DNS: The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources. It helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names.

For example, if someone types "example.com" into a web browser, a server behind the scenes maps that name to the corresponding IP address

IP Address: IP address is a unique logical address assigned to a machine over the network. An IP address exhibits the following properties:

- IP address is the unique address assigned to each host present on Internet.
- IP address is 32 bits (4 bytes) long.
- IP address consists of two components: **network component** and **host component**.
- Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

Uniform Resource Locator (URL): Uniform Resource Locator (URL) refers to a web address which uniquely identifies a document over the internet. This document can be a web page, image, audio, video or anything else present on the web.

For example, **www.tutorialspoint.com/internet_technology/index.html** is an URL to the index.html which is stored on tutorialspoint web server under internet technology directory.

URL Types: There are two types of URL:

- Absolute URL
- Relative URL

1. **Absolute URL:** Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example http:// www.tutorialspoint.com / internet_technology /index.htm. where:

- **http** is the protocol.
- **tutorialspoint.com** is the server name.
- **index.htm** is the file name.
-

2. **Relative URL:** Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL**.** Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

For example: To link an image on tutorialspoint.com/internet_technology/internet_referemce_models, we can use the relative URL which can take the form like **/internet_technologies/internet-osi_model.jpg.**

Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

Domain Names: Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as com, edu, gov, net etc, while some country level domain names such as au, in, za, us etc.

The following table shows the **Generic** Top-Level Domain names:

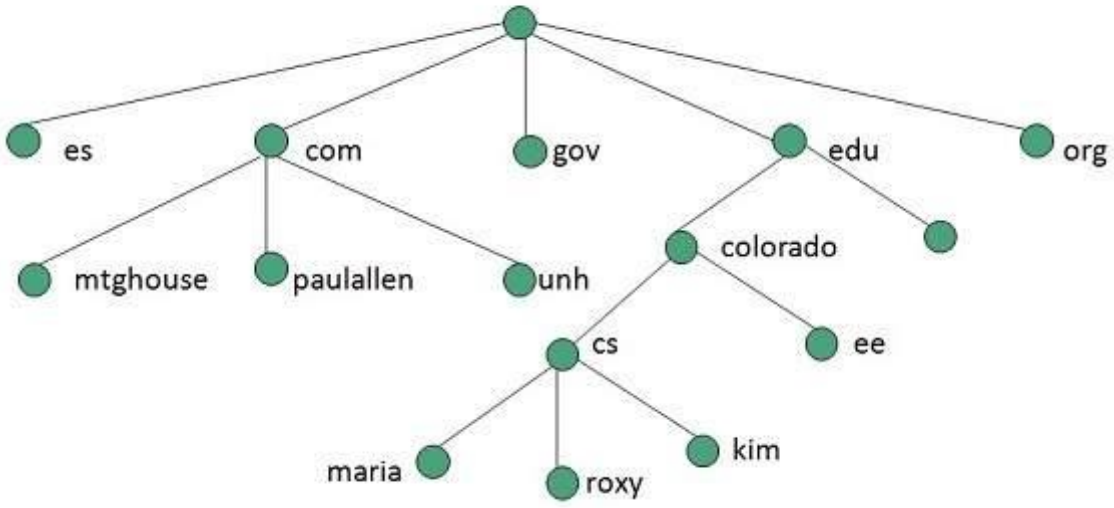
Domain Name	Meaning
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

The following table shows the **Country top-level** domain names:

Domain Name	Meaning
au	Australia
In	India
Cl	Chile
Fr	France
Us	United States
Za	South Africa

Uk	United Kingdom
Jp	Japan
Es	Spain
De	Germany
Ca	Canada
Ee	Estonia
Hk	Hong Kong

Domain Name Space: The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

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 (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name: Each node in the tree has a domain name. A full **domain name** is a sequence of labels separated by dots (.). The domain names are always read from the node up to the

root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

Fully Qualified Domain Name (FQDN) If a label is terminated by a null string, it is called a **fully qualified domain name (FQDN).** An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host. For example, the domain name is the FQDN of a computer named *challenger* installed at the Advanced Technology Center (ATC) at De Anza College. A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

challenger.atc.fhda.edu.

Partially Qualified Domain Name (PQDN) If a label is not terminated by a null string, it is called a **partially qualified domain name (PQDN).** A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the *suffix*, to create an FQDN. For example, if a user at the *fhda.edu.* site wants to get the IP address of the challenger computer, he or she can define the partial name

challenger

The DNS client adds the suffix *atc.fhda.edu*. before passing the address to the DNS server.
The DNS client normally holds a list of suffixes. The following can be the list of suffixes at De Anza College. The null suffix defines nothing. This suffix is added when the user defines an FQDN.

atc.fhda.edu fhda.edu null

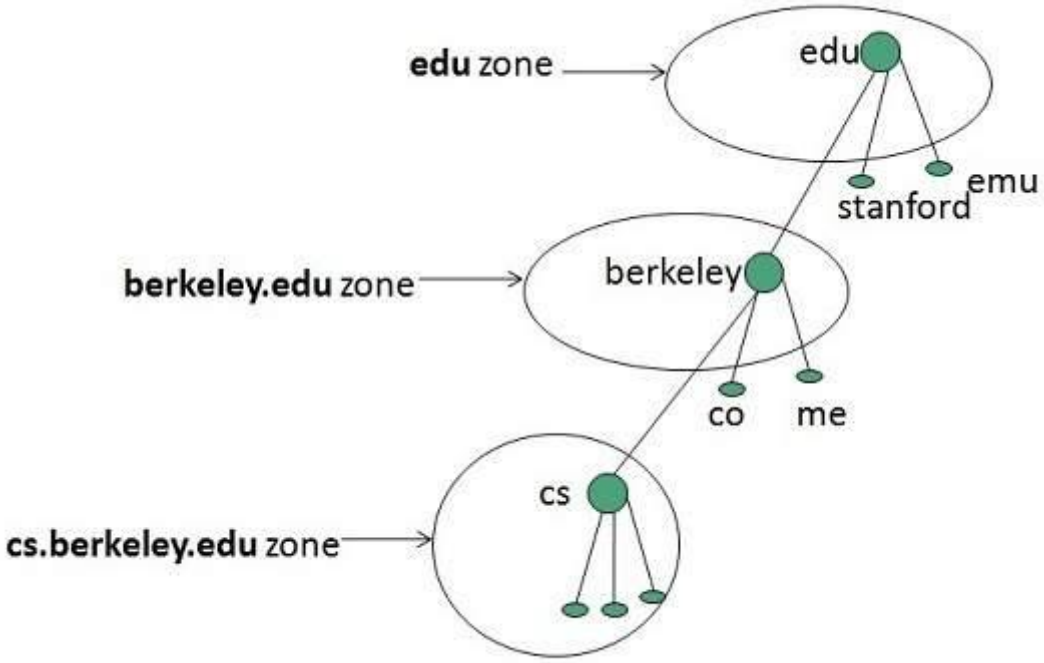
Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

Zone is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone.



If the domain is not further divided into sub domains then domain and zone refers to the same thing.

The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

Types of Name Servers

Following are the three categories of Name Servers that manages the entire Domain Name System:

- Root Server
- Primary Server
- Secondary Server

Root Server: Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server

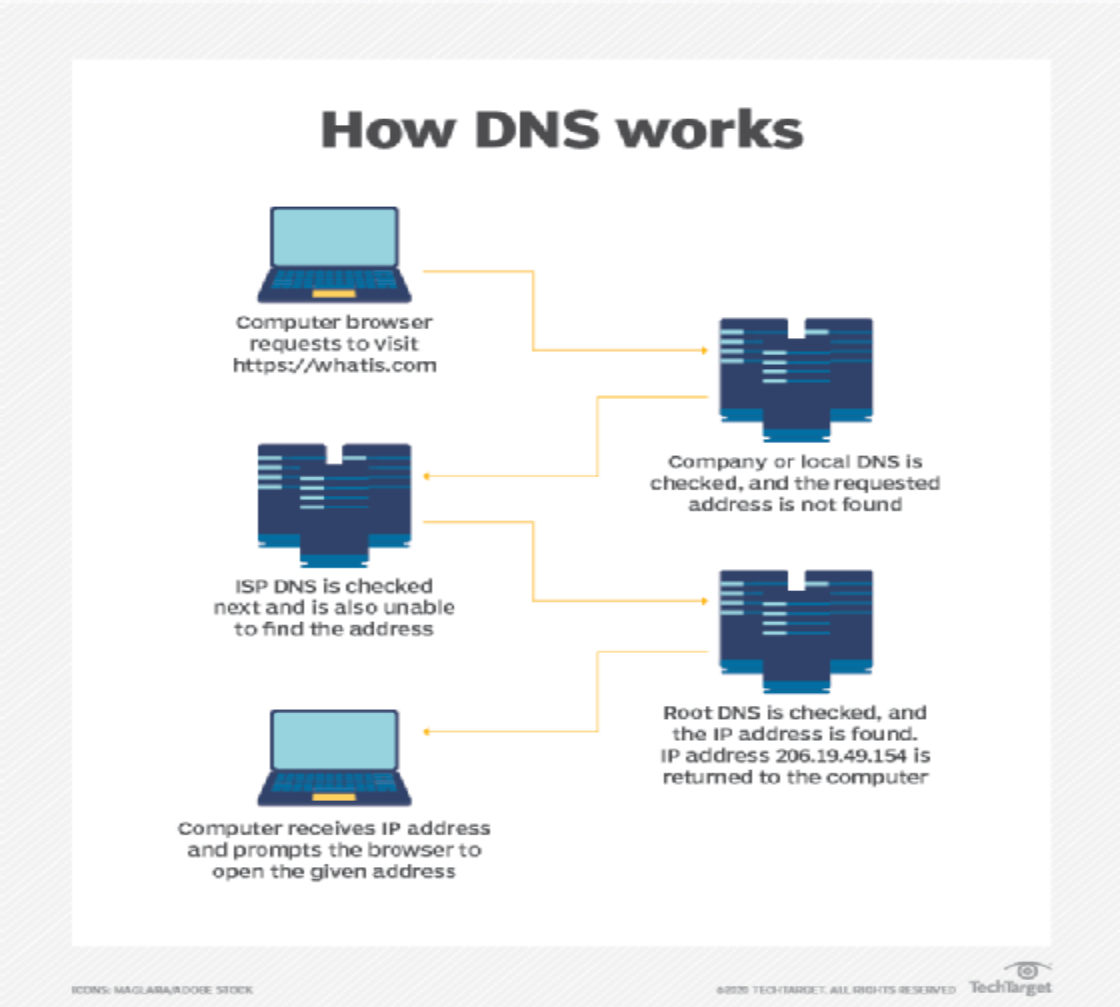
Primary Servers: Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

Secondary Server: Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

DNS Resolution (How DNS works): DNS servers convert URLs and domain names into IP addresses that computers can understand and use. They translate what a user types into a browser into something the machine can use to find a webpage. This process of translation and lookup is called *DNS resolution*.
The basic process of a DNS resolution follows these steps:

1. The user enters a web address or domain name into a browser.

2. The browser sends a message, called a *recursive DNS query*, to the network to find out which IP or network address the domain corresponds to.
3. The query goes to a recursive DNS server, which is also called a *recursive resolver*, and is usually managed by the internet service provider (ISP). If the recursive resolver has the address, it will return the address to the user, and the webpage will load.
4. If the recursive DNS server does not have an answer, it will query a series of other servers in the following order: DNS root name servers, top-level domain (TLD) name servers and authoritative name servers.
5. The three server types work together and continue redirecting until they retrieve a DNS record that contains the queried IP address. It sends this information to the recursive DNS server, and the webpage the user is looking for loads. DNS root name servers and TLD servers primarily redirect queries and rarely provide the resolution themselves.
6. The recursive server stores, or caches, the A record for the domain name, which contains the IP address. The next time it receives a request for that domain name, it can respond directly to the user instead of querying other servers.
7. If the query reaches the authoritative server and it cannot find the information, it returns an error message.



DNS servers talk to each other to answer a query from a client. Some DNS servers will have the necessary information cached and relay that back to the client so they can get online.

Types of DNS queries

The following types of DNS queries are the main ones that take place at different points in the DNS resolution:

- **Recursive DNS queries** are those that take place between the recursive server and the client. The answer provided is either the full name resolution or an error message saying that the name cannot be found. Recursive queries end in either the answer or an error.
- **Iterative DNS queries** take place between the recursive resolver, which is a local DNS server, and the nonlocal name servers, like the root, TLD and authoritative name servers. Iterative queries do not demand a name resolution; the name servers may instead respond with a referral. The root server refers the recursive server to the TLD, which refers it to an authoritative server. The authoritative server provides the domain name to the recursive server if it has it. Iterative queries resolve in either an answer or a referral.
- **Nonrecursive queries** are those for which the recursive resolver already knows where to get the answer. The answer is either cached on the recursive server or the recursive server knows to skip the root and TLD servers and go directly to a specific authoritative server. It is nonrecursive because there is no need -- and, therefore, no request -- for any more queries. Nonrecursive queries resolve in the answer. If a recursive resolver has cached an IP address from a previous session and serves that address upon the next request, that is considered a nonrecursive query.

DNS MESSAGES: DNS has two types of messages: query and response. Both types have the same format. The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records

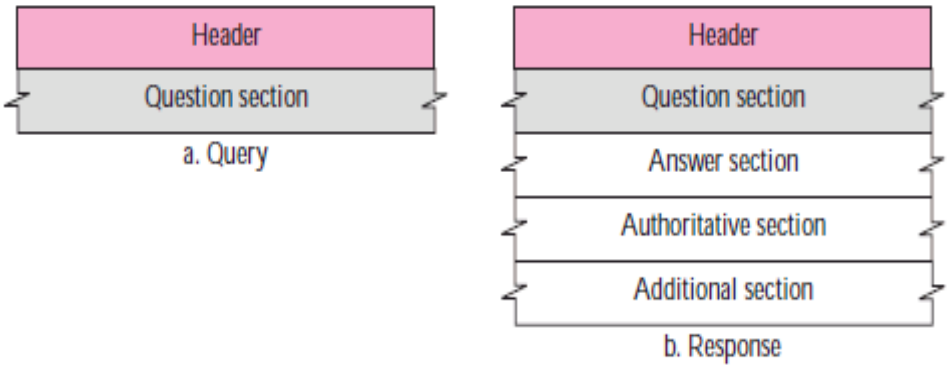
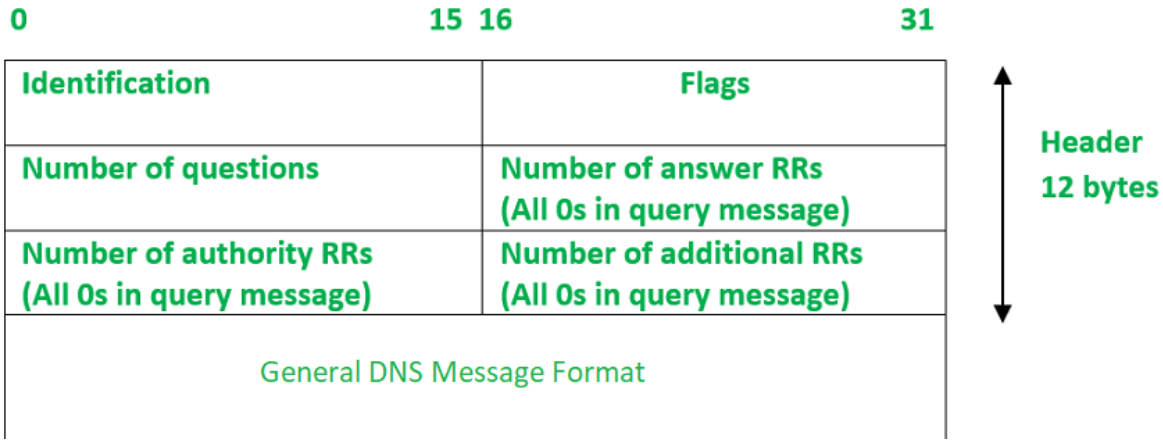


Fig4: Query and response messages

Header: Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes and its format is shown in Figure



The header fields are as follows:

❑ **Identification.** This is a 16-bit field used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response.

❑ **Flags.** This is a 16-bit field consisting of the subfields shown in Figure 19.16.

QR	Opcode	AA	TC	RD	RA	zero	rCode
1	4	1	1	1	1	3	4

A brief description of each flag subfield follows.

a. QR (query/response). This is a 1-bit subfield that defines the type of message. If it is 0, the message is a query. If it is 1, the message is a response.

b. OpCode. This is a 4-bit subfield that defines the type of query or response (0 if standard, 1 if inverse, and 2 if a server status request).

c. AA (authoritative answer). This is a 1-bit subfield. When it is set (value of 1) it means that the name server is an authoritative server. It is used only in a response message.

d. TC (truncated). This is a 1-bit subfield. When it is set (value of 1), it means that the response was more than 512 bytes and truncated to 512. It is used when DNS uses the services of UDP (see Section 19.8 on Encapsulation).

e. RD (recursion desired). This is a 1-bit subfield. When it is set (value of 1) it means the client desires a recursive answer. It is set in the query message and repeated in the response message.

f. RA (recursion available). This is a 1-bit subfield. When it is set in the response, it means that a recursive response is available. It is set only in the response message

g. **Reserved.** This is a 3-bit subfield set to 000.

h. rCode. This is a 4-bit field that shows the status of the error in the response. Of course, only an authoritative server can make such a judgment.

❑ **Number of question records.** This is a 16-bit field containing the number of queries in the question section of the message.

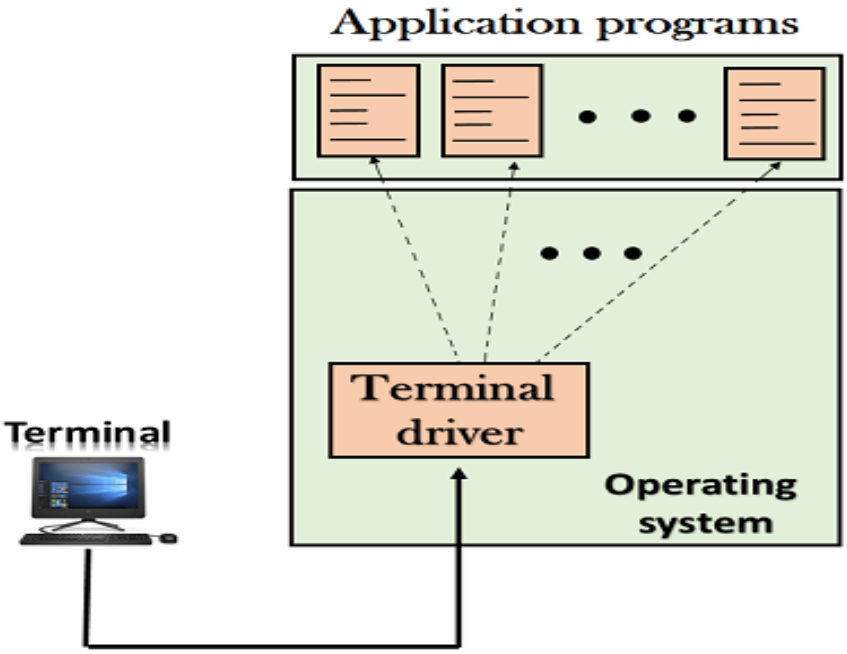
❑ **Number of answer records.** This is a 16-bit field containing the number of answer records in the answer section of the response message. Its value is zero in the query message.
❑ **Number of authoritative records.** This is a 16-bit field containing the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.
❑ **Number of additional records.** This is a 16-bit field containing the number of additional records in the additional section of a response message. Its value is zero in the query message.

Telnet: Telnet is an abbreviation for **Terminal Network**. Computer which starts connection known as the **local computer**. Computer which is being connected to i.e. which accepts the connection known as **remote computer**.

For example, users want to run different application programs at the remote site and transfer a result to the local site. Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side. During telnet operation whatever that is being performed on the remote computer will be displayed by local computer. Telnet operates on client/server principle. Local computer uses telnet client program and the remote computers uses telnet server program.

There are two types of login:

1) Local Login:

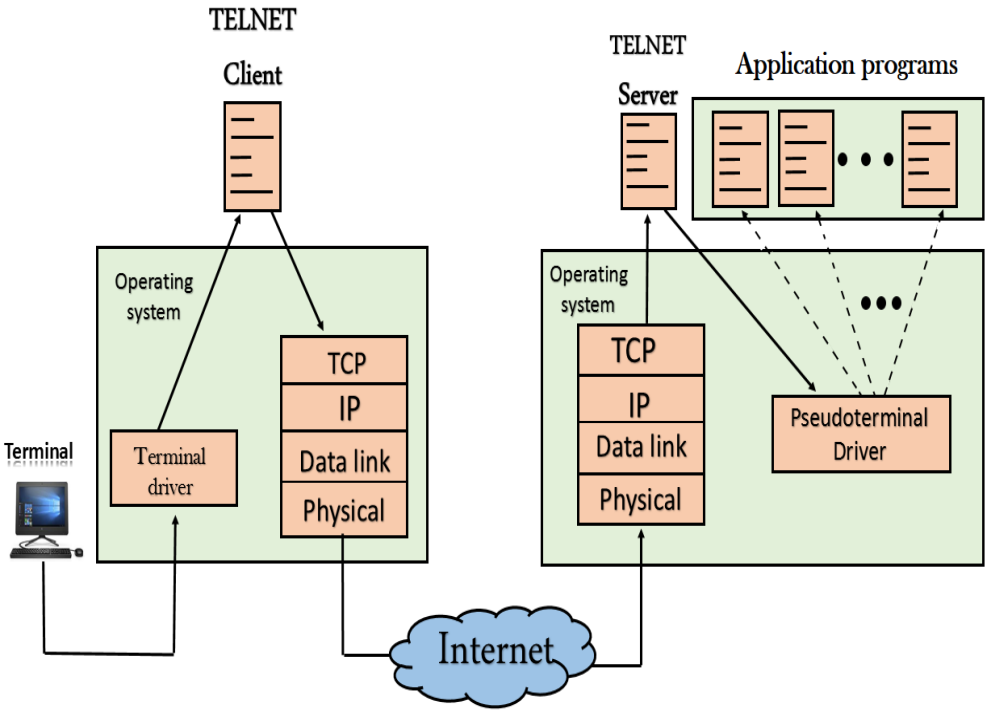


- When a user logs into a local computer, then it is known as local login.
- When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes

these characters to the operating system which in turn, invokes the desired application program.

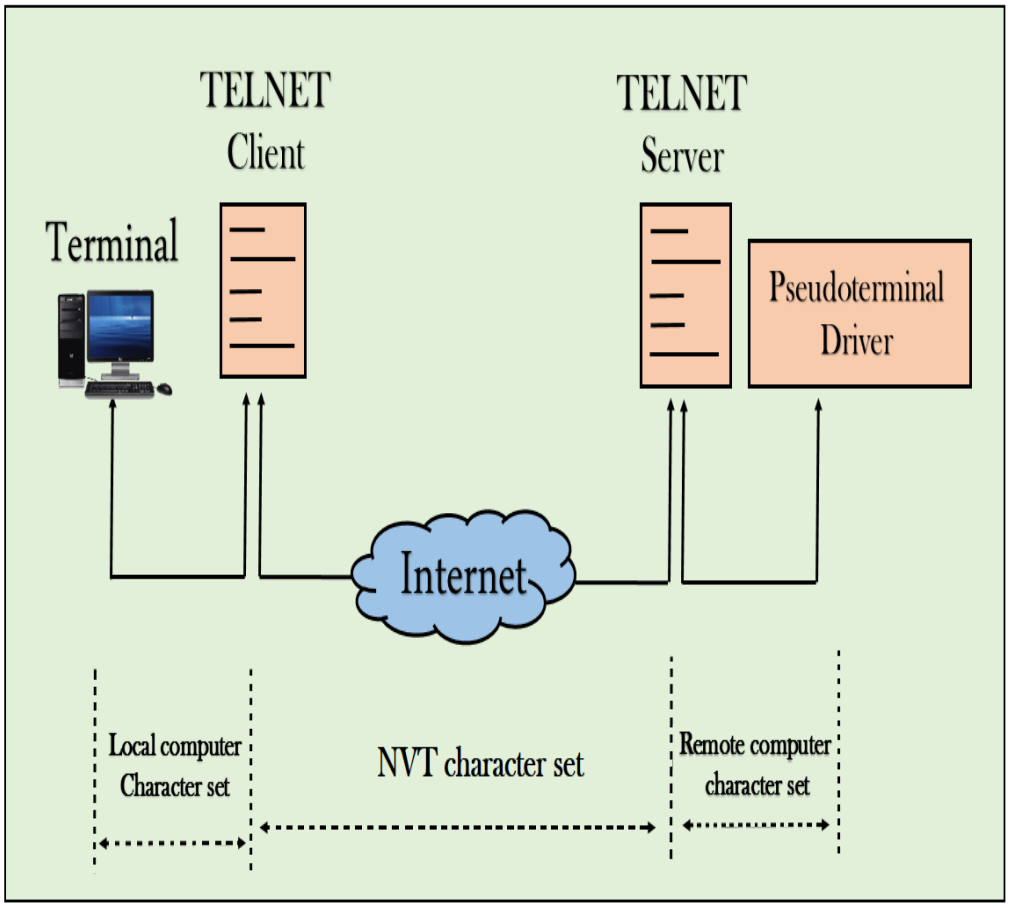
- However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters has special meanings such as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.

2) Remote login



- When the user wants to access an application program on a remote computer, then the user must perform remote login. The commands in NVT forms are transmitted to the TCP/IP at the remote machine. Here, the characters are delivered to the operating system and then pass to the TELNET server. The TELNET server transforms the characters which can be understandable by a remote computer. However, the characters cannot be directly passed to the operating system as a remote operating system does not receive the characters from the TELNET server. Therefore it requires some piece of software that can accept the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

Network Virtual Terminal (NVT)



- The network virtual terminal is an interface that defines how data and commands are sent across the network.
- In today's world, systems are heterogeneous. For example, the operating system accepts a special combination of characters such as end-of-file token running a DOS operating system *ctrl+z* while the token running a UNIX operating system is *ctrl+d*.
- TELNET solves this issue by defining a universal interface known as network virtual interface.
- The TELNET client translates the characters that come from the local terminal into NVT form and then delivers them to the network. The Telnet server then translates the data from NVT form into a form which can be understandable by a remote computer

Modes of Operation : Most telnet implementation operates in one of the following **three modes :**

1. **Default Mode :**

- If there is no other modes are invoked then this mode is used.
- Echoing is performed in this mode by client.

- In this mode, user types a character and client echoes the character on the screen but it does not send it until whole line is completed.

2. Character Mode :

- Each character typed in this mode is sent by client to server.
- Server in this type of mode is normally echoes character back to be displayed on the client's screen.

3. Line Mode :

- Line editing like echoing, character erasing etc. is done from the client side.
- Client will send the whole line to the server.

FTP: File Transfer Protocol (FTP) is an application layer protocol provided by TCP/IP and used for transmitting the files from one host to another. It runs on the top of TCP, like HTTP. It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet. It is also used for downloading the files to computer from other servers.

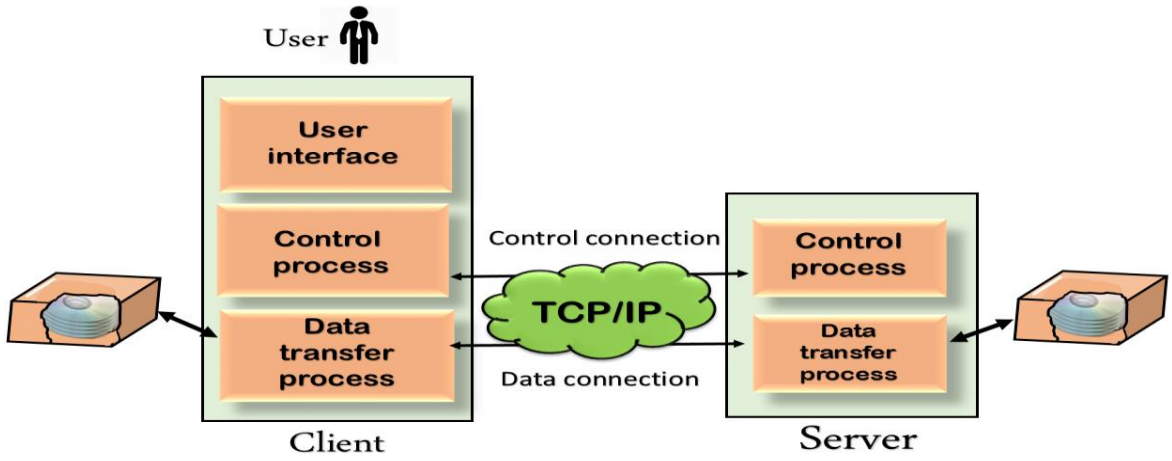
Objectives of FTP:

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Why FTP?

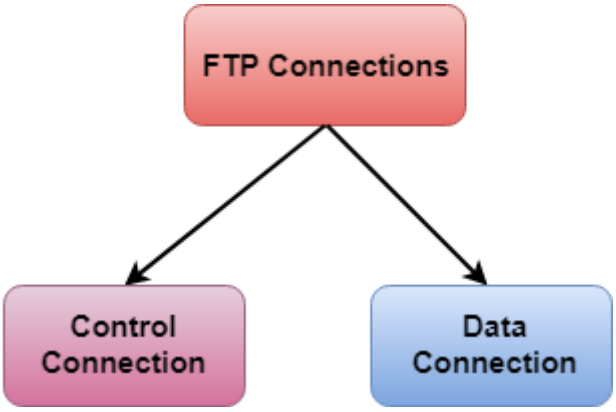
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP:



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP: The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.



- **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.
- **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

FTP Session: When an FTP session is started between a client and a server, the client initiates a control TCP connection with the server-side. The client sends control information over this. When the server receives this, it initiates a data connection to the client-side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session. As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

File Transfer: File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember that file transfer in FTP means one of three things:

❑ A file is to be copied from the server to the client (download). This is called *retrieving a file*. It is done under the supervision of the RETR command.

❑ A file is to be copied from the client to the server (upload). This is called *storing a file*. It is done under the supervision of the STOR command.

❑ A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.

Data Structures: FTP allows three types of data structures :

1. **File Structure –** In file structure, there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure –** In record structure, the file is made up of sequential records.
3. **Page Structure –** In page structure, the file is made up of independent indexed pages.

File Type: FTP can transfer one of the following file types across the data connection:

❑ **ASCII file.** This is the default format for transferring text files. Each character is encoded using NVT ASCII. The sender transforms the file from its own representation into NVT ASCII characters and the receiver transforms the NVT ASCII characters to its own representation.

❑ **EBCDIC file.** If one or both ends of the connection use EBCDIC encoding, the file can be transferred using EBCDIC encoding.

❑ **Image file.** This is the default format for transferring binary files. The file is sent as continuous streams of bits without any interpretation or encoding. This is mostly used to transfer binary files such as compiled programs.

Transmission Mode: FTP can transfer a file across the data connection using one of the following three transmission modes:

❑ **Stream mode.** This is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes. TCP is responsible for chopping data into segments of appropriate size. If the data is simply a stream of bytes (file structure), no end-office is needed. End-of-file in this case is the closing of the data connection by the
Sender. If the data are divided into records (record structure), each record will have a 1-byte end-of-record (EOR) character and the end of the file will have a 1-byte end-of-file (EOF) character.

❑ **Block mode.** Data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header. The first byte is called the *block descriptor;* the next two bytes define the size of the block in bytes.

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

FTP Commands

Command Processing: FTP uses the control connection to establish a communication between the client control process and the server control process. During this communication, the commands are sent from the client to the server and the responses are sent from the server to the client

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. We can roughly divide the commands into six groups:

1. Access commands,
2. File management commands,
3. Data formatting commands,
4. Port defining commands,
5. File transferring commands,
6. Miscellaneous commands.

Some of the FTP commands are:

USER – This command sends the user identification to the server.
PASS – This command sends the user password to the server.
CWD – This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information.
RMD – This command causes the directory specified in the path name to be removed as a directory.
MKD – This command causes the directory specified in the pathname to be created as a directory.
PWD – This command causes the name of the current working directory to be returned in the reply.
RETR – This command causes the remote host to initiate a data connection and to send the requested file over the data connection.
STOR – This command causes to store of a file into the current directory of the remote host.
LIST – Sends a request to display the list of all the files present in the directory.
ABOR – This command tells the server to abort the previous FTP service command and any associated transfer of data.
QUIT – This command terminates a USER and if file transfer is not in progress, the server closes the control connection.

FTP Replies – Some of the FTP replies are:
200 Command okay.
530 Not logged in.
331 User name okay, need a password.
225 Data connection open; no transfer in progress.
221 Service closing control connection.
551 Requested actions aborted: page type unknown.
502 Command not implemented.
503 Bad sequence of commands.
504 Command not implemented for that parameter.

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

Security for FTP: The FTP protocol was designed when the 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:

- **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest ways to transfer the files from one computer to another computer.
- **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
- **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.
- **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:

- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provide encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.

- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

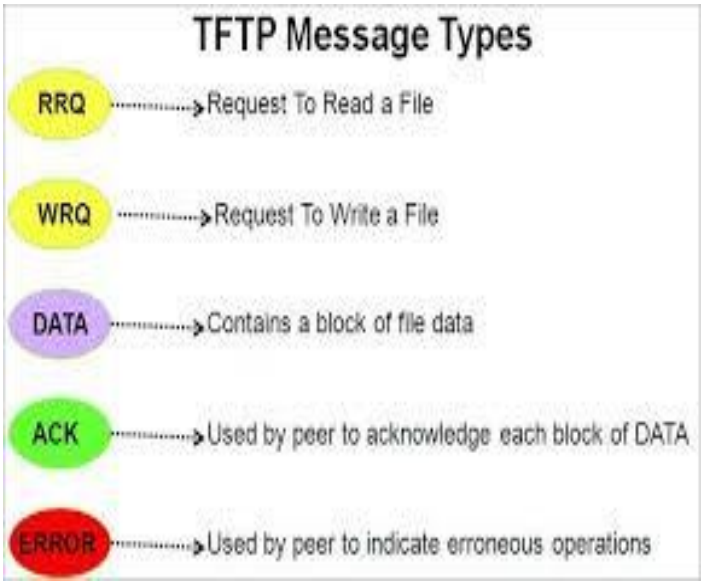
TFTP: Trivial File Transfer Protocol (TFTP) is designed for these types of file transfer. It is so simple that the software package can fit into the read-only memory of a diskless workstation. It can be used at bootstrap time. The reason that it fits on ROM is that it requires only basic IP and UDP. However, there is no security for TFTP. TFTP can read or write a file for the client. *Reading* means copying a file from the server site to the client site. *Writing* means copying a file from the client site to the server site.

The benefit of using TFTP is that it enables bootstrapping code to use the similar underlying TCP/IP protocols that the operating framework uses once it starts execution. Thus it is the possibility for a device to bootstrap from a server on another physical network.

Features of TFTP: The main features of TFTP are as follows**:**

- TFTP is based on the client-server principle and uses well-known UDP port number 69 for the TFTP server.
- TFTP is an unsecured protocol and does not support authentication.
- TFTP incorporates idle − RQ (stop and wait) error recovery mechanism.
- Every TFTP data unit bears a sequence number.
- Each data unit is separately acknowledged. After taking the acknowledgement, the next data unit is transmitted.
- Error recovery is by retransmission after timeout. TFTP uses adaptive timeout with an exponential back-off algorithm.

TFTP Messages: There are five types of TFTP messages, RRQ, WRQ, DATA, ACK, and ERROR, as shown in Fig:



1. **Read Request** −The client uses this command to get 0 copy of a file from the server

Read Request (1)	File Name	0	mode	0
2 octets	variable	1 octet	variable	1 octet

2. **Write Request** − The client uses this command to write a file into the server

Read Request (1)	File Name	0	mode	0
2 octets	variable	1 octet	variable	1 octet

3. **Data** − This TFTP message contains blocks of data.

Data (3)	Sequence Number	Data
2 octets	2 octets	up to 512 octets

4. **Acknowledgement**− The client and the server used this to acknowledge the received data units.

Ack (4)	Sequence Number
2 octets	2 octets

ERROR: The ERROR message is used by the client or the server when a connection cannot be established or when there is a problem during data transmission. It can be sent as a negative response to RRQ or WRQ. It can also be used if the next block cannot be transferred during the actual data transfer phase. The error message is not used to declare a damaged or duplicated message.

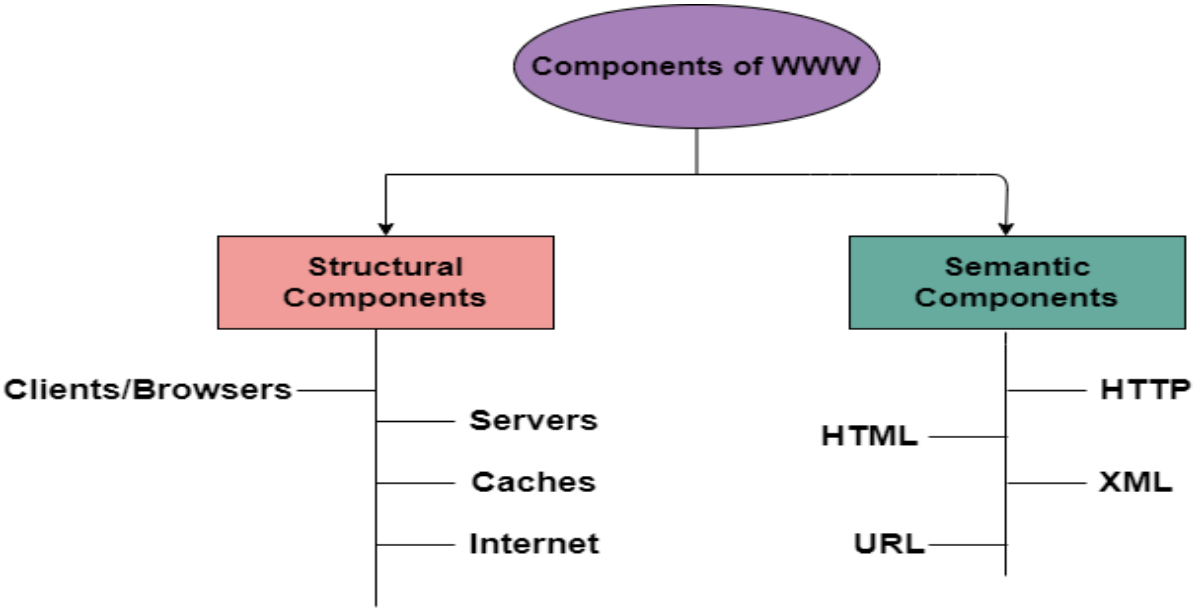
Difference Between FTP and TFTP:

FTP	TFTP
FTP stands for File Transfer Protocol.	TFTP stands for Trivial File Transfer Protocol.
The software of FTP is larger than TFTP.	While software of TFTP is smaller than FTP.
FTP works on two ports: 20 and 21.	While TFTP works on 69 Port number.
FTP services are provided by TCP.	While TFTP services are provided by UDP.
The complexity of FTP is higher than TFTP.	While the complexity of TFTP is less than FTP complexity.
There are many commands or messages in FTP.	There are only 5 messages in TFTP.
FTP need authentication for communication.	While TFTP does not need authentication for communication.
FTP is generally suited for uploading and downloading of files by remote users.	While TFTP is mainly used for transmission of configurations to and from network devices.
FTP is a reliable transfer protocol.	While; TFTP is an unreliable transfer protocol.
FTP is based on TCP.	While; TFTP is based on UDP.
FTP is slower.	TFTP is faster as compared to FTP.

WWW (World Wide Web): The **World Wide Web** or Web is basically a collection of information that is linked together from points all over the world. It is also abbreviated as **WWW.** It provides flexibility, portability, and user-friendly features. It mainly consists of a worldwide collection of electronic documents (i.e, Web Pages). It is basically a way of exchanging information between computers on the Internet. The WWW is mainly the network of pages consists of images, text, and sounds on the Internet which can be simply viewed on the browser by using the browser software. It was invented by Tim Berners-Lee.

Components of WWW: The Components of WWW mainly falls into two categories:

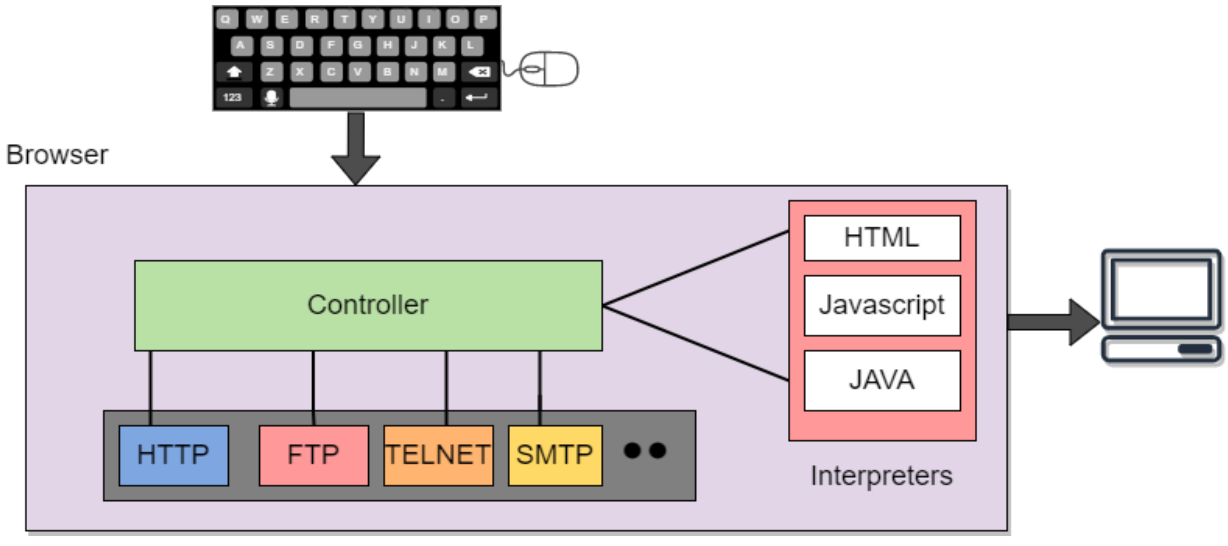
1. Structural Components
2. Semantic Components



1. **Client/Browser:**

- The Client/Web browser is basically a program that is used to communicate with the webserver on the Internet.

- Each browser mainly comprises of three components and these are:
  - Controller
  - Interpreter
  - Client Protocols
- The Controller mainly receives the input from the input device, after that it uses the client programs in order to access the documents.
- After accessing the document, the controller makes use of an interpreter in order to display the document on the screen.
- An interpreter can be Java, HTML, javascript mainly depending upon the type of the document.
- The Client protocol can be FTP, HTTP, TELNET.



2. Server: The Computer that is mainly available for the network resources and in order to provide services to the other computer upon request is generally known as the **server.**

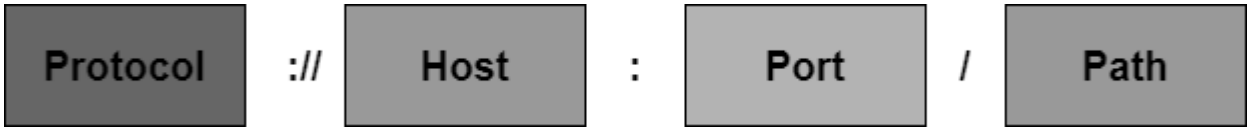
- The Web pages are mainly stored on the server.
- Whenever the request of the client arrives then the corresponding document is sent to the client.
- The connection between the client and the server is TCP.
- It can become more efficient through multithreading or multiprocessing. Because in this case, the server can answer more than one request at a time.

3. **URL**: URL is an abbreviation of **the Uniform resource locator.**

- It is basically a standard used for specifying any kind of information on the Internet.
- In order to access any page the client generally needs an address.
- To facilitate the access of the documents throughout the world HTTP generally makes use of Locators.

URL mainly defines the four things:

- **Protocol** It is a client/server program that is mainly used to retrieve the document. A commonly used protocol is HTTP.
- **Host Computer** It is the computer on which the information is located. It is not mandatory because it is the name given to any computer that hosts the web page.
- **Port** The URL can optionally contain the port number of the server. If the port number is included then it is generally inserted in between the host and path and is generally separated from the host by the colon.
- **Path** It indicates the pathname of the file where the information is located.



4.HTML: HTML is an abbreviation of Hypertext Markup Language.

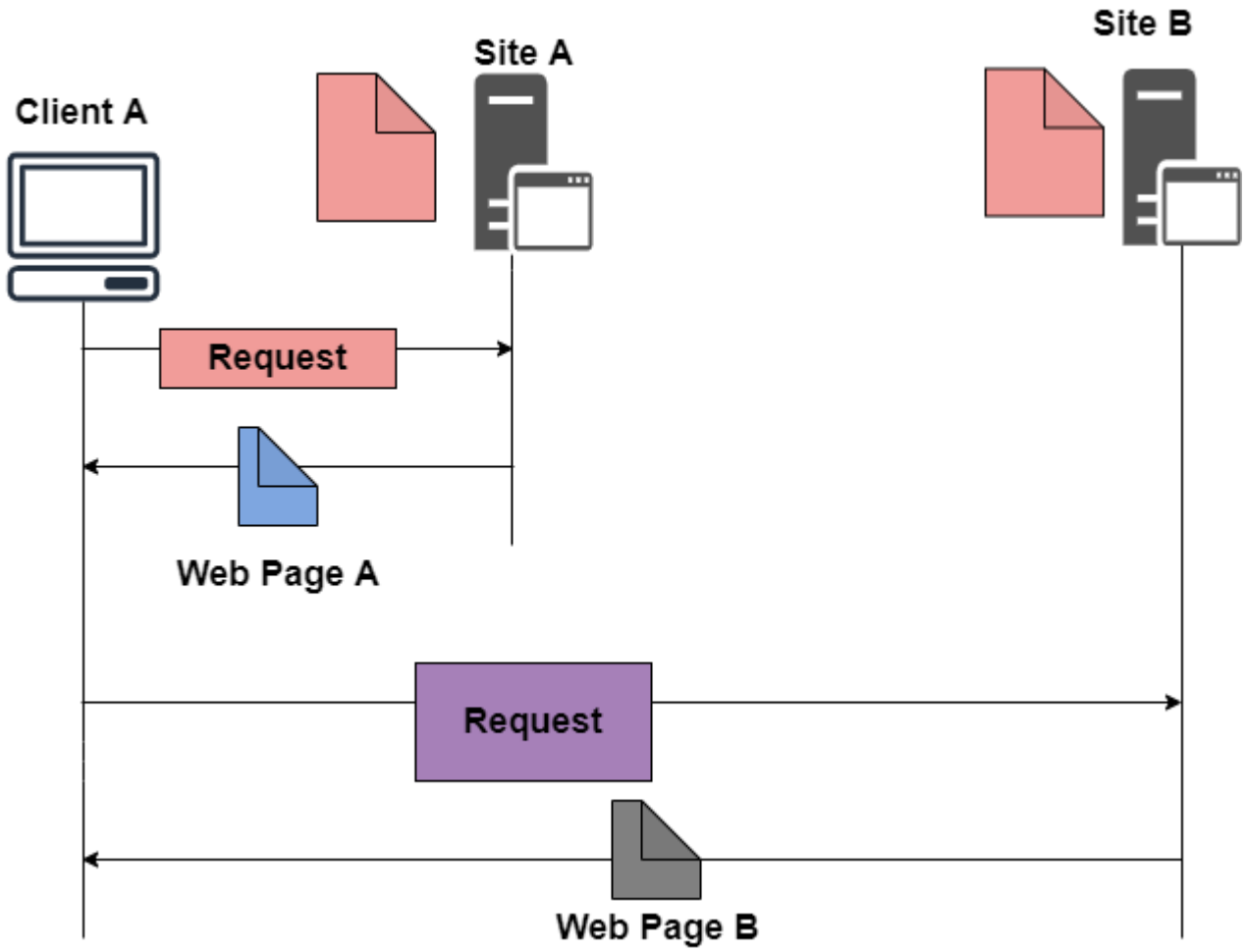
- It is generally used for creating web pages.
- It is mainly used to define the contents, structure, and organization of the web page.

5.XML: XML is an abbreviation of Extensible Markup Language. It mainly helps in order to define the common syntax in the semantic web.

Architecture of WWW: The **WWW** is mainly a distributed **client/server** service where a client using the browser can access the service using a server. The Service that is provided is distributed over many different locations commonly known as **sites/websites.**

- Each website holds one or more documents that are generally referred to as **web pages.**

- Where each web page contains a link to other pages on the same site or at other sites.
- These pages can be retrieved and viewed by using browsers.



In the above case, the client sends some information that belongs to **site A**. It generally sends a request through its browser (It is a program that is used to fetch the documents on the web). and also the request generally contains other information like the address of the site, web page(URL).

The server at **site A** finds the document then sends it to the client. after that when the user or say the client finds the reference to another document that includes the web page at **site B**. The reference generally contains the URL of site B. And the client is interested to take a look at this document too. Then after the client sends the request to the new site and then the new page is retrieved

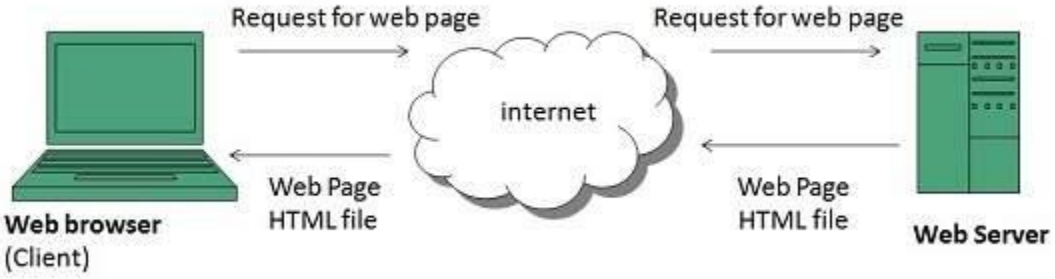
Features of WWW

- Provides a system for Hypertext information
- Open standards and Open source
- Distributed.
- Mainly makes the use of Web Browser in order to provide a single interface for many services.

- Dynamic
- Interactive
- Cross-Platform

WWW Operation: WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



Advantages of WWW

- It mainly provides all the information for Free.
- Provides rapid Interactive way of Communication.
- It is accessible from anywhere.
- It has become the Global source of media.
- It mainly facilitates the exchange of a huge volume of data.

Disadvantages of WWW

- It is difficult to prioritize and filter some information.
- There is no guarantee of finding what one person is looking for.
- There occurs some danger in case of overload of Information.
- There is no quality control over the available data.
- There is no regulation.

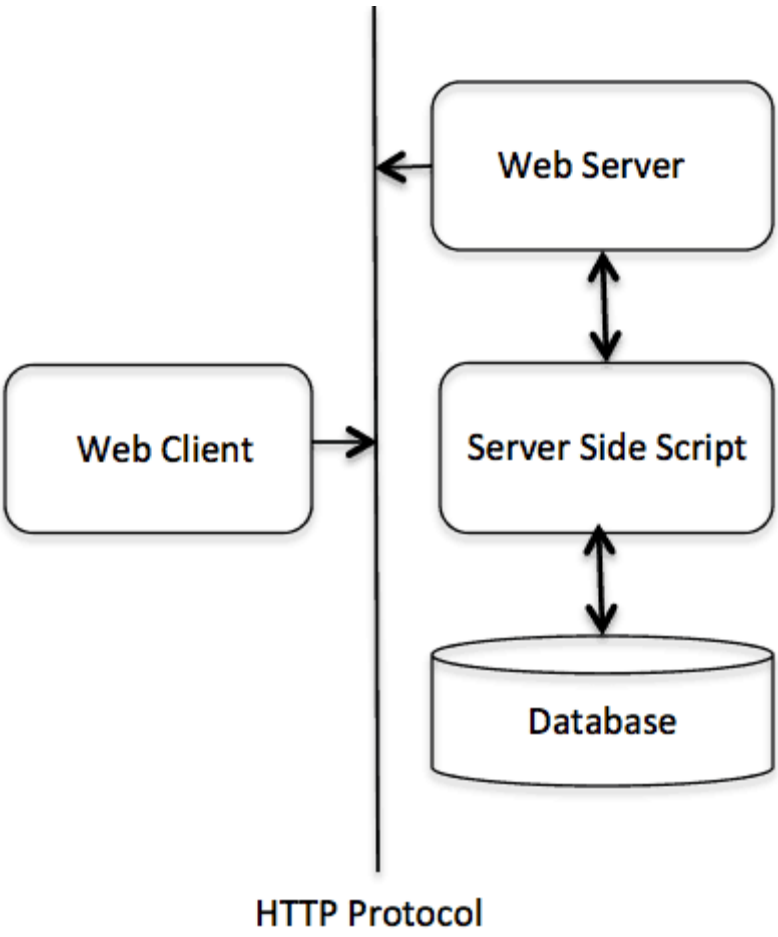
Hypertext Transfer Protocol (HTTP): HTTP stands for **HyperText Transfer Protocol**. It is an application level protocol used to access the data on the World Wide Web (www). It can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on. It is efficient that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.

- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files. HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

Features of HTTP:

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server know about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTP Architecture: The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients and the Web server acts as a server.

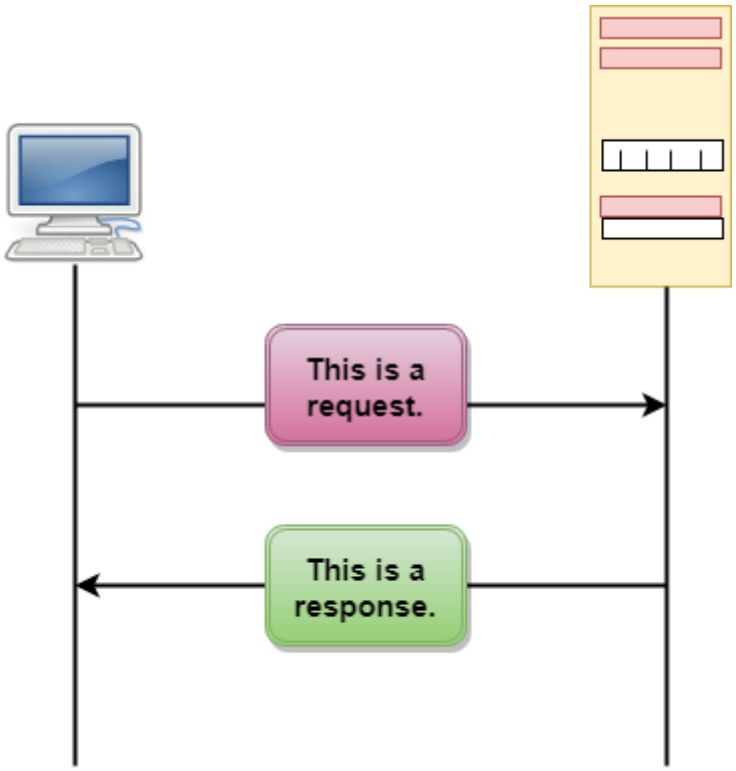
HTTP Client: The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

HTTP Server: The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

Header Fields: HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers:

- **General-header:** These header fields have general applicability for both request and response messages.
- **Request-header:** These header fields have applicability only for request messages.
- **Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** These header fields define meta information about the entity-body or, if nobody is present, about the resource identified by the request

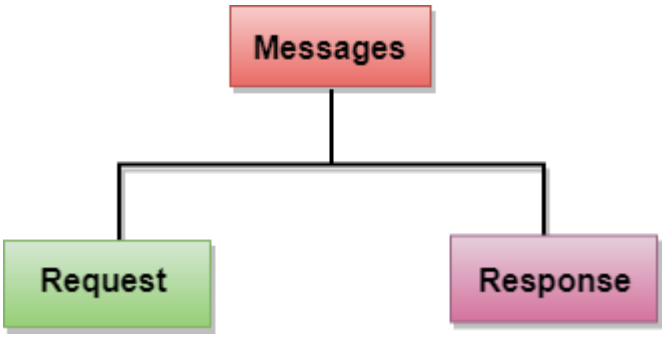
HTTP Transactions



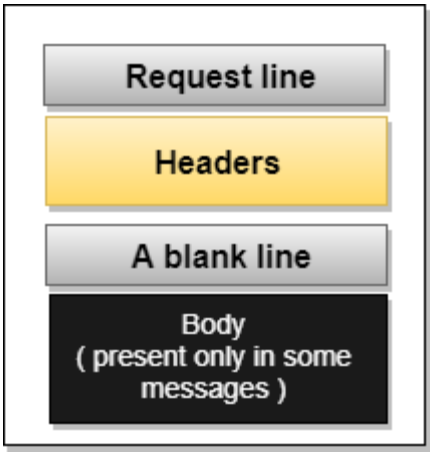
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

HTTP Messages

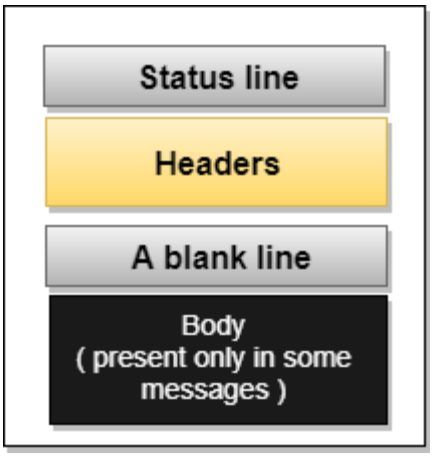
HTTP messages are of two types: request and response. Both the message types follow the same message format.



Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.



Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



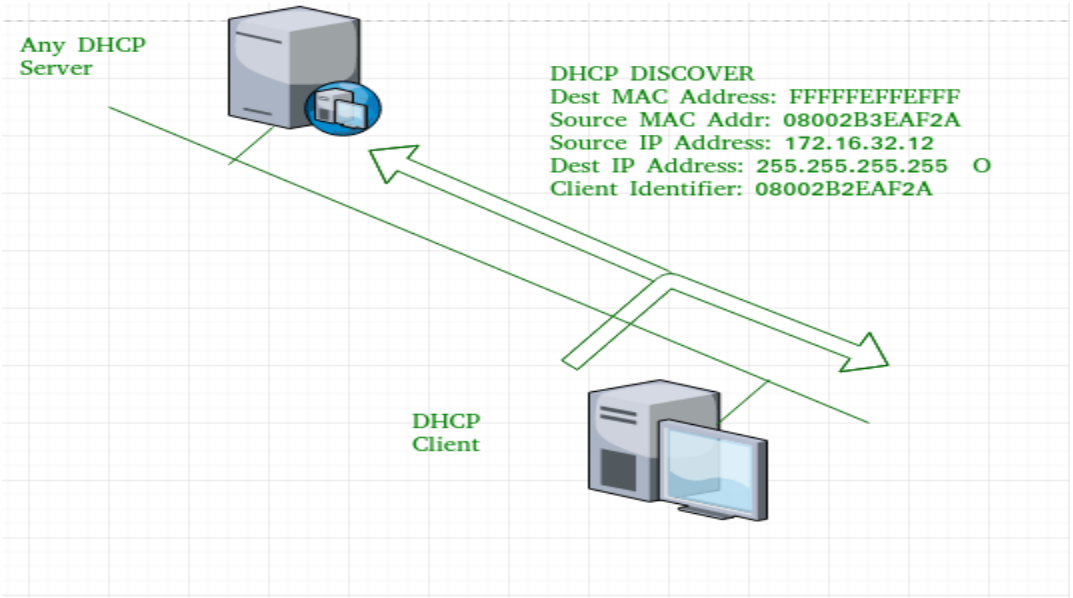
Uniform Resource Locator (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.

- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contains slashes that separate the directories from the subdirectories and files.

Dynamic Host Configuration Protocol (DHCP) is an application layer protocol based on discovery, offer, request, and ACK. DHCP **port number** for server is 67 and for the client is 68. It is a Client server protocol which uses UDP services. IP address is assigned from a pool of addresses. In DHCP, the client and the server exchange mainly 4 DHCP messages in order to make a connection, also called DORA process, but there are 8 DHCP messages in the process. These messages are given as below:

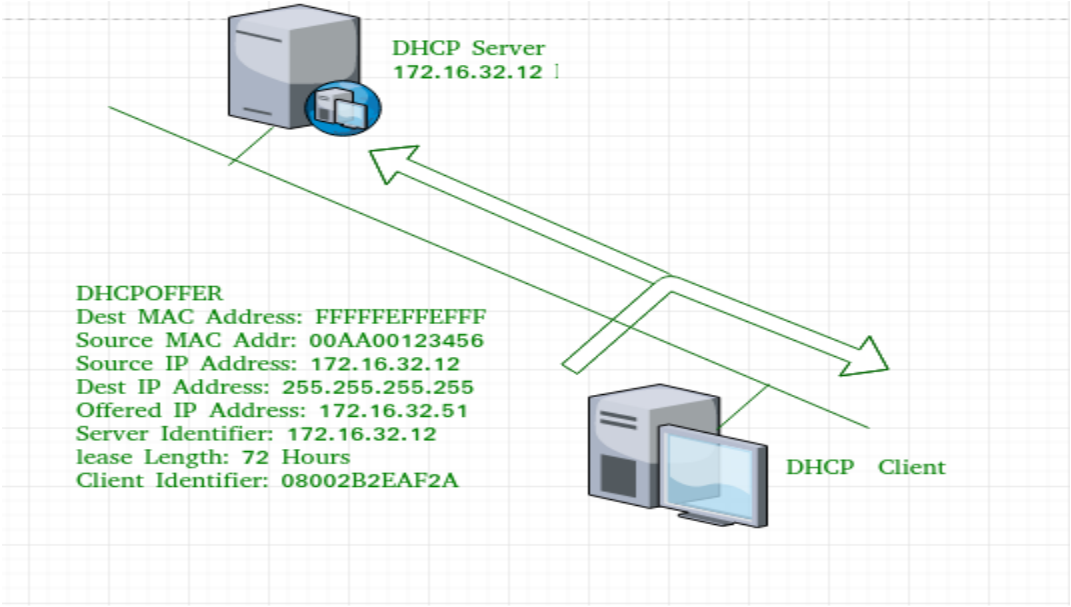
1. **DHCP discover message –** This is a first message generated in the communication process between server and client. This message is generated by Client host in order to discover if there is any DHCP server/servers are present in a network or not. This message is broadcasted to all devices present in a network to find the DHCP server. This message is 342 or 576 bytes long.



As shown in the figure, source MAC address (client PC) is 08002B2EAF2A, destination MAC address(server) is FFFFFFFFFFFF, source IP address is 0.0.0.0(because PC has no IP address till now) and destination IP address is 255.255.255.255 (IP address used for broadcasting). As the discover message is broadcast to find out the DHCP server or servers

in the network therefore broadcast IP address and MAC address is used.

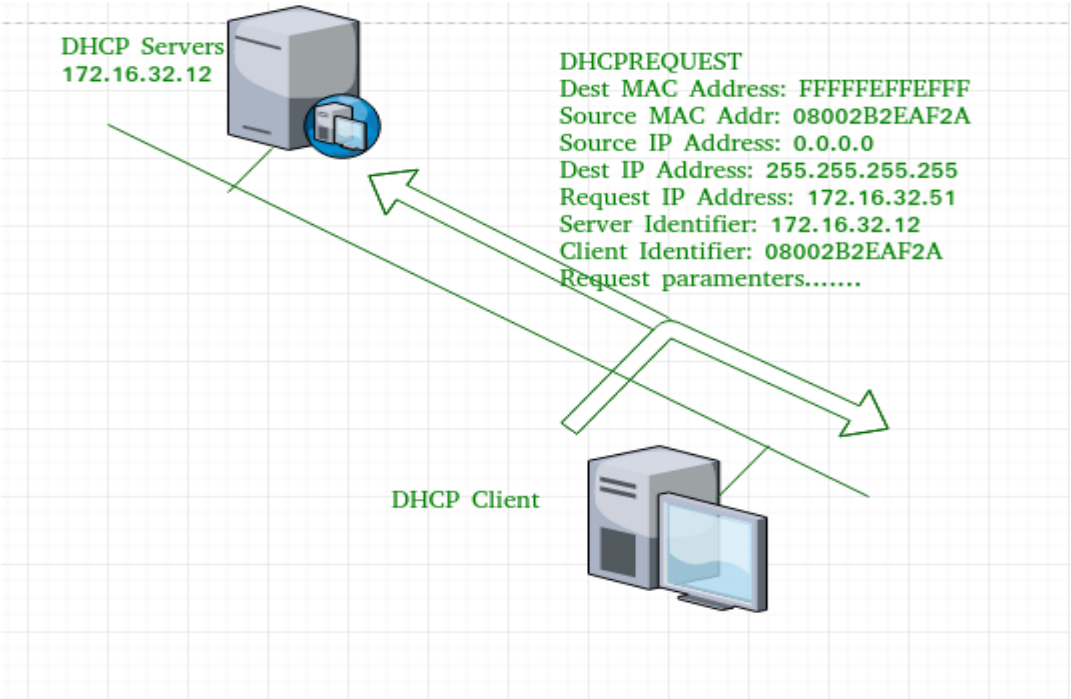
2. **DHCP offer message** – The server will respond to host in this message specifying the unleased IP address and other TCP configuration information. This message is broadcasted by server. Size of message is 342 bytes. If there are more than one DHCP servers present in the network then client host will accept the first DHCP OFFER message it receives. Also a server ID is specified in the packet in order to identify the server.



Now, for the offer message, source IP address is 172.16.32.12 (server's IP address in the example), destination IP address is 255.255.255.255 (broadcast IP address) ,source MAC address is 00AA00123456, destination MAC address is FFFFFFFFFFFF. Here, the offer message is broadcast by the DHCP server therefore destination IP address is broadcast IP address and destination MAC address is FFFFFFFFFFFF and the source IP address is server IP address and MAC address is server MAC address.

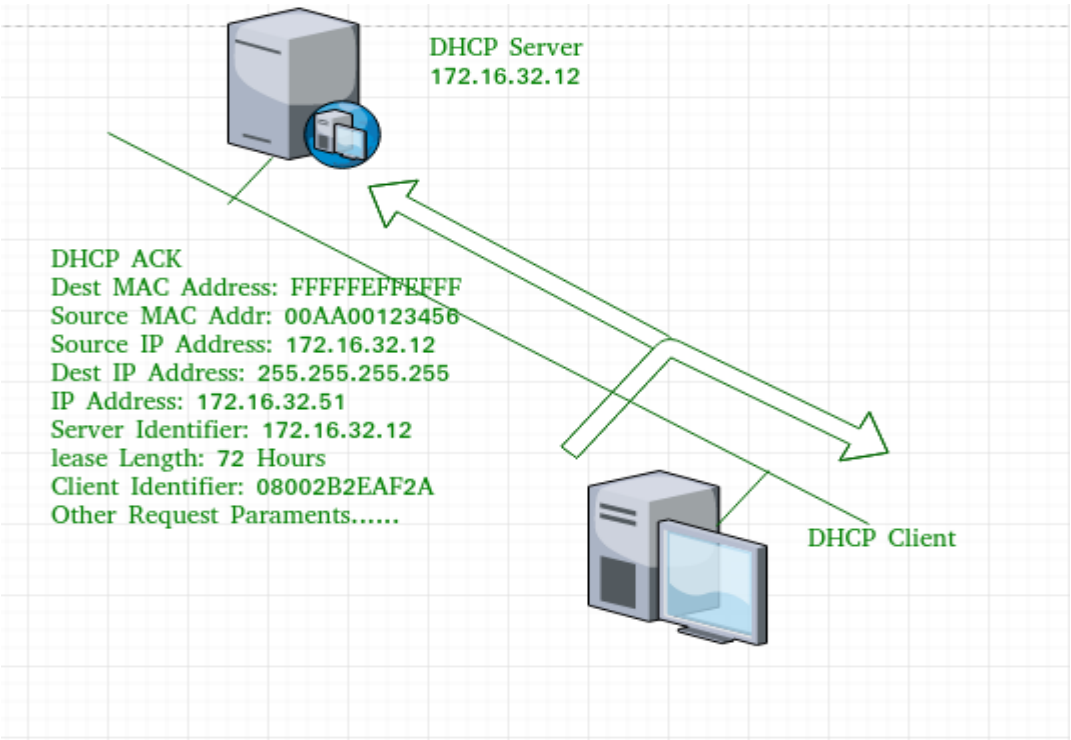
Also the server has provided the offered IP address 192.16.32.51 and lease time of 72 hours(after this time the entry of host will be erased from the server automatically) . Also the client identifier is PC MAC address (08002B2EAF2A) for all the messages.

3. **DHCP request message** – When a client receives a offer message, it responds by Broadcasting a DHCP request message. The client will produce a gratuitous ARP in order to find if there is any other host present in the network with same IP address. If there is no reply by other host, then there is no host with same TCP configuration in the network and the message is broadcasted to server showing the acceptance of IP address. A Client ID is also added in this message.



Now, the request message is broadcast by the client PC therefore source IP address is 0.0.0.0(as the client has no IP right now) and destination IP address is 255.255.255.255 (broadcast IP address) and source MAC address is 08002B2EAF2A (PC MAC address) and destination MAC address is FFFFFFFFFFFF.

4. **DHCP acknowledgement message** – In response to the request message received, the server will make an entry with specified client ID and bind the IP address offered with lease time. Now, the client will have the IP address provided by server.



Now the server will make an entry of the client host with the offered IP address and lease time. This IP address will not be provided by server to any other host. The destination MAC address is FFFFFFFFFFFF and the destination IP address is 255.255.255.255 and the source IP address is 172.16.32.12 and the source MAC address is 00AA00123456 (server MAC address).

5. **DHCP negative acknowledgement message –** Whenever a DHCP server receives a request for IP address that is invalid according to the scopes that is configured with, it send DHCP Nak message to client. Eg-when the server has no IP address unused or the pool is empty, then this message is sent by the server to client.

6. **DHCP decline –** If DHCP client determines the offered configuration parameters are different or invalid, it sends DHCP decline message to the server .When there is a reply to the gratuitous ARP by any host to the client, the client sends DHCP decline message to the server showing the offered IP address is already in use.

7. **DHCP release –** A DHCP client sends DHCP release packet to server to release IP address and cancel any remaining lease time.

8. **DHCP inform –**If a client address has obtained IP address manually then the client uses a DHCP inform to obtain other local configuration parameters, such as domain name. In reply to the dhcp inform message, DHCP server generates DHCP ack message with local configuration suitable for the client without allocating a new IP address. This DHCP ack message is unicast to the client.

Advantages – The advantages of using DHCP include:

1. centralized management of IP addresses
2. ease of adding new clients to a network
3. reuse of IP addresses reducing the total number of IP addresses that are required
4. simple reconfiguration of the IP address space on the DHCP server without needing to reconfigure each client
5. The DHCP protocol gives the network administrator a method to configure the network from a centralised area.
6. With the help of DHCP, easy handling of new users and reuse of IP address can be achieved.

Disadvantages – Disadvantage of using DHCP is:

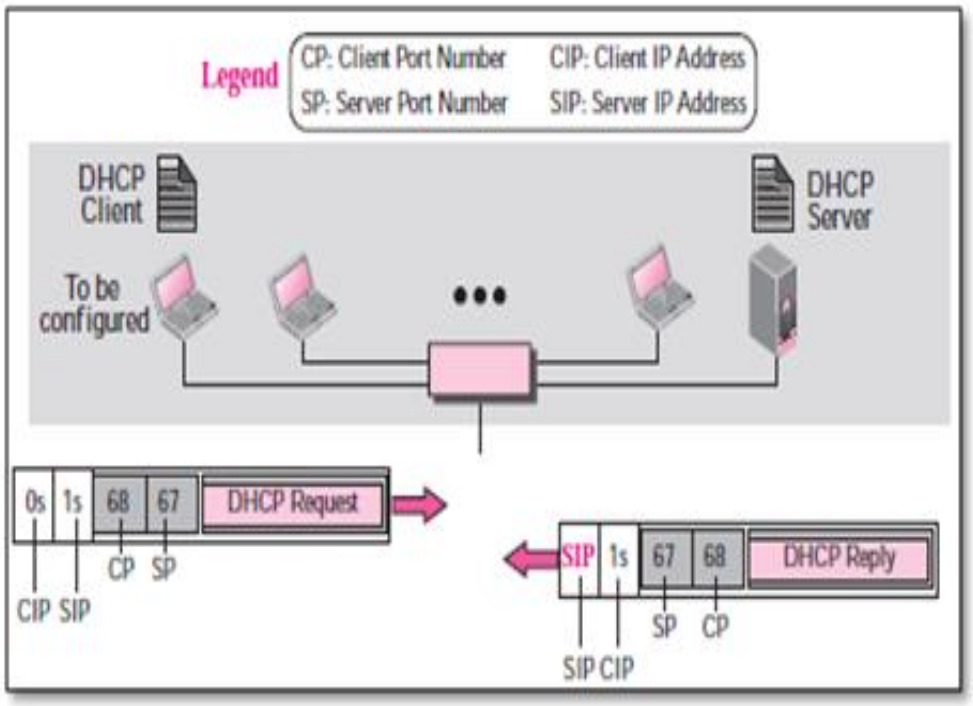
1. IP conflict can occur

DHCP OPERATION: The DHCP client and server can either be on the same network or on different networks.

.

Same Network

Although the practice is not very common, the administrator may put the client and the server on the same network as shown in Figure



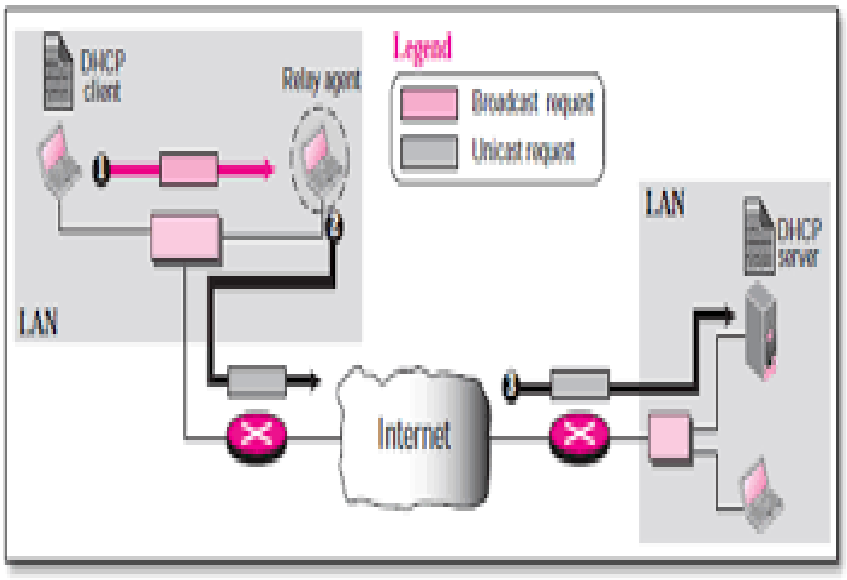
In this case, the operation can be described as follows:

1. The DHCP server issues a passive open command on UDP port number 67 and waits for a client.

2. A booted client issues an active open command on port number 68 (this number will be explained later). The message is encapsulated in a UDP user datagram, using the destination port number 67 and the source port number 68. The UDP user datagram, in turn, is encapsulated in an IP datagram. The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client uses all 0s as the source address and all 1s as the destination address.

3. The server responds with either a broadcast or a unicast message using UDP source port number 67 and destination port number 68. The response can be unicast because the server knows the IP address of the client. It also knows the physical address of the client, which means it does not need the services of ARP for logical to physical address mapping. However, some systems do not allow the bypassing of ARP, resulting in the use of the broadcast address.

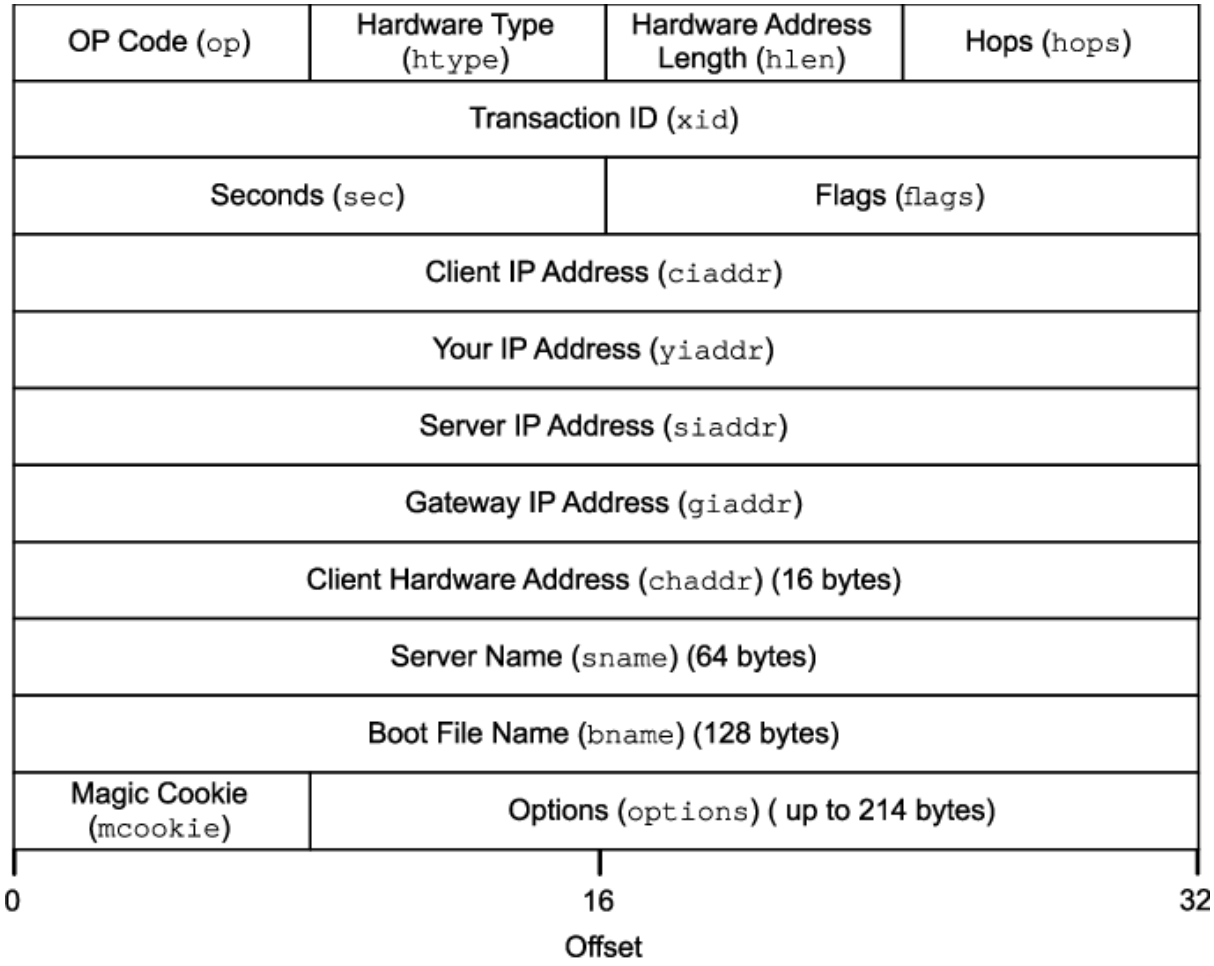
Different Networks: As in other application-layer processes, a client can be in one Network and the server in another, separated by several other networks. Figure shows the situation.



However, there is one problem that must be solved. The DHCP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. A router receiving such a packet discards it. Recall that an IP address of all 1s is a limited broadcast address.

To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay. The host in this case is called a **relay agent.** The relay agent knows the unicast address of a DHCP server and listens for broadcast messages on port 67. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the DHCP server. The packet, carrying a unicast destination address, is routed by any router and reaches the DHCP server. The DHCP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the DHCP client.

DHCP Packet Format: Figure shows the format of a DHCP packet.



The following describes each field:

❑ **Operation code.** This 8-bit field defines the type of DHCP packet: request (1) or reply (2).

❑ **Hardware type.** This is an 8-bit field defining the type of physical network. Each type of network has been assigned an integer. For example, for Ethernet the value is 1.

❑ **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.

❑ **Hop count.** This is an 8-bit field defining the maximum number of hops the packet can travel.

❑ **Transaction ID.** This is a 4-byte field carrying an integer. The transaction identification is set by the client and is used to match a reply with the request. The server returns the same value in its reply.

❑ **Number of seconds.** This is a 16-bit field that indicates the number of seconds elapsed since the time the client started to boot.

❑ **Flag.** This is a 16-bit field in which only the leftmost bit is used and the rest of the bits should be set to 0s. A leftmost bit specifies a forced broadcast reply (instead of unicast) from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client. Since the client does not know its IP address, it may discard the packet. However, if the IP datagram is broadcast, every host will receive and process the broadcast message.

❑ **Client IP address.** This is a 4-byte field that contains the client IP address. If the client does not have this information, this field has a value of 0.
❑ **Your IP address.** This is a 4-byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.
❑ **Server IP address.** This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.
❑ **Gateway IP address.** This is a 4-byte field containing the IP address of a router. It is filled by the server in a reply message.
❑ **Client hardware address.** This is the physical address of the client. Although the server can retrieve this address from the frame sent by the client, it is more efficient if the address is supplied explicitly by the client in the request message.
❑ **Server name.** This is a 64-byte field that is optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the domain name of the server. If the server does not want to fill this field with data, the server must fill it with all 0s.
❑ **Boot filename.** This is a 128-byte field that can be optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the full pathname of the boot file. The client can use this path to retrieve other booting information. If the server does not want to fill this field with data, the server must fill it with all 0s.
❑ **Options.** This is a 64-byte field with a dual purpose. It can carry either additional information (such as the network mask or default router address) or some specific vendor information. The field is used only in a reply message. The server uses a number, called a **magic cookie,** in the format of an IP address with the value of 99.130.83.99. When the client finishes reading the message, it looks for this magic cookie. If present, the next 60 bytes are options. An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field. The length field defines the length of the value field, not the whole option.

DHCP CONFIGURATION: The DHCP has been devised to provide static and dynamic address allocation.

Static Address Allocation: In this capacity, a DHCP server has a database that statically binds physical addresses to IP addresses. When working in this way, DHCP is backward compatible with the deprecated protocol BOOTP.

Dynamic Address Allocation: DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP 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. The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (for example, a subscriber to a service provider). DHCP provides temporary IP addresses for a limited period of time. The addresses assigned from the pool are temporary addresses. The DHCP server issues a **lease** for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.