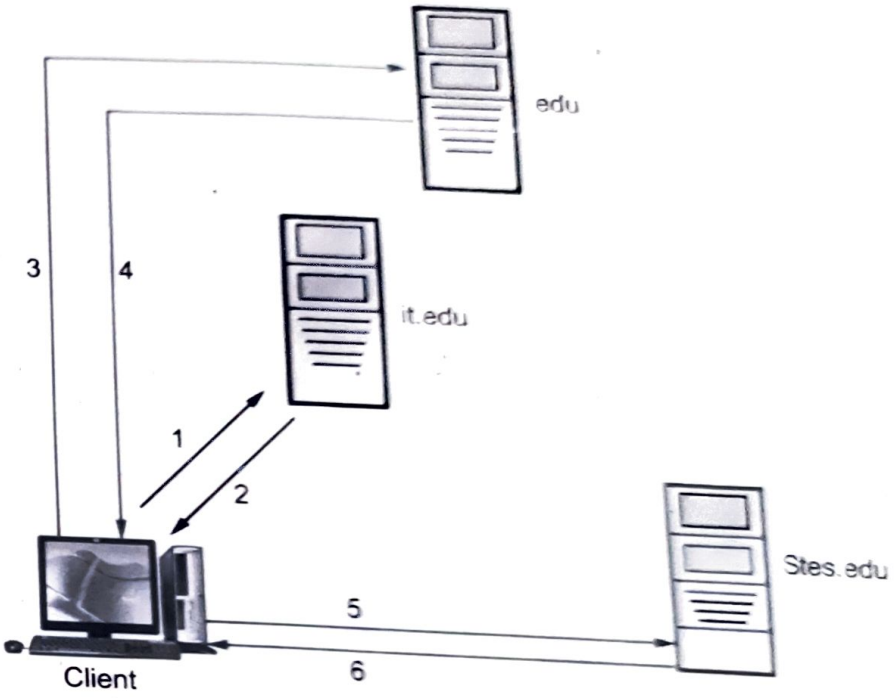

Fig. 1.3.10 Iterative resolution

- Conceptually, name resolution proceeds in a top - down fashion.
- Name resolution can occur in one of two different ways : Recursive resolution and Iterative resolution
- Name servers use name caching to optimize search costs.
- Time To Live (TTL) is used to determine a guaranteed name binding during it's time interval. When time expires, the cache name binding is no longer valid, so the client must make a direct name resolution request once again.

Reverse Name Resolution :

- Reverse name resolution is important task of DNS on the internet or the translation of IP addresses back to domain names. For example, servers can

determine and record the full domain name of machine connecting to them over the network.

- It is not efficient to use the same set of DNS records for reverse name resolution. Instead, a separate domain called "IN-ADDR.ARPA" has been set aside to provide a hierarchy for translating IP addresses into names.
- A DNS lookup of "barg.oo.msstate.edu" would reveal it has the IP address "130.19.60.10". If one has the IP address and wishes to know the name, one must perform a DNS lookup of "10.60.19.130. in -addr.arpa", which will return the name.
- Reverse name resolution fields use the PTR resource record, which points to the correct position in the normal DNS space. The hierarchy under "IN-ADDR.ARPA" can be delegated of course just like any other domain.
- To obtain the IP address of a named server, each host has a client protocol known as the name resolver. On receipt of the name, the client application protocol posses it to the name resolver using the standard interprocess communication primitive supported by the local operating system.
- The resolver then creates a resolution request message in the standard message format of the domain name server protocol.
- A resolver can have multiple request outstanding at any time. Hence the identification field is used to relate a subsequent response message to an earlier request message.
- The name resolver passes the request message to its local domain name server using TCP/IP. If the request is for a server on this network, the local domain name server obtains the corresponding IP address from its DIB and returns it in a reply message.

1.3.6 Message Format

- *Messages are sent between domain clients and domain servers with a specific format.*
- *All messages of this format are used for name resolution and naming queries.*
- *Question sent by the client and answers provided by the server are included within different fields of the same message.*
- *DNS has two types of messages : Query and Response. Both types have the same format.*
- *The query message consists of the header and the question records, the response message consists of a header, question record, answer record, authoritative record and additional records.*

- Fig. 1.3.11 shows the query and response messages.

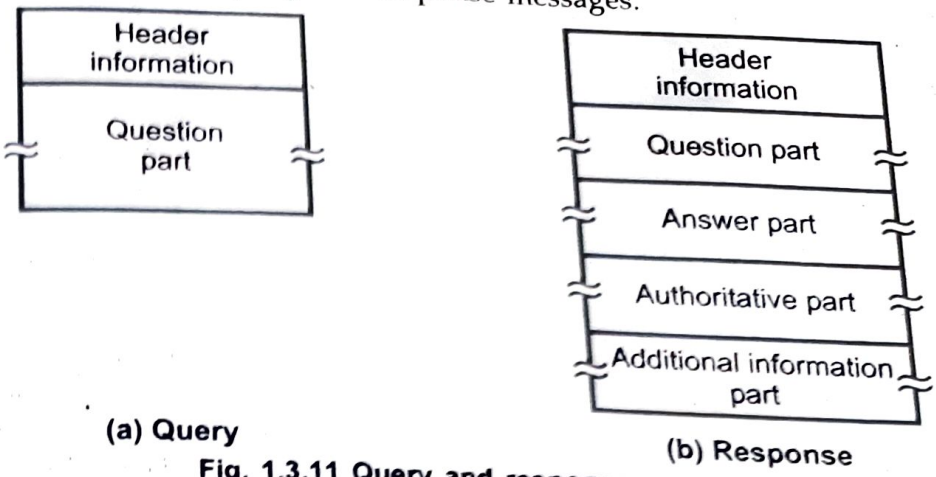

Fig. 1.3.11 Query and response message

- Fig. 1.3.12 shows the header format of the DNS.

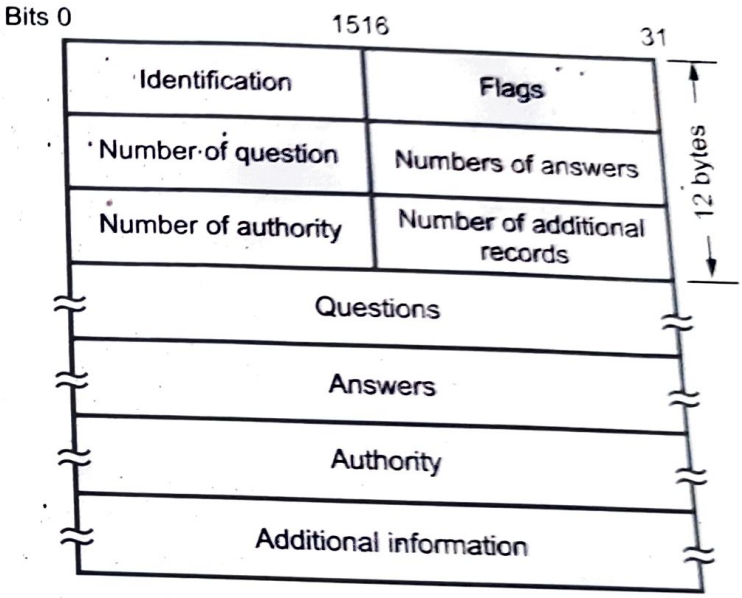

Fig. 1.3.12 General format of DNS

- **Identification :** It is 16 bits fields and unique value used by the client to match responses to queries.
- **Flags :** It is the collection of subfields that define the type of messages and type of the answers requested and so on.
- Number of question record contains the number of queries in the question section of the message.
- Number of answer record contains the number of answer records in the answer section of the response message.

- Number of authority record contains the number of authority records in the authoritative section of the response message.
- Number of additional records contains the number of additional records in the additional section of the response message. The message has a fixed 12-byte header followed by 4 variable length fields. The identification field is set by client and returned by the server. It lets the client, match responses to requests.
- Fig. 1.3.13 flag fields in DNS header.

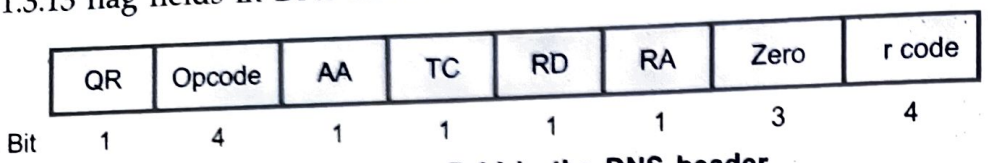

Fig. 1.3.13 Flags field in the DNS header

- The flags field is divided into 8 parts.

QR = 0 For message is a query

= 1 It is response

Opcode = 0 Standard query

= 1 Inverse query

= 2 Server status request

AA = Authoritative answer

TC = Truncated

RD = Recursive query

RA = Recursion available

r code = Return code

- RD field is 1-bit and can be set in a query and is then returned in the response. This flag tells the name server to handle the query itself, called a recursive query.
- RA is a 1-bit field and set to 1 in the response if the server support recursion. There is a 3-bit field that must be zero.
- r code is a 4-bit field. The common value are 0 for no error and 3 for name error. A name error is returned only from an authoritative name server and means the domain name specified in the query does not exist.
- The next four 16-bit fields specify the number of entries in the four variable length fields that complete the record.

1.3.7 Resource Records

- Different types of resource records are used in DNS. An IP address has a type of A and PTR means pointer query.
- There are about 20 different types of resource records available. Some PR are listed below.
  1) A = It defines an IP address. It is stored as a 32 - bit binary value.
  2) CNAME = "Canonical name". It is represented as a domain name.
  3) HINFO = Host information, two arbitrary character strings specifying the CPU and operating system (OS).
  4) MX = Mail exchange records. It provide domain willing to accept e-mail.
  5) PTR = Pointer record used for pointer queries. The IP address is represented as a domain name in the in-addr.arpa domain.
  6) NS = Name Server record. These specify the authoritative name server for a domain. They are represented as domain names.

1.3.8 Name Servers

- When a resolver has a query about a domain name, it passes the query to one of the local name servers. If the domain is remote and no information about the requested domain is available locally, the name server sends a query message to the top level name server for the domain requested.
- Fig. 1.3.14 shows the eight steps for resolving the remote name. (See Fig. 1.3.14 on next page).
- A resolver on flits.cs.vu.nl wants to know the IP address of the host linda.cs.yale.edu.

Steps

1. It sends a query to the local name server cs.vu.nl. This query contains the domain name, sought, the type (A) and the class (IN).
2. and 3. Suppose the local name server has never had a query for this domain before and knows nothing about it. It may ask a few other nearby name servers, but if none of them know, it sends a UDP packet to the server for edu given in its database, edu-server.net. This server knows all its children, so it forwards the request to the name server for yale.edu.
4. This forwards the request to cs.yale.edu, which must have the authoritative resource records.
5. to 8. Each request is from a client to a server, the resource record requested works its way back in these steps.

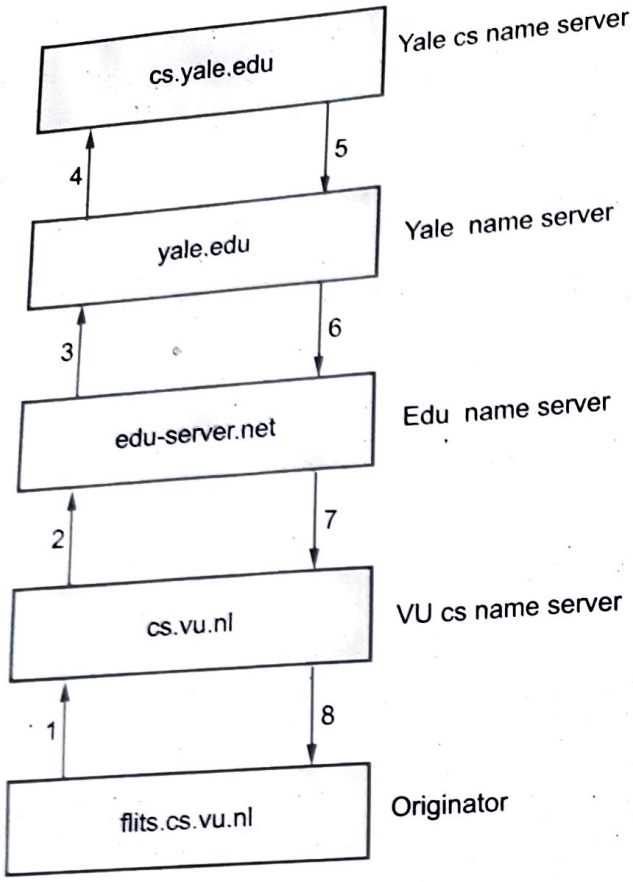

Fig. 1.3.14 Remote name resolve

1.3.9 LDAP

- LDAP is Lightweight Directory Access Protocol. It provides X-500 features. LDA is an application-level protocol that is implemented directly on top of TCP.
- It stores entries, which is similar to objects. Each entry must have a distinguishe name, which un-equally identifies the entry. Entries can also have attributes.
- LDAP provides binary, string and time types. It allows the defination of obje classes with attribute name of types. Entries are organized into a director information tree, according to their distinguished names.
- LDAP defines a network protocol for carrying out data defination an manipulation.
- LDAP has been widely adopted, particularly for internet directory services. provides secured access to directory data through authentication.

1.3.10 Dynamic Domain Name System (DDNS)

- DDNS is a service that maps internet domain names to IP addresses. DDNS serves a similar purpose to DNS : DDNS allows anyone hosting a Web or FTP server to advertise a public name to prospective users.
- Unlike DNS that only works with static IP addresses, DDNS works with dynamic IP addresses, such as those assigned by an ISP or other DHCP server.
- DDNS is popular with home networkers, who typically receive dynamic, frequently-changing IP addresses from their service provider.
- To use DDNS, one simply signs up with a provider and installs network software on their host to monitor its IP address.
- Compared to ordinary DNS, the disadvantage of DDNS is that additional host software, a new potential failure point on the network, must be maintained.

1.3.11 Advantages of DNS

1. DNS has hierarchical structure and database.
2. DNS has small and manageable zones.
3. It is scalable.
4. DNS helps in eliminating host tables.
5. It is consistant on all hosts.
6. The Internet couldn't exist without it.
7. Easy to implement with minimal configuration changes in DNS server.

1.4 File Transfer Protocol (FTP)

- FTP is designed for distributing files to a number of users. FTP uses a client-server system, in which files are stored at a central computer and transferred between that computer and other, widely distributed computers.
- The central computer runs FTP server software and widely distributed computer runs FTP client software. FTP is interactive.
- The FTP program accepts a sequence of commands. To interact with a remote computer, a user must identify the computer and allow FTP to establish contact.
- FTP uses TCP/IP software to contact the computer. FTP provides 58 separate commands, an average user only needs to understand the three basic commands to connect to a remote computer, retrieve a copy of a file and exit the FTP program. Following are the list of commands.

Sr. No.	Command	
1.	open	Connect to a rem
2.	get	Retrive a file from the computer
3.	bye	Terminate the connection and leave the FTP program

Table 1.4.1

- These 3 commands are used for FTP clients for file transfer and terminates the connection after down loading or up loading the file. When user transfer a file by either uploading or downloading-user use one of two modes.
- User may need to select the mode. The modes is as follows.
  1) ASCII mode 2) Binary (Image) mode
- ASCII mode is used for transferring a text files including HTML files. Different computer systems use different characters to indicate the ends of lines. In ASCII mode, the FTP software automatically adjusts line endings for the system to which the file is transferred.
- In binary mode, transferring of files consists of anything but unformatted text. In this mode, the FTP software does not make any changes to the contents of the file during transfer.
- Use binary mode when transferring graphic files, audio files, video files, program or any other kind of file other than plain text.
- Choosing between binary and ASCII transfer can be difficult. When unsure about the content of file, enter the FTP command binary before transferring the file. FTP uses the client-server approach.

akes an FTP program on the computer, instructs it to contact a remote the transfer of one or more files.

TCP to connect on FTP server file transfer, the

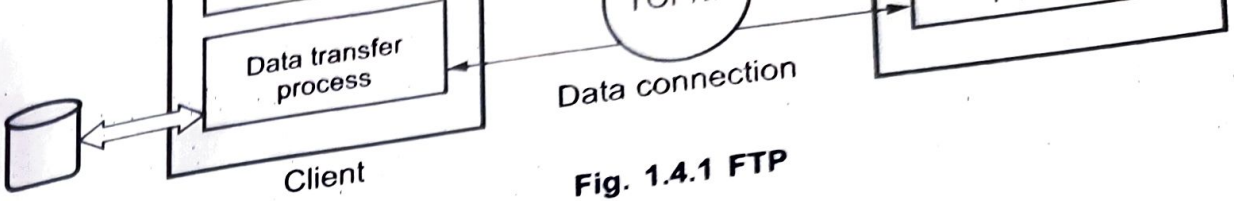

Fig. 1.4.1 FTP

Detail steps of FTP

- FTP client contacts FTP server at port 21 specifying TCP as transport protocol.
- Client obtain authorization over control connection.
- Client browse remote directory by sending commands over control connection.
- When server receives a command for a file transfer, the server open a TCP data connection to client.
- After transfering one file, server closes connection.
- Server opens a second TCP data connection to transfer another file.
- FTP server maintains state i.e. current directory, earlier authentication.
- Fig. 1.4.2 shows the FTP data connection

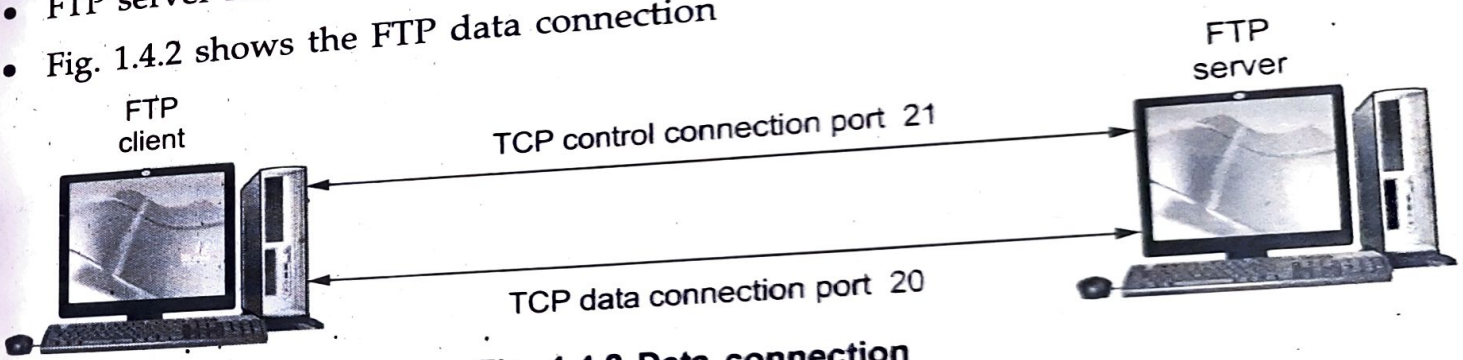

Fig. 1.4.2 Data connection

5.	dir or ls	Provides a directory listing of the current working directory
6.	help	Displays a list of all client FTP commands
7.	remotehelp	Displays a list of all server FTP commands
8.	type	Allows the user to specify the file type
9.	struct	Specifies the files structure

1.4.1 Trivial File Transfer Protocol (TFTP)

- It is a UDP-based file transfer program which is frequently used to allow diskless hosts to boot over the network. TFTP is implemented by the tftp client program and by the tftp server program. As TFTP has no user authentication, it may be possible for unwanted file transfer to occur. It is a significant threat that TFTP may be used to steal password files.
- TFTP is a simple protocol to transfer files. It is implemented on top of the Internet User Datagram Protocol (UDP or Datagram). The design of a TFTP is small and easy to implement, therefore, lacks most of the features of a regular FTP.
- TFTP can only read and write files (or mail) from/to a remote server. It cannot list directories, and currently it has no provisions for user authentication.
- In TFTP, any transfer always begins with a request to read or write a file, which serves to request a connection. When the server grants the request, the connection is opened and the file is sent in fixed length blocks of 512 bytes.
- Each data packet contains one block of data, and it must be acknowledged by an acknowledgement packet before sending the next packet.
- A data packet of less than 512 bytes signals causes termination of a transfer. If a packet is lost in the network, the intended recipient will timeout and may

1.4.2 Difference between FTP and TFTP

Sr. No.	FTP	TFTP
1.	FTP uses two connections	TFTP uses one connection
2.	Provides many commands	Provides only five commands
3.	Uses TCP	Uses UDP
4.	Client must login to server	No login procedure
5.	Allow for user authentication	TFTP does not allow for user authentication
6.	FTP provides a reliable service	TFTP must handle it's own retransmissions

1.5 Hypertext Transfer Protocol (HTTP)

- The standard web transfer protocol is Hyper Text Transfer Protocol (HTTP).
- The HTTP protocol consists of two fairly distinct items: The set of requests from browsers to servers and the set of responses going back the other way.
- All the newer versions of HTTP support two kinds of requests: Simple requests and full requests. A simple request is just a single GET line naming the page desired, without the protocol version.
- The response is just the raw page with no headers, no MIME, and no encoding. To see how this works, try making a Telnet connection to port 80 of www.w3.org and then type.

  GET /hypertext/www/TheProject.html

  but without the HTTP/1.0 this time. The page will be returned with no indication of its content type. This mechanism is needed for backward compatibility. It will decline as browsers and

- When accessing general objects, additional object-specific methods may also b available. The names are case sensitive, so, GET is a legal method but get is not.

HTTP Transaction

- HTTP uses the services of TCP. HTTP is a stateless protocol.
- The client initilizes the transaction by sending a request message. The serve replies by sending a response.
- Fig. 1.5.1 shows HTTP transaction

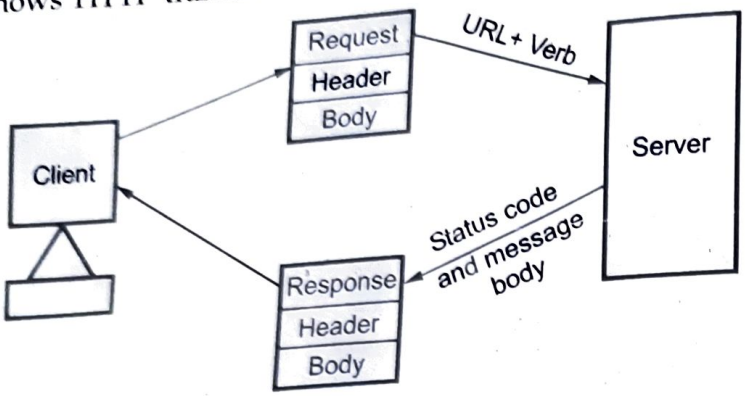

Fig. 1.5.1 HTTP transaction

1.5.1 HTTP Messages

- HTTP messages are two types
  1. Request 2. Response
- Both message type used same format.
- Request **message** consists of a request line, headers and a body. Fig. 1.5.2 shows request message.

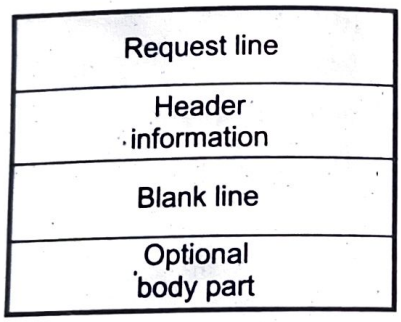

Fig. 1.5.2 Request message

Request line

- *Request line defines the*
  1. *Request type*
  2. *Resource*
  3. *HTTP version*
- *Request type categorizes the request message into several methods for HTTP version 1.1.*
- *Fig. 1.5.3 shows the request line.*

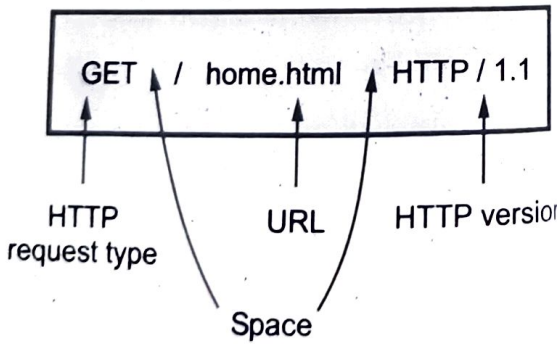

Fig. 1.5.3 Request line

- URL is a standard for specifying any kind of information on the internet. The URL define four things.
  1. Method 2. Host computer
  3. Port 4. Path

Fig. 1.5.4 shows the URL.

- The method is the prototocol used to retrieve the document. Several different protocols can retrieve a document, among them are FTP and HTTP.
- The host is the computer where the information is located, although the name of the computer can be alias.

Fig. 1.5.4 (a) URL

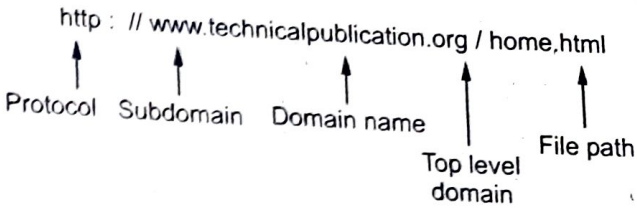

Fig. 1.5.4 (b) URL example

- Web pages are usually stored in computers and computers are given alias nam that usually begin with the character www.
- The URL can optionally contain the port number of the server.
- Path is the path name of the file where the information is located.
- The request type field in a request message defines several kinds of messa referred to as methods.

Sr. No.	Method	Purposes
1.	GET	Used when the client wants to retrieve a document from the server. Server responds with the contents of the document.
2.	HEAD	Used when client wants some information about a document but not the document itself.
3.	POST	Used by the client to provide some information to the server i.e. input to the server.
4.	PUT	Used by the client to provide a new or replacement document to be stored on the server.
5.	PATCH	Similar to PUT except that the request contains a list of differences that should be implemented in the existing file.
6.	DELETE	Removes a document on the server.

8.	MOVE	Move a file to another
9.	LINK	Creates a link or links from a document to another location.
10.	UNLINK	UNLINK method deletes links created by the LINK method.
11.	OPTION	This method is used by the client to ask the server about available options.

- The GET method requests the server to send the page (by which we mean object, in the most general case), suitably encoded in MIME. However, if the GET request is followed by an If-Modified-Since header, the server only sends the data if it has been modified since the data supplied. Using this mechanism, a browser that is asked to display a cached page can conditionally ask for it from the server, giving the modification time associated with the page.
- If the cache page is still valid, the server just sends back a status line announcing that fact, thus eliminating the overhead of transferring the page again.
- The HEAD method just asks for the message header, without the actual page. This method can be used to get a page's time of last modification, to collect information for indexing purposes, or just to test a URL for validity. Conditional HEAD request do not exist.
- The PUT method is the reverse of GET : Instead of reading the page, it writes the page. This method makes it possible to build a collection of web pages on a remote server.
- The body of the request contains the page. It may be encoded using MIME, in which case the lines following the PUT might include content type and authentication headers, to prove that the caller indeed has permission to perform the requested operation.
- Somewhat similar to PUT is the POST method. It too bears a URL, but instead of replacing the existing data, the new data is "appended" to it in some generalized sense.
- Posting a message to a news group or adding a file to a bulletin board system are example of appending in this context. It is clearly the intention here to have the Web take over the functionality of the USENET news system.
- DELETE does what you might expect; it removes the page. As with PUT authentication and permission play a major role here. There is no guarantee that

existing

Response Message

- Fig. 1.5.5 shows the response message. It contains a status line, a header and body.
- Status line defines the status of the response message. It consists of the
  a. HTTP version b. Space
  c. Status code d. Space e. Status phrase

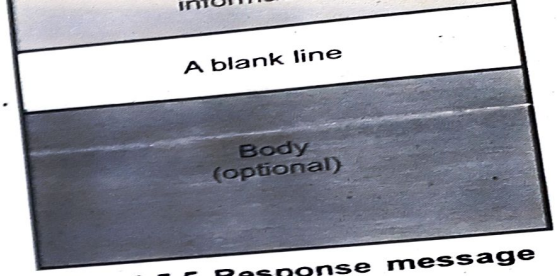

Fig. 1.5.5 Response message

1.5.2 HTTP Headers

- Header can be one or more header lines. Each header line is made of a header name, a colon, a space and a header value.
- The header exchange additional information between the client and the server.
- A header line belongs to one of four categories : general header, request header, response header and entity header.
- Fig. 1.5.6 shows the header format.

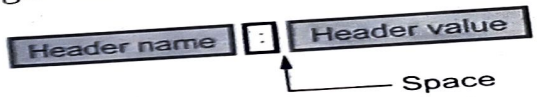

Fig. 1.5.6 Header format

- *General header* includes general information about the message. Request and a response both contains general header.
- *Response header* can be present only in a response message. It specifies the servers configuration and special information about the request.
- Request header can be present only in a request message. It specifies the clients configuration and the client preferred document format.
- *Entity header* gives information about the body of the document. It is mostly present in response messages, some request message, such as POST and PUT methods, that contain a body also use this type of header.

- Fig. 1.5.7 shows the headers.

Status line	HTTP/1.1 300 OK
General Headers	Date : Wed, 8 Oct 2014 13:00:13 GMT
	Connection : close
Response Headers	Server : Apache /1.3.27
	Accept-Ranges : bytes
Entity Headers	Content-Type : text/html
	Content-Length : 200
	Last-Modified : 2 Oct 2014 13:00:13 GMT
Blank Line	
Message Body	<html>
	<head>
	<title> Welcome to the India <title>
	<head>
	<body>

Fig. 1.5.7 Response message header

1.5.3 Persistent and Non-persistent Connection

- HTTP connections are of two types
  1. Persistent HTTP
  2. Non-persistent HTTP

1.5.3.1 Non - persistent Connections

- In this type of connection, one TCP connection is made for eac request / response.
- Suppose the page consists of a base HTTP file and ten JPEG images and that a 11 of these objects reside on the same server.
- Suppose the URL for the base HTML file is
  www.vtubooks.com / ITDept / home.index

The sequence of events are as follows :

1. The HTTP client initiates a TCP connection to the server www.vtubook.com o port number 80. It is default port number for HTTP.
2. HTTP client sends an HTTP request message to the server via the socke Request mesage includes the path name/ITDept/home.index.

3. HTTP server receives the request message via the socket.
4. HTTP server tells TCP to close the TCP connection.
5. HTTP client receives the response message. The TCP connection terminates.
6. The first four steps are then repeated for each of the referenced JPEG objets.

- As the brower receives the web pages, it displays the page to the user.

Round Trip Time (RTT)

- RTT is the time it takes for a small packet to travel from client to server and then back to the client.
- RTT includes packet propagation delays, packet queuing delays in intermediate routers and switches and packet processing delays.
- Fig. 1.5.8 shows operation when user clicks on a hyperlink.

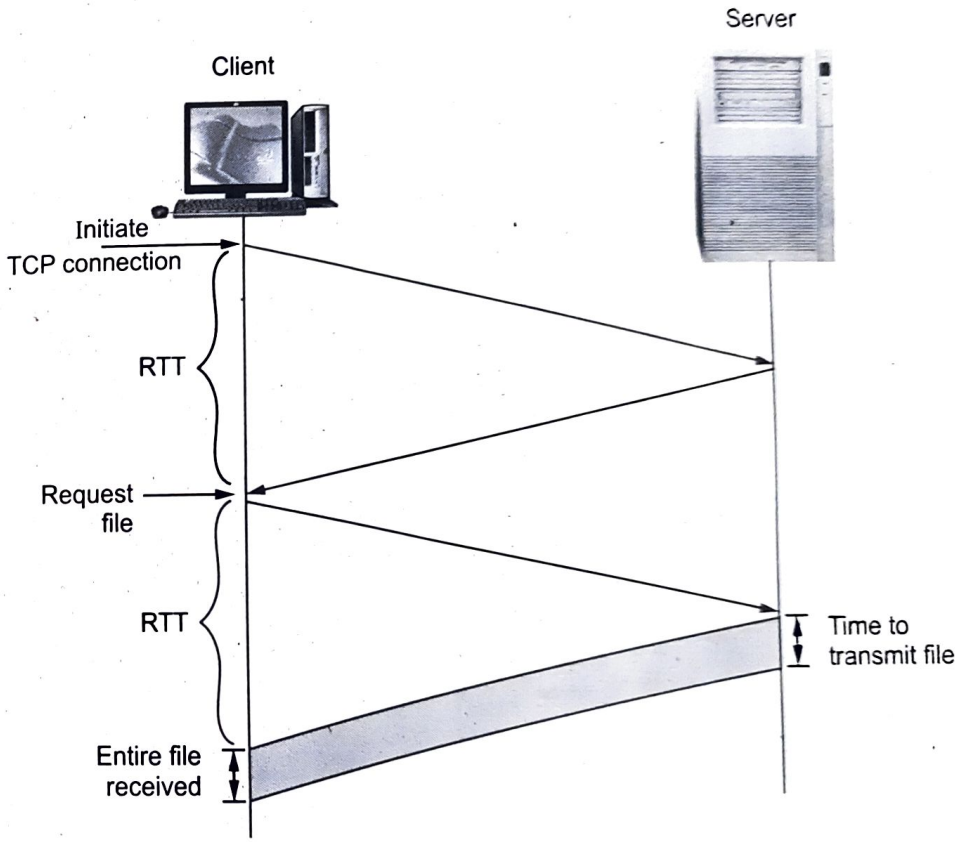

Fig. 1.5.8 Calculation for requesting file

- Browser to initiate TCP connection between the browser and the web server. It requires three way handshake.
- The client sends a small TCP segment to the server.
- The server acknowledges and responds with a small TCP segment.
- Finally, the client acknowledges back to the server.

- The initial design HTTP 1.0 uses nonpersistent connections. The TCP connection is closed after each request/response interaction.
- Each subsequent request from the same client to the same server involves the setting up and tearing down of an additional TCP connection.

Disadvantages of non - persistent

1. TCP processing and memory resource wasted in the server and the client.
2. It requires delay of 2 RTT associated with the transfer of each object.
3. Each TCP connection setup involves the exchange of three segments between client and server machines.

1.5.3.2 Persistent Connection

- HTTP 1.1. made persistent connections the default mode.
- The server now keeps the TCP connection open for a certain period of time after sending a response.
- This enables the client to make multiple requests over the same TCP connection and hence avoid the inefficiency and delay of the nonpersistent mode.

Types of persistent connections

- There are two versions of persistent connections :
  1. Without pipelining 2. With pipelining

Without pipelining

- The client issues a new request only when the previous response has been received.
- The client experiences one RTT in order to request and receive each of the referenced objects.
- Disadvantage : TCP connection is idle i.e. does nothing while it waits for another request to arrive. This idling wastes server resoruces.

With pipeling

- Default mode of HTTP 1.1. uses persistent connections with pipeling.
- Client issues a request as soon as it encounters a references. The HTTP client can make back to back requests for the referenced objects.
- It can make a new request before receiving a response to a previous request.
- When the server receives the back-to-back requests, it sends the objects back-to-back.
- It uses only one RTT.

- Pipelined TCP connection remains idle for a smaller fraction of time.
- **Persistent HTTP** connections have a number of **advantages**.
  1. By opening and closing fewer TCP connections, CPU time is saved in routers and hosts.
  2. Requests and responses can be pipelined on a connection.
  3. Network congestion is reduced by reducing the number of packets caused by TCP opens.
  4. Latency on subsequent requests is reduced.

roxy server

- HTTP supports the proxy servers. A proxy server is a computer that keeps copies of responds 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 proxy server, the client must be configured to access the proxy instead of the target server.

.5.4 Difference between Persistent and Non-persistent

Sr. No.	Persistent HTTP	Non-persistent HTTP
1.	Persistent version is 1.1.	Non-persistent HTTP version is 1.0.
2.	It uses one RTT.	It uses two RTT.
3.	TCP connection is not closed.	TCP connection is closed after every request-response.
4.	Client make multiple request over the same TCP connection.	Client make multiple request over the multiple TCP connection.
5.	It is default mode.	It is not default mode.
6.	Request methods are GET, HEAD, POST, PUT, DELETE, TRACE and OPTIONS.	Request methods used are GET, POST and HEAD.

1.6 Simple Mail Transfer Protocol (SMTP)

- SMTP is application layer protocol of TCP/IP model.
- SMTP transfers message from sender's mail servers to the recipients mail servers.
- SMTP interacts with the local mail system and not the user.
- SMTP uses a TCP socket on port 25 to transfer e-mail reliably from client server.
- E-mail is temporarily stored on the local and eventually transfered directly receiving server.
- Client / Server interaction follows and commond/reponse paradigm.
  a] Commands are plain ASCII text.
  b] Responses are a status code and an optional phase.
  c] Command and response lines terminated with CRLF.
- Mail client application interacts with a local SMTP server to initiate the delivery an e-mail message.
- There is an input queue and an output queue at the interface between the loc mail system and the client and the server parts of the SMTP.
- The client is concerned with initiating the transfer of mail to another system whil server is concerned with receiving mail. Before the e-mail message can b transferred, the application process must be set up a TCP connection to the loc SMTP server. The local mail system retains a mailbox for each user into which th user can deposit or retrieve mail. Mail handling system must use a uniqu addressing system.
- Adderssing system used by SMTP consists of two parts : A local part and a globa part. The local part is the user name and is unique only within that local mai system. Global part of the address is the domain name. Domain name is identif of the host, must be unique within the total Internet.
- SMTP uses different types of component. They are MIME and POP.

Scenario : Alice sends message to Bob

1. Alice uses User Agent (UA) to compose message and to bob@sinhgad.edu.
2. Alice's UA sends message to her mail server, message placed in message queue.
3. Client side of SMTP opens TCP connection with Bob's mail server.
4. SMTP client sends Alice's message over the TCP connection.
5. Bob's mail server places the message in Bob's mailbox.
6. Bob invokes his user agent to read message.

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

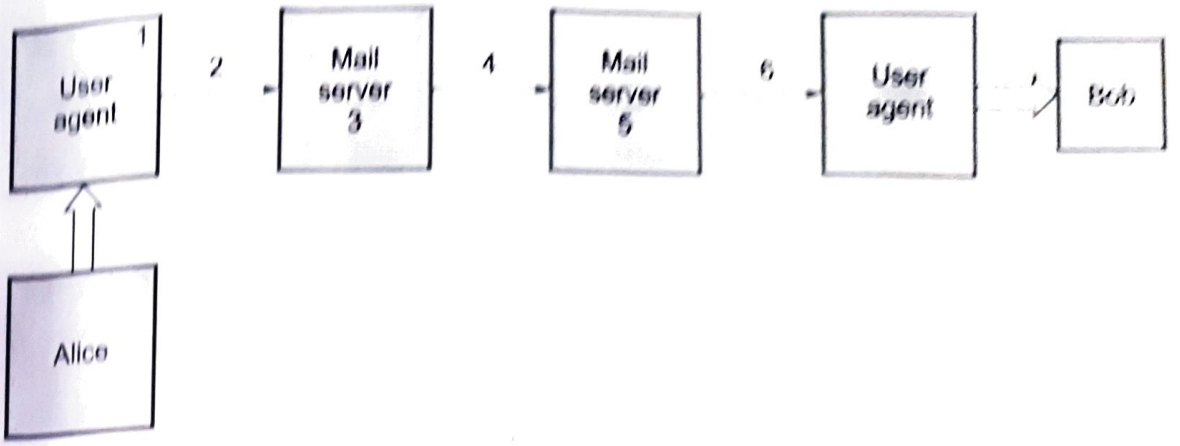

Fig. 1.6.1 Message scenario

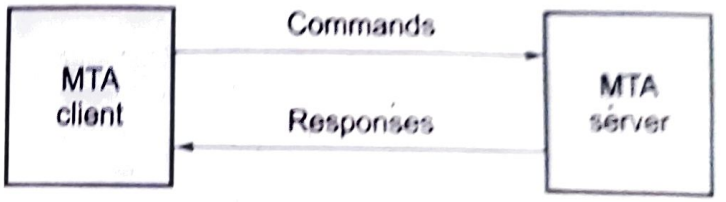

Fig. 1.6.2 Command / Response

- Each command or reply is terminated by a two character end of line token.
- Commands are sent from the client to the server. SMTP defines 14 commands. SMTP commands consist of human readable ASCII strings.
- SMTP commands are as follows
  - i) **HELO** : Initiate a mail transaction, identifying the sender to the recipient.
  - ii) **MAIL FROM** : Tells the remote SMTP that a new mail transaction is beginning.
  - iii) **RCPT TO** : The sending SMTP sends a RCPT command for each intended receiver.
  - iv) **DATA** : If accepted, the sender transfers the actual message. End of message is indicated by sending a " • " on a line by itself.
  - v) **QUIT** : Terminate the connection.

Sample SMTP Interaction

- Following are messages exchanged between an SMTP client (C) and an SMTP server (S).

- The host name of the client is iresh.fr and the host name of the server sinhgad.edu.

S : 220 sinhgad.edu

C : HELO iresh.fr

S : 250 Hello iresh.fr, pleased to meet you

C : MAIL FROM : <rupali@iresh.fr>

S : 250 rupali@iresh.fr ... sender ok

C : RCPT TO : < rakshita@sinhgad.edu>

S : 250 rakshita@sinhgad.edu ..... Recipient ok

C : DATA

S : 354 Enter Mail, end with " . " on a line by itself

C : Do you like Apple ?

C : What about school ?

C : .

S : 250 message accepted for delivery

C : QUIT

S : 221 sinhgad.edu closing connection

1.7 Multipurpose Internet Mail Extensions (MIME)

- MIME is a supplementary protocol that allows non-ASCII data to be sent throu SMTP.
- MIME defined by IETF to allow transmission of non-ASCII data via e-mail.
- It allows arbitary data to be encoded in ASCII for normal transmission.
- All media types that are sent or received over the world wide web (www) encoded using different MIME types.
- Messages sent using MIME encoding include information that describes the ty of data and the encoding that was used.
- RFC822 specifies the exact format for mail header lines as well as their semar interpretations.
- Fig. 1.7.1 shows the working of MIME.

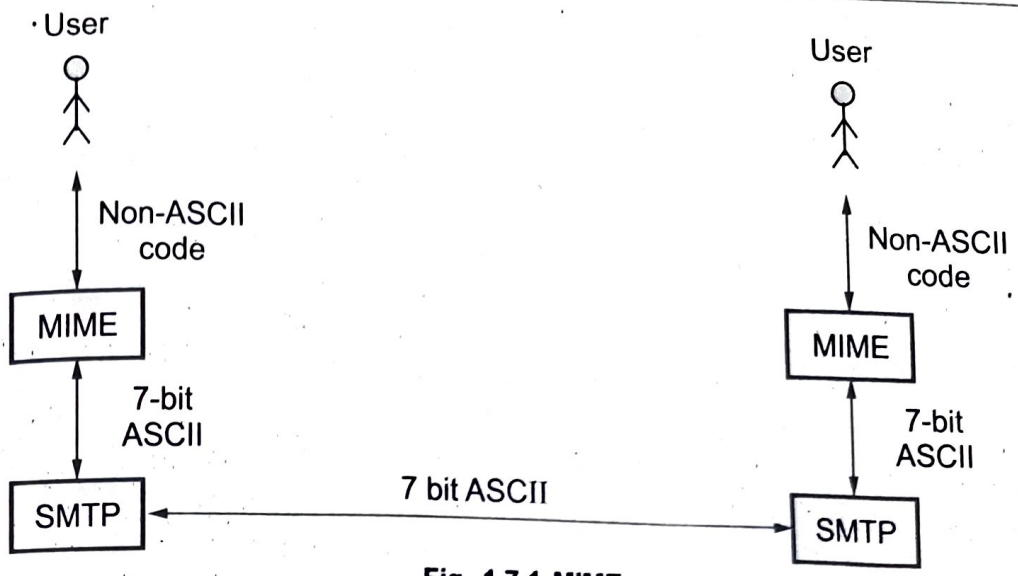

Fig. 1.7.1 MIME

- MIME define five headers.
  1. MIME - Version
  2. Content - Type
  3. Content - Transfer - Encoding
  4. Content - Id
  5. Content - Description

Mail Message Header

- From : iresh@e-mail.com
- TO : rupali@sinhgad.edu
- MIME - Version : 1.0
- Content - Type : image/gif
- Content - Transfer - Encoding : base64

..... data for the image .....

....

....

MIME Types and SubTypes

- Each MIME content - type must contain two identifiers :
  - Content type
  - Content subtype
- There are seven standardized content-types that can appear in a MIME content - type declaration.

Type	Subtype	Description
Text	Plain	Unformatted text.
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.
	Alternative	Parts are different versions of the same message.
Video	MPEG	Video is in MPEG format.
Audio	Basic	Single channel encoding of voice at 8 kHz. (Sound file)
Image	JPEG	Image is in JPEG format.
	GIF	Image is in GIF.
Message	Partial and external body	An entire e-mail message or an external reference to a message.
	Postscript	Adobe postscript.
Application	Octet stream	General binary data.

- Transfer Encoding

his header defines the method to encode the messages into 0 and 1 for transpor
Content-Transfer-Encoding : < Type >
he five types of encoding is listed below.

	Description
	CII characters and short lines.
	racters and short lines.
	h unlimited length lines.
	8-bit ASCII characters.
	ign followed

- After TCP connection established, POP3 progresses three phases :

i) Authorization ii) Transaction iii) Update

- In authorization phase, user agent sends a user name and a password to authenticate the user downloading the mail.
- In transaction phase, the user agent retrieves messages. In this phase, user agent can also mark messages for deletion, remove deletion marks.
- In update phase, it occurs after the client has issued the quit command, ending the POP3 session.
- POP3 has two modes : Delete mode and the keep mode.
- In the delete mode, mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval.
- Fig. 1.8.1 shows downloading using POP3.

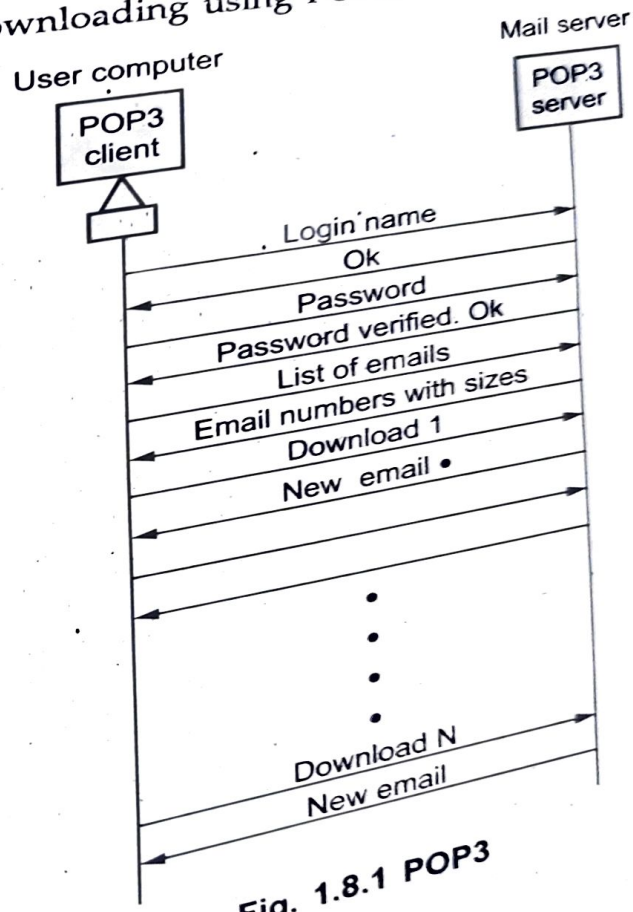

Fig. 1.8.1 POP3

POP3 the user to organize mail on the server, the user cannot have

check the contents of the e-mail before

Status line	HTTP / 1.1 300 ok
General headers	Date : Wed , 8 Oct 2014 13:00:13 GMT
	Connection : close
Entity headers	Server : Apache / 1.3.27
	Accept-range : bytes
	Content-type : text / html
	Content-length : 200
	Last-modified : 2 Oct 2014 13:00:13 GMT
Blank line	
Message body	<html> <head> <title> Welcome to the India <title> <head> <body>

Internet Mail Access Protocol. IMAP4 is more powerful and mo
P is similar to SMTP.

ed to help the user who uses multiple computers.

ot copy e-mail to the user's personal machine because the user ma

nt connects to a server by using TCP.

ts the following modes for accessing e-mail messages :
ie ii) Online mode iii) Disconnected mode

client periodically connects to the server to download e-ma
wnloading, messages are deleted from the server. POP3 support th

ent process e-mail messages on the server. The e-mail messages ar
itself but are processed by an application on the client's end.

e : In this mode, both offline and online modes are supported.

following extra functions :

the e-mail header prior to downloading.
lly download e-mail.
te, delete or rename mailboxes on the mail server.
te a hierarchy of mailboxes in a folder for e-mail storage.
the contents of the e-mail for a specific string of charac

Fig. 1.9.1 shows IMAP state transition diagram.

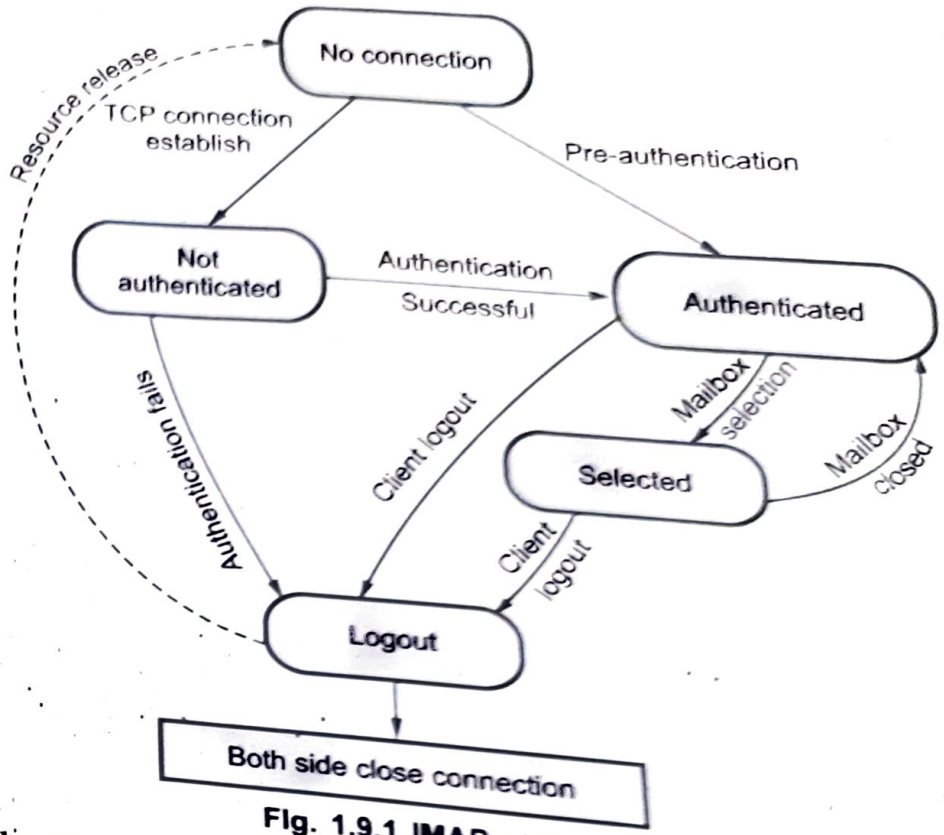

Fig. 1.9.1 IMAP state diagram

1. **Not authenticated :** Client provides authentication information to the server.
2. **Authenticated :** Server verify the information and client is now allowed to perform operations on a mailbox.
3. **Selected :** Client is allowed to access of manipulate individual messages within the mailbox.
4. **Logout :** Client send logout command for closing IMAP session.

1.10 DHCP

- The Bootstrap Protocol (BOOTP) is a static configuration protocol. Each client has a permanent network connection.
- When a client requests its IP address, BOOTP server checks a table that matches the physical address of the client with its IP address. The binding is predefined,
- If the client moves from one network to another then its creates a proble BOOTP cannot handle these situations because the binding and IP addresses is static and fixed in administrator. If there is needs to

- So, to remove the limitations of BOOTP, Dynamic Host Configuration Protocol (DHCP) protocol is used.
- DHCP does not require an administrator to add entry for each connection to the database. DHCP provides a mechanism that allows a computer to join a new network and obtain an IP address without manual intervention. The DHCP work like plug and play networking.
- The DHCP provides static and dynamic address allocation. Static addresses are created manually whereas dynamic addresses are created automatically.
- **Static Address Allocation :** DHCP is backward compatible with BOOTP, which means a computer running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.
- **Dynamic Address Allocation :** DHCP has a pool of available IP addresses. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available IP addresses (unused addresses) and assigns an IP address for a negotiable period of time.
- When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.
- DHCP provides temporary IP addresses for a limited time. The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific time. When the lease is expired, the client must either stop using the IP address or renew the lease. The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

1.10.1 DHCP Message Format

- The format of the DHCP messages is based on the format of BOOTP messages in order to keep backward compatible with BOOTP clients.
- The DHCP message format is given in Fig. 1.10.1.

- Each
information used to p

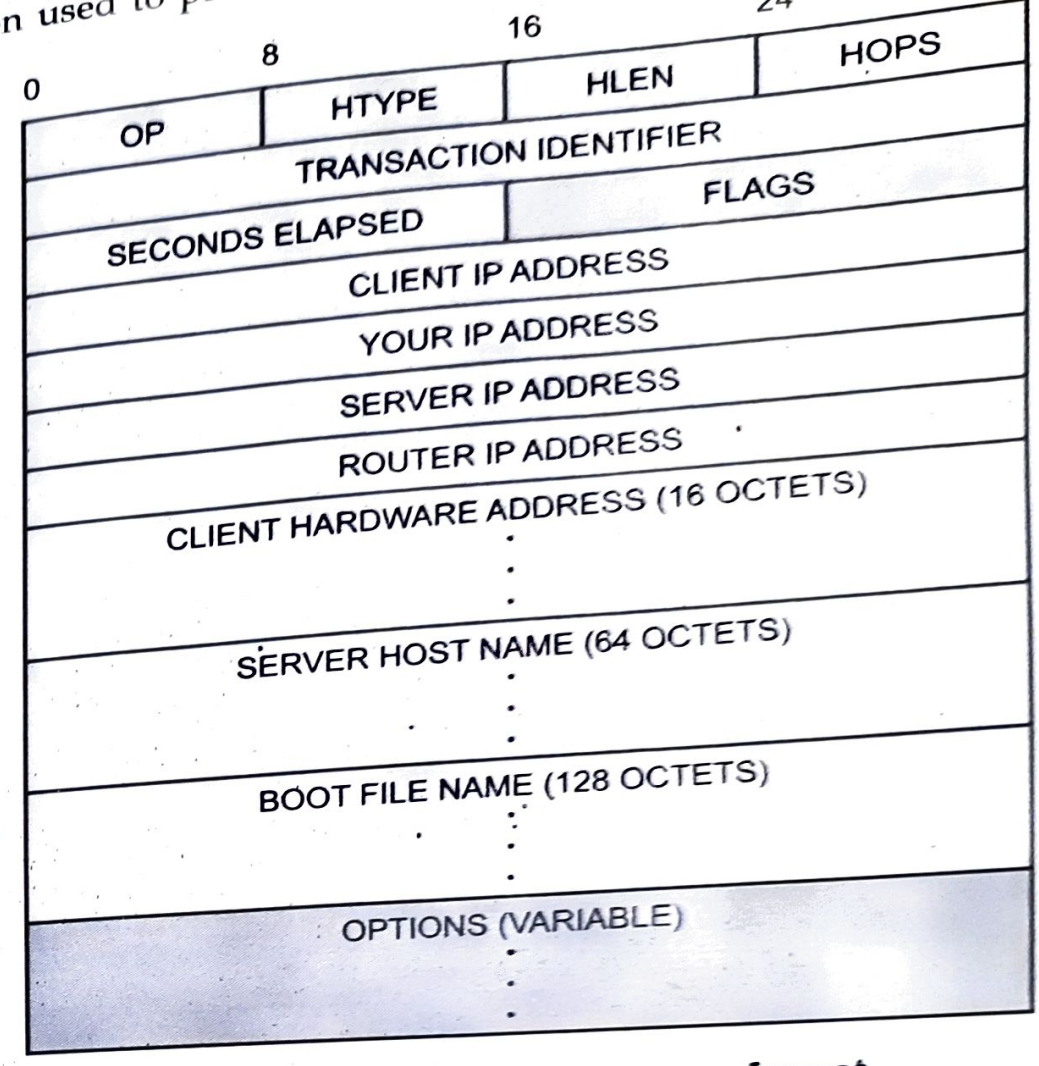

Fig. 1.10.1 The DHCP message format

Field Name	Size (bytes)	Description
OP	1	**Operation Code :** Specifies the general type of message. A value of 1 indicates a request message, while a value of 2 is a reply message.
HTYPE	1	**Hardware Type :** This field specifies the type of hardware used for the local network.
HLEN	1	**Hardware Address Length :** Specifies how long hardware addresses are in this message. For Ethernet or other networks using IEEE 802 MAC addresses, the value is 6.
HOPS	1	**Hops :** Specifies how many servers forwarded the request
TRANSACTION IDENTIFIER	4	**Transaction Identifier :** A 32-bit identification field generated by the client, to allow it to match up the request with replies received from DHCP servers.

SECONDS	2	**Seconds :** It is defined as the number of seconds elapsed since a client began an attempt to acquire or renew a lease.
FLAGS	2	**Flags :** In flag filed, only the leftmost bit is used and the rest of the bits should be set to 0s. A leftmost bit 0 specifies unicast and 1 specifies broadcast.
CLIENT IP ADDRESS	4	**Client IP Address :** The client puts its own current IP address in this field if and only if it has a valid IP address while in the BOUND, RENEWING or REBINDING states; otherwise, it sets the field to 0.
YOUR IP ADDRESS	4	**Your IP Address :** The IP address that the server is assigning to the client if client does not know its address.
SERVER IP ADDRESS	4	**Server IP Address :** It is the address of the server that the client should use for the next step in the bootstrap process, which may or may not be the server sending this reply.
ROUTER IP ADDRESS	4	**Router IP Address :** This field containing the IP address of a router. It is filled by the server in a reply message.
CLIENT HARDWARE ADDRESS	16	**Client Hardware Address :** The hardware address of the client, which is used for identification and communication.
SERVER HOST NAME	64	**Server Name :** The server sending a DHCPOFFER or DHCPACK message may optionally put its name in this field.
BOOT FILE NAME	128	**Boot Filename :** Optionally used by a client to request a particular type of boot file in a DHCPDISCOVER message. Used by a server in a DHCPOFFER to fully specify a boot file directory path and filename.
OPTIONS	Variable	**Options :** This field specifies to encode additional information. To distinguish among various messages that a client uses to discover servers or request an address or that a server uses to acknowledge. The field is used only in a reply message.

1.10.2 Transition States for DHCP

- To provide dynamic address allocation, the DHCP client acts as a state machine that performs transitions from one state to another depending on the messages it receives or sends.
- The Fig. 1.10.2 shows the transition diagram with main states.

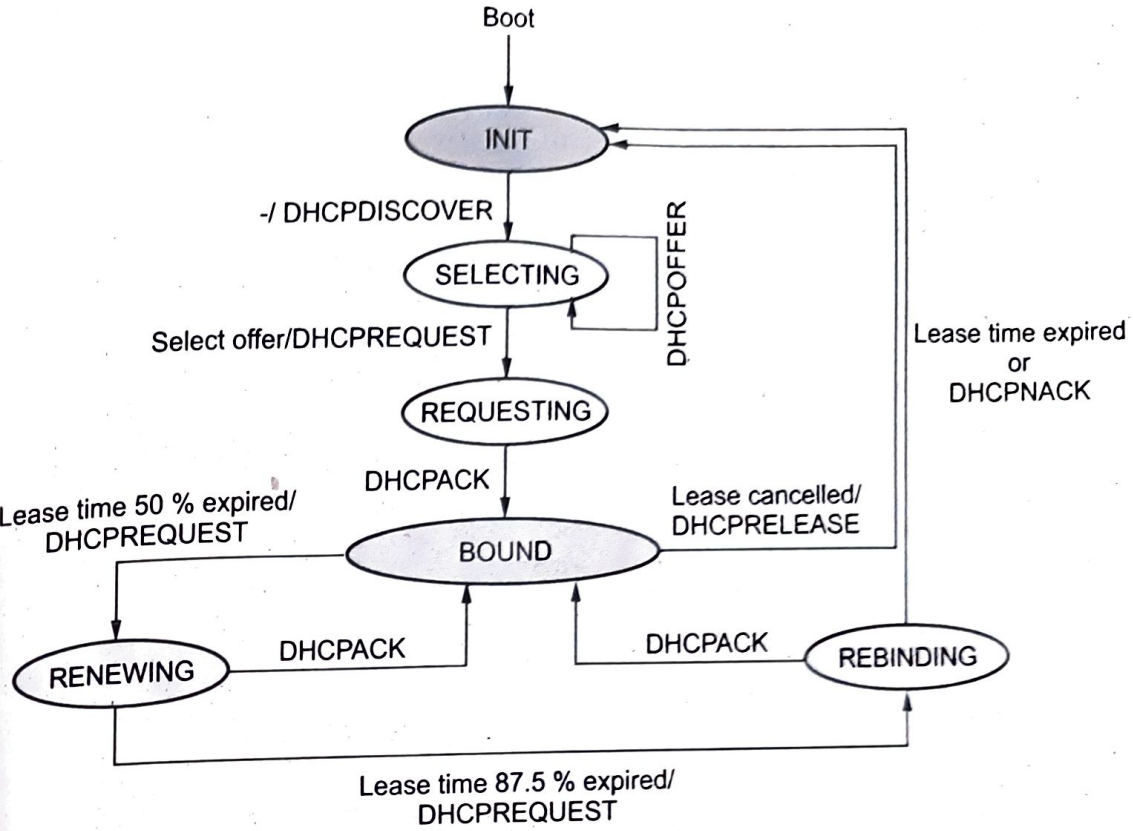

Fig. 1.10.2 DHCP transition diagram

T State :

When the DHCP client first boot, it enters in the INIT state (initializing state). It has no IP and does not even know where a DHCP server may be on the network.

So, DHCP client broadcasts DHCPDISCOVER message on the physical subnet to find DHCP servers. This message is in the form of a UDP packet and usually at least contains the clients MAC address in the OPTIONS section.

LECTING State :

After sending the DHCPDISCOVER message, the client goes to the selecting state.

In this state, the client collects DHCPOFFER responses from DHCP servers. The servers offer an IP address, lease duration (The default is 1 hour) etc.

The client must choose one of the offers and negotiate with the server for a lease. To do so, the client sends a DHCPREQUEST message to the selected server.

It then goes to the requesting state.

QUESTING State :

The client remains in the requesting state until it receives a DHCPACK message from the server that creates the binding between the client physical address and its IP address.

After receipt of the DHCPACK, the client goes to the bound state.

4. BOUND State :

- In this state, the client can use the IP address until it's the lease expires.
- When 50 percent of the lease period is reached, the client sends DHCPREQUEST to ask for renewal.
- It then goes to the renewing state. When in the bound state, the client cancel the lease and go to the initializing state.

5. RENEWING State :

- The client remains in the renewing state until one of two events receive a DHCPACK, which extends the lease agreement.
- In this case, the client resets its timer and goes back to the bound state DHCPACK is not received by the time 87.5 percent of the lease expires, goes to the rebinding state.

6. REBINDING State :

- The client remains in the rebinding state until one of two events happens client receives a DHCPACK, which extends the lease expires and it goes back to the bound state and tries to get a new IP address.
- If the client receives a DHCPNACK, it goes to the initialize state and restarts the process.

Early Release

- If the client no longer needs its IP address before its lease expires, it can send a release message to the server.
- The server acknowledges the release message and returns the IP address to the pool of available addresses.
- The client can then release its IP address and terminate the DHCP process without sending any further messages to the server.

1.11 TELNET

- TELNET (Terminal Network) is a general-purpose client/server application program.
- Client - server model can create a mechanism that allows a user to establish a session on the remote machine and then run its application. This application is known as remote login. Telnet is the example of remote login.
- The purpose of the TELNET protocol is to provide a general, bi-directional, byte oriented communications facility.
- Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other.
- NET was the first TCP/IP application and still is widely used as a terminal ulator. Many application protocols are built upon the TELNET protocol.
- client program running on the user's machine communicates using TELNET protocol with a server program running on the remote machine. The TELNET program perform two important functions :
  - Interacting with the user terminal on the local host
  - Exchanging messages with the TELNET server.

For the connections, TELNET uses the TCP protocol. The TELNET service is offered in the host machine's TCP port 23. The user at the terminal interacts with the local telnet client. The TELNET client acts as a terminal accepting any keystrokes from the keyboard, interpreting them and displaying the output on the screen. The client on the computer makes the TCP connection to the host machine's port 23 where the TELNET server answers. The TELNET server interacts with applications in the host machine and assists in the terminal emulation.

is designed to work on heterogeneous system i.e. two hosts on different tforms, the protocol assume that the two hosts run a Network Virtual Terminal (NVT). The TCP connection is set using two NVT terminals. The NVT is very simple character device with a keyboard and a printer, data typed by the user on

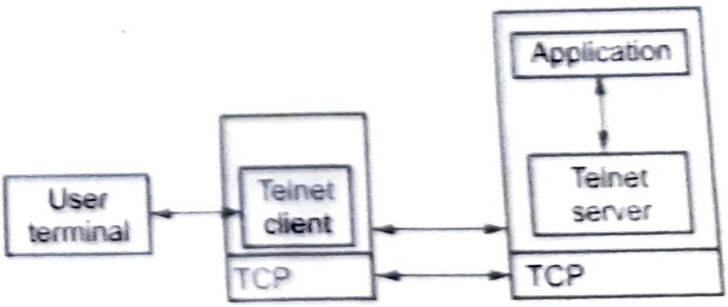

Fig. 1.11.1 TELNET protocol model

4. BOUND State :

- In this state, the client can use the IP address until it's the lease expires.
- When 50 percent of the lease period is reached, the client sends anoth DHCPREQUEST to ask for renewal.
- It then goes to the renewing state. When in the bound state, the client can al cancel the lease and go to the initializing state.

5. RENEWING State :

- The client remains in the renewing state until one of two events happens. It c receive a DHCPACK, which renews the lease agreement.
- In this case, the client resets its timer and goes back to the bound state. Or, if DHCPACK is not received, and 87.5 percent of the lease time expires, the clie goes to the rebinding state.

6. REBINDING State :

- The client remains in the rebinding state until one of three events happens. If th client receives a DHCPNACK or the lease expires, it goes back to the initializin state and tries to get another IP address.
- If the client receives a DHCPACK, it goes to the bound state and resets the timer.

Early Release

- If the client is no longer needed an IP address, then it sends a DHCPRELEAS message to the server.
- The server accepts the request and assigns this IP address to another client waitin for the address.
- A DHCP client that has been assigned an address for a period of time may releas the address before the expiration time. The client may send a DHCPRELEAS message to tell the server that the address is no longer needed. This helps th server to assign the address to another client waiting for it.

Timers

- The three timers generally the client use. When the server allocates an IP addres to the client and does not specify the time out values (lease period), the client us the default value.
- The default value for each timer is shown below :

  Renewal timer : 50 % of lease time

  Rebinding timer : 87.5 % of lease time

  Expiration timer : 100 % of lease time

1.11 TELNET

- TELNET (Terminal Network) is a general-purpose client/server application program.
- Client - server model can create a mechanism that allows a user to establish a session on the remote machine and then run its application. This application is known as remote login. Telnet is the example of remote login.
- The purpose of the TELNET protocol is to provide a general, bi-directional, byte oriented communications facility.
- Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other.
- TELNET was the first TCP/IP application and still is widely used as a terminal emulator. Many application protocols are built upon the TELNET protocol.
- A client program running on the user's machine communicates using TELNET protocol with a server program running on the remote machine. The TELNET client program perform two important functions :
  - Interacting with the user terminal on the local host
  - Exchanging messages with the TELNET server.
- For the connections, TELNET uses the TCP protocol. The TELNET service is offered in the host machine's TCP port 23. The user at the terminal interacts with the local telnet client. The TELNET client acts as a terminal accepting any keystrokes from the keyboard, interpreting them and displaying the output on the screen. The client on the computer makes the TCP connection to the host machine's port 23 where the TELNET server answers. The TELNET server interacts with applications in the host machine and assists in the terminal emulation.
- Telnet is designed to work on heterogeneous system i.e. two hosts on different platforms, the protocol assume that the two hosts run a Network Virtual Terminal (NVT). The TCP connection is set using two NVT terminals. The NVT is very simple character device with a keyboard and a printer, data typed by the user on

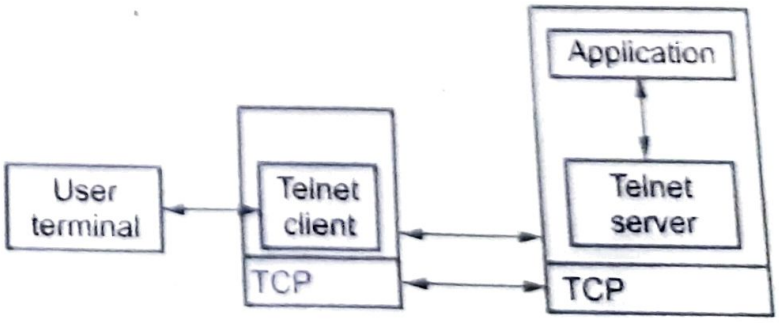

Fig. 1.11.1 TELNET protocol model

options and these options can be negotiated through which th
protocol inside the TELNET. The following table shows some commo
options.

Sr. No.	Code	Option	Meaning
1	0	Binary	Interpret as 8 bit binary transmission
2	1	Echo	Echo the data received on one side to the other
3	3	Suppress go ahead	Suppress go-ahead signals after data
4	5	Status	Request the status of TELNET
5	6	Timing mark	Define the timing marks
6	24	Terminal type	Set the terminal type
7	32	Terminal speed	Set the terminal speed
8	34	Line mode	Change to line mode

Table 1.11.1 Option

Option Negotiation :

- To use any of the options mentioned in above table, first requires opti
negotiation between the client and the server. The following table shows so
negotiation options.

Sr. No.	Sender	Receiver	Meaning
1.	WILL →	← DO	Sender wants to active a option, and receiver agrees
2.	WILL →	← DON'T	Sender wants to active

	← WONT	Sender wants receiver to active a option, and receiver agrees
		Sender wants receiver to active a option, and receiver refuses

Table 1.11.2 Option negotiation

- TELNET has the following properties.
- Client programs are built to use the standard client/server interface without knowing the details of server programs.
- A client and server can negotiate above data format options.
- Once a connection is established through Telnet, both ends of the connection are treated symmetrically.

Mode of Operation in Telnet :

- **Default Mode :** In this mode, the echoing is done by the client.
- **Character Mode :** In this mode, each character typed is sent by the client to the server. The server echoes the character back to the client screen and it can be delayed if the transmission time is long/slow. It also creates overhead (traffic) for the network.
- **Line Mode :** In this mode, Line editing, Line erasing, character erasing is done by client.

1.11.1 TELNET Commands

- Typical for the terminals directly connected to a computer are that the keystrokes by the user are immediately interpreted by the computer's operating system. For the purpose certain keystroke combinations were invented. For example by pressing ctrl+'z', the processes were suspended or ctrl+'c' was used for killing current process. The TELNET cannot transmit such codes as they are since the codes are commands containing two keystrokes and do not map to the 7-bit ASCII chart used in the NVT. This requires that the client has to translate the terminal's control codes to the TELNET commands and transmit the commands to the server host's operating system.
- Some of the TELNET commands are presented in the following table.

Sr. No.	Name	Code	Meaning
1			

3	ABORT	238	Abort process
4	EOR	239	End of record
5	NOP	241	No operation
6	BRK	243	Break
7	Go Ahead	249	The GA signal
8	WILL	251	Option negotiation
9	WONT	252	Option negotiation
10	DO	253	Option negotiation
11	DONT	254	Option negotiation
12	IAC	255	Interpret as Command

Table 1.11.3

1.11.2 Comparison between FTP and TELNET

Sr. No.	FTP	TELNET
1.	FTP is a two-way system - it can be used to copy or move files from a server to a client computer as well as upload or transfer files from a client to a server	TELNET is two –way system (with authorization) it can be used to copy moves files from other computer
2.	FTP systems generally encode and transmit their data in binary sets which allow for faster data transfer	TELNET while connection client-serve communication is non-coded
3.	Commands : ASCII, Binary, Open, Del and Get	Commands : Open, Close, Display, Status and Quit

1.12 Short Answered Questions

Q.1 What are the four groups of HTTP headers ?

Ans. : The four groups of HTTP headers are : General headers, Entity headers, Reques headers and Response headers.

Q.2 What are the four main properties of HTTP ?

Ans. : The four main properties of HTTP are :

1. Global uniform resource identifier.
2. Request - response exchange.
3. Statelessness.
4. Resource metadata.

Q.3 Mention the types of HTTP messages.

Ans. : Types of HTTP messages : Request and Response

Q.4 What are the transmission modes of FTP ?

Ans. : Transmission modes of FTP are :

1. Stream mode : Default mode and data is delivered from FTP to TCP as a continuous stream of data.
2. Block mode : Data is delivered from FTP to TCP in terms of blocks. Each data block follows the three byte header.
3. Compressed mode : File is compressed before transmitting if size is big. Run length encoding method is used for compression.

Q.5 Mention the application of FTP.

Ans. :

1. Used for remote login and data transfer.
2. FTP provides good security.
3. It is often used to upload web pages and other documents from a private development machine to a public web - hosting server.

Q.6 What are the basic functions of e - mail ?

Ans. : Basic functions of e - mail are : Composition, Transfer, Reporting, Displaying, Disposition.

Q.7 Why email security is necessary ?

Ans. : Email security is the process of using email encryption to send messages that can only be opened by the intended recipient. Sending a message without secure email encryption is similar to dropping a postcard in the mail - it can be read by almost anyone handling the postcard during its journey from sender to receiver. Secure email encryption protects both online data and customers' sensitive information.

Q.8 When web pages are sent out, they are prefixed by MIME headers. Why ?

Ans. : The MIME headers tell the browser what type of file is contained on the Web page and also what type of helper application or plug - in needs to be used to display the content.

Q.9 State the difference between SMTP and MIME.

Ans. :

Sr. No.	SMTP	MIME
1.	SMTP is protocol used to exchange messages between mail servers.	MIME expands the messaging abilities of SMTP and supports all formats.
2.	SMTP is the most widely used internet application.	MIME allows multimedia and other non - textual formats to be handled reliably throughout the message transport process.

Q.10 What is DNS ?

Ans. : DNS is a client/server application that identifies each host on the internet with a unique user friendly name.

Q.11 What is the Domain Name System responsible for ?

Ans. : The Domain Name System converts domain names (of the form "www.vtubooks.com") into IP numbers.

Q.12 Why do we need a Domain Name System ?

Ans : IP numbers uniquely identify hosts on the internet; however they are difficult t remember. We therefore need a more memorable way of identifying hosts Furthermore, since multiple domains may be hosted by a single computer we need way of mapping multiple domains to a single host. Finally, since domains may b hosted on a number of different machines over a period of time we need a method fo changing the IP number representing a host without having to change the informatio people use to access that host (that is the domain name remains constant but the I number may change).

Q.13 List the two types of DNS message.

Ans. : DNS messages are : Query and Response. The query message consists of th header and the question records. The response message consists of a header, questio record, answer record, authoritative record and additional record.

Q.14 What do you mean by TELNET ?

Ans. : TELNET is a client/server application that allows a user to log on to a remo machine giving the user access to the remote system.

TELNET is used to connect remote computers and issue commands on those compute

1.13 Multiple Choice Questions

Q.1 HTTP stands for _____.

a Hyper Text Transfer Packet
b Host Text Transfer Protocol
c Hyper Text Transparent Protocol
d Hyper Text Transfer Protocol

Q.2 The DNS protocol runs over UDP and uses port _____.

a 25
b 53
c 80
d 110

Q.3 The _____ is the universal language of the web.

a XML
b Java
c HTML
d Python

Q.4 SMTP uses a TCP socket on port _____ to transfer e-mail reliably from client to server

a 25
b 53
c 80
d 110

Q.5 Basic functions of e-mail _____.

a composition
b transfer
c reporting
d all of the above

Q.6 The _____ converts domain names into IP numbers.

a world wide web
b HTTP
c domain Name system
d email

Q.7 TELNET is a client - server protocol, based on TCP and clients generally connect to port _____ on the host providing the service.

a 23
b 25
c 53
d 80

Q.8 The local name server is responsible for continuing the resolution by issuing further queries. This mechanism is called an _____ query.

a recursive
b iterative
c distributed
d none