UNIT 5 - Application Layer

Introduction:

The Application Layer is topmost layer in the Open System Interconnection (OSI) model. This layer provides several ways for manipulating the data (information) which actually enables any type of user to access network with ease. This layer also makes a request to its bottom layer, which is presentation layer for receiving various types of information from it. The Application Layer interface directly interacts with application and provides common web application services. This layer is basically highest level of open system, which provides services directly for application process.

- Application Layer provides a facility by which users can forward several emails and it also provides a storage facility.
- This layer allows users to access, retrieve and manage files in a remote computer.
- It allows users to log on as a remote host.
- This layer provides access to global information about various services.

Application Layer Protocols: The application layer provides several protocols which allow any software to easily send and receive information and present meaningful data to its users.
The following are some of the protocols which are provided by the application layer.

- **TELNET:** Telnet stands for Telecommunications Network. This protocol is used for managing files over the Internet. It allows the Telnet clients to access the resources of Telnet server. Telnet uses port number 23.
- **DNS:** DNS stands for Domain Name System. The DNS service translates the domain name (selected by user) into the corresponding IP address. For example- If you choose the domain name as www.abcd.com, then DNS must translate it as 192.36.20.8 (random IP address written just for understanding purposes). DNS protocol uses the port number 53.
- **DHCP:** DHCP stands for Dynamic Host Configuration Protocol. It provides IP addresses to hosts. Whenever a host tries to register for an IP address with the DHCP server, DHCP server provides lots of information to the corresponding host. DHCP uses port numbers 67 and 68.
- **FTP:** FTP stands for File Transfer Protocol. This protocol helps to transfer different files from one device to another. FTP promotes sharing of files via remote computer devices with reliable, efficient data transfer. FTP uses port number 20 for data access and port number 21 for data control.

- **SMTP:** SMTP stands for Simple Mail Transfer Protocol. It is used to transfer electronic mail from one user to another user. SMTP is used by end users to send emails with ease. SMTP uses port numbers 25 and 587.
- **HTTP:** HTTP stands for Hyper Text Transfer Protocol. It is the foundation of the World Wide Web (WWW). HTTP works on the client server model. This protocol is used for transmitting hypermedia documents like HTML. This protocol was designed particularly for the communications between the web browsers and web servers, but this protocol can also be used for several other purposes. HTTP is a stateless protocol (network protocol in which a client sends requests to server and server responses back as per the given state), which means the server is not responsible for maintaining the previous client's requests. HTTP uses port number 80.
- **NFS:** NFS stands for Network File System. This protocol allows remote hosts to mount files over a network and interact with those file systems as though they are mounted locally. NFS uses the port number 2049.
- **SNMP:** SNMP stands for Simple Network Management Protocol. This protocol gathers data by polling the devices from the network to the management station at fixed or random intervals, requiring them to disclose certain information. SNMP uses port numbers 161 (TCP) and 162 (UDP).

Domain Name System (DNS)

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- Domain Name System is an Internet service that translates domain names into IP addresses.
- The DNS has a distributed database that resides on multiple machines on the Internet.
- DNS has some protocols that allow the client and servers to communicate with each other.

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.

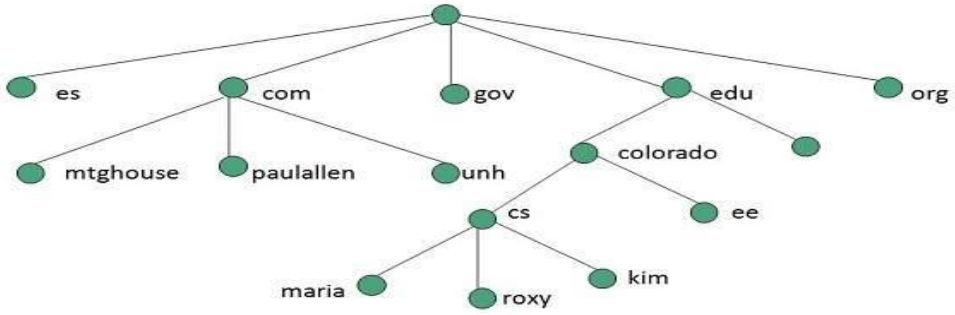
Domain Name	Meaning
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

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

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.

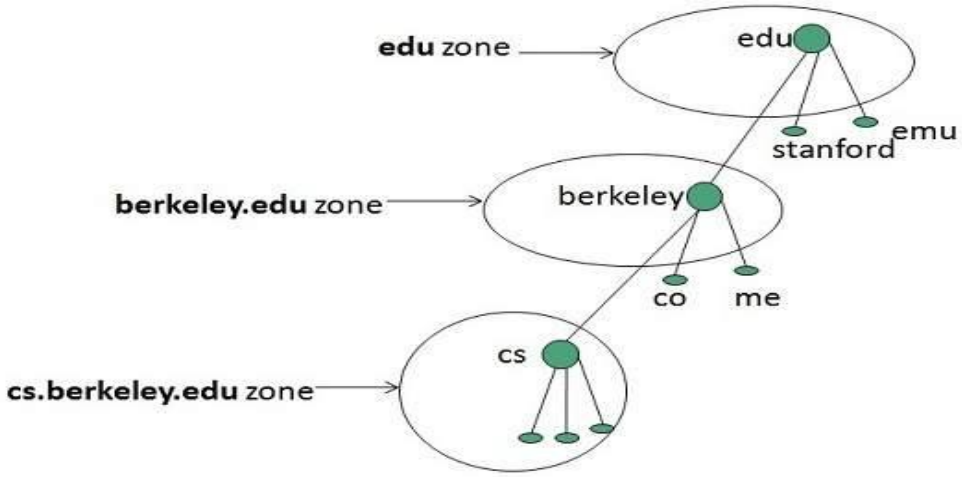
Name Server

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.

DNS Server Types

To complete a **DNS** resolution, different types of servers are used. The order in which a query passes through the four name servers is listed below. They can either supply the requested domain name or refer you to alternative name servers.

- ✓ **DNS Resolver**

An application, such as a web browser, sends DNS queries to the recursive server. It's the user's **first resource**, and it either provides the response to the query if it has it cached, or it goes to the next-level server if it doesn't. Before answering the client, this server may undergo multiple iterations of querying.

- ✓ **Root Name Server**

If the recursive server does not have the answer cached, it first sends a query to this server. The root name server is a directory of all the servers that will hold the requested information. The Internet Corporation for Assigned Names and Numbers **(ICANN)**, specifically the Internet Assigned Numbers Authority, is in charge of these servers.

- ✓ **Top-Level Domain Server**

A TLD nameserver keeps track of all domain names with the same domain extension, such as .com, .net, or whatever comes after the last dot in a URL. For example, a **'.com' TLD nameserver** includes information for every website that ends in '.com'. If a user searched for scaler.com, the DNS resolver would submit a query to a .com **TLD nameserver**, which would answer by referring to the authoritative nameserver for that domain.

- ✓ **Authoritative name server**

A **DNS resolver** will be directed to an authoritative nameserver when it receives a response from a TLD nameserver. The resolver's final stage in the path for an IP address is usually the authoritative nameserver. The server contains information specific to the domain name it serves **(e.g., scaler.com)**. If the authoritative name server has access to the requested record, it will return the requested hostname's IP address to the DNS Recursor (the librarian) who initiated the request.

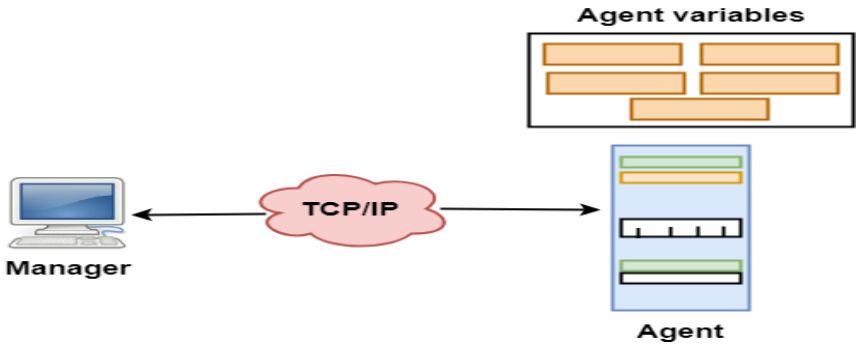
How Does DNS Works?

DNS is concerned with translating a domain name into an **IP address**.

1. If you type facebook.com into a web browser, the query is transmitted over the Internet and received by a DNS resolver.
2. The DNS resolver then queries a **DNS root** nameserver.
3. After then, the root server responds to the DNS resolver with the address of a **TLD DNS** server (such as .com or .net), which keeps the information for the resolver's domains. Our request for facebook.com is directed to the .com **top-level domain** (TLD).
4. The DNS resolver then requests the .com TLD after receiving the address of the TLD by the root server.
5. The IP address of the domain nameserver, facebook.com, is then returned by the TLD server.
6. Finally, the DNS resolver sends a query to the domain's nameserver.
7. The **nameserver returns** the IP address, for facebook.com, to the resolver.
8. The DNS resolver then returns the IP address of the domain that was requested originally to the web browser.
9. An HTTP request is sent to the IP address by the browser.
10. The server returns the webpage to be **rendered** in the browser at that IP.
11. Finally, after all the processes mentioned above, the user can now view the web page on their machine.

Simple Network Management Protocol (SNMP)

- SNMP stands for **Simple Network Management Protocol**.
- SNMP is a framework used for managing devices on the internet.
- It provides a set of operations for monitoring and managing the internet.



- SNMP has two components **Manager** and **agent**.

- The manager is a host that controls and monitors a set of agents such as routers.
- It is an application layer protocol in which a few manager stations can handle a set of agents.
- The protocol designed at the application level can monitor the devices made by different manufacturers and installed on different physical networks.
- It is used in a heterogeneous network made of different LANs and WANs connected by routers or gateways.

Managers & Agents

- A manager is a host that runs the SNMP client program while the agent is a router that runs the SNMP server program.
- Management of the internet is achieved through simple interaction between a manager and agent.
- The agent is used to keep the information in a database while the manager is used to access the values in the database. For example, a router can store the appropriate variables such as a number of packets received and forwarded while the manager can compare these variables to determine whether the router is congested or not.
- Agents can also contribute to the management process. A server program on the agent checks the environment, if something goes wrong, the agent sends a warning message to the manager.

Management with SNMP has three basic ideas:

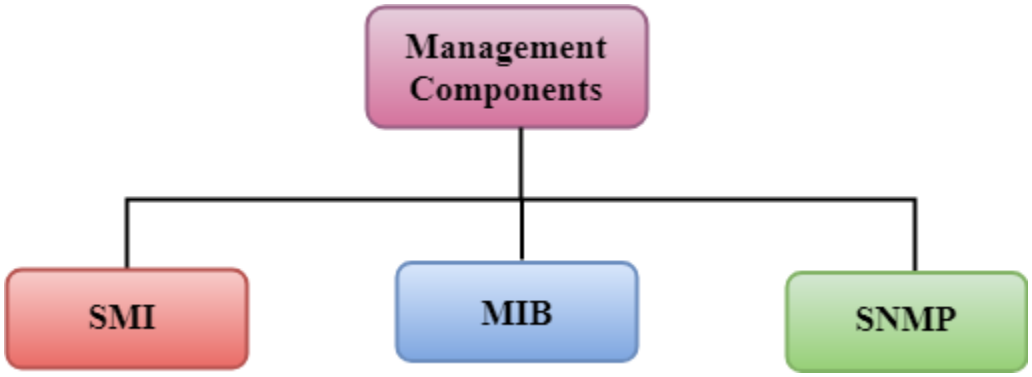
- A manager checks the agent by requesting the information that reflects the behavior of the agent.
- A manager also forces the agent to perform a certain function by resetting values in the agent database.
- An agent also contributes to the management process by warning the manager regarding an unusual condition.

Management Components

- Management is not achieved only through the SNMP protocol but also the use of other protocols that can cooperate with the SNMP protocol. Management is

achieved through the use of the other two protocols: SMI (Structure of management information) and MIB(management information base).

- Management is a combination of SMI, MIB, and SNMP. All these three protocols such as abstract syntax notation 1 (ASN.1) and basic encoding rules (BER).



SMI

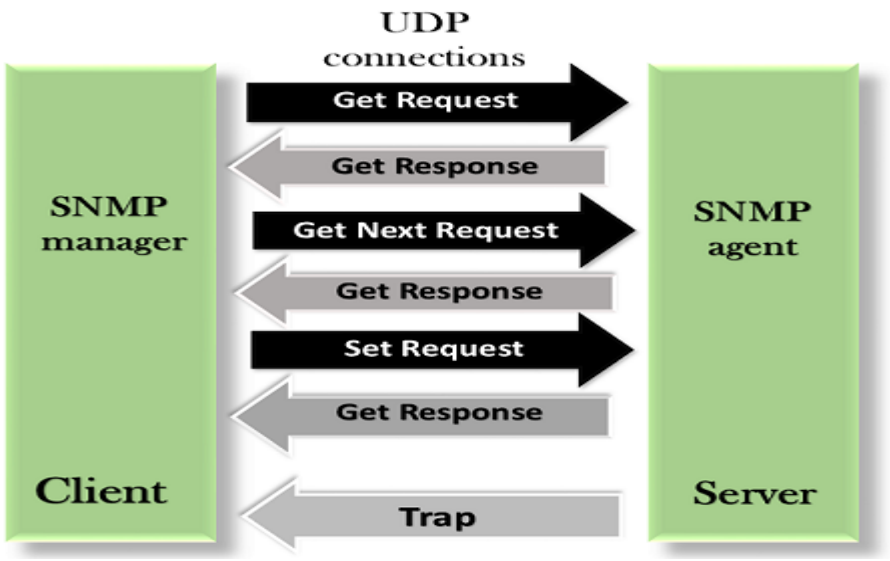
The SMI (Structure of management information) is a component used in network management. Its main function is to define the type of data that can be stored in an object and to show how to encode the data for the transmission over a network.

MIB

- The MIB (Management information base) is a second component for the network management.
- Each agent has its own MIB, which is a collection of all the objects that the manager can manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.

SNMP

SNMP defines five types of messages: GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap.



GetRequest: The GetRequest message is sent from a manager (client) to the agent (server) to retrieve the value of a variable.

GetNextRequest: The GetNextRequest message is sent from the manager to agent to retrieve the value of a variable. This type of message is used to retrieve the values of the entries in a table. If the manager does not know the indexes of the entries, then it will not be able to retrieve the values. In such situations, GetNextRequest message is used to define an object.

GetResponse: The GetResponse message is sent from an agent to the manager in response to the GetRequest and GetNextRequest message. This message contains the value of a variable requested by the manager.

SetRequest: The SetRequest message is sent from a manager to the agent to set a value in a variable.

Trap: The Trap message is sent from an agent to the manager to report an event. For example, if the agent is rebooted, then it informs the manager as well as sends the time of rebooting.

➢ Electronic Mail (E-mail)

Electronic mail is often referred to as E-mail and it is a method used for **exchanging digital messages**.

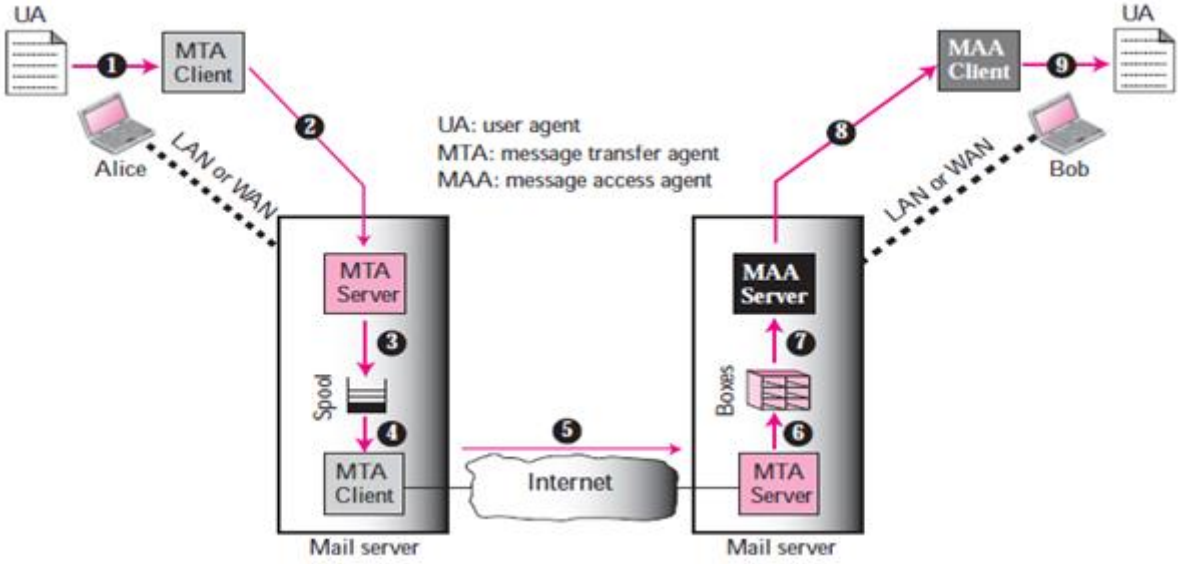
- Electronic mail is mainly designed for **human use**.

- It allows a message to includes **text, image, audio** as well as **video**.
- This service allows one message to be **sent to one or more than one recipient**.
- The E-mail systems are mainly based on the **store-and-forward model** where the E-mail server system accepts, forwards, deliver and store the messages on behalf of users who only need to connect to the infrastructure of the Email.
- The Person who **sends the email** is referred to as **the Sender** while the person who receives an email is referred to as **the Recipient**.

Email architecture:

Email architecture consists of three components:

- **User Agent (UA)**
- **Message Transfer Agent (MTA)**
- **Message Access Agent (MAA)**



User Agent (UA)

The User-Agent is a simple software that sends and receives mail. It is also known as a mail reader. It supports a wide range of instructions for sending, receiving, and replying to messages and manipulating mailboxes.

Some of the services supplied by the User-Agent are listed below:

- Reading a Message
- Sending a reply to a Message
- Message Composition
- Forwarding a Message
- Handling the Message

Message Transfer Agent (MTA)

The Message Transfer Agent manages the actual e-mail transfer operation (MTA). Simple Mail Transfer Protocol sends messages from one MTA to another. A system must have a client MTA

and a system MTA to send an e-mail. If the recipients are connected to the same computer, it sends mail to their mailboxes. If the destination mailbox is on another computer, it sends mail to the receiver's MTA.

Message Access Agent (MAA)

The Simple Mail Transfer Protocol is used for the first and second stages of e-mail delivery. The pull protocol is mainly required at the third stage of e-mail delivery, and the message access agent is used at this point. POP and IMAP4 are the two protocols used to access messages.

Services provided by E-mail system :

- **Composition –** The composition refer to process that creates messages and answers. For composition any kind of text editor can be used.
- **Transfer –** Transfer means sending procedure of mail i.e. from the sender to recipient.
- **Reporting –** Reporting refers to confirmation for delivery of mail. It help user to check whether their mail is delivered, lost or rejected.
- **Displaying –** It refers to present mail in form that is understand by the user.
- **Disposition –** This step concern with recipient that what will recipient do after receiving mail i.e save mail, delete before reading or delete after reading.

➢ World Wide Web (WWW):

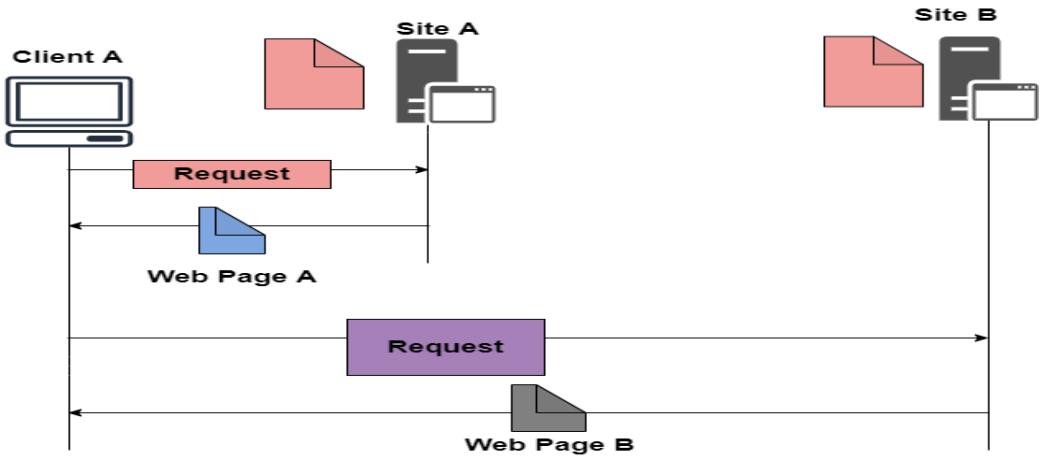
The **World Wide Web** or Web is basically a collection of information that is linked together from points all over the world. It is also abbreviated as **WWW.**

- World wide web 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.

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**.

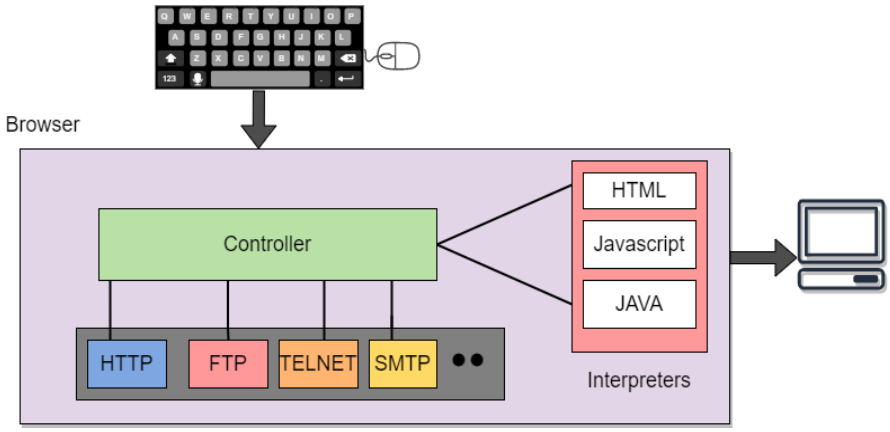
The reference generally contains the URL of site B. And the client is interested to take a look at this document too. Then after the client sends the request to the new site and then the new page is retrieved.

Components of WWW

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.

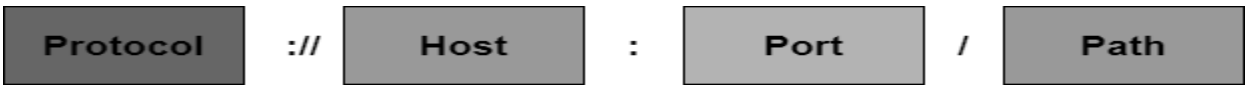
3.URL

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

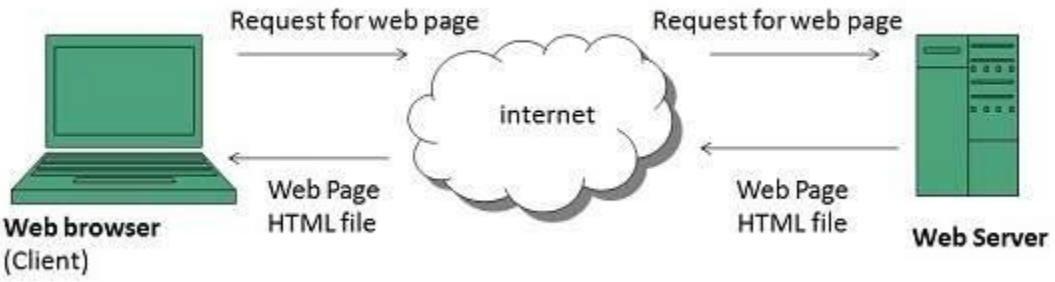
HTML is an abbreviation of Hypertext Markup Language.

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

WWW Operation

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

1. User enters the URL (say, **http://www.facebook.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.



➢ HTTP Protocol

when we enter a URL **(Uniform Resource Locator)** in the browser to visit a website,The browser fetches the IP address corresponding to the entered URL using DNS **(Domain Name System)**. Once the browser gets the IP address corresponding to the entered URL, the browser sends a request to the server at the backend (along with the IP address of the website) to fetch the webpage of the website. In return, the browser receives a response from the server and this response contains the HTML (Hypertext Markup Language) information of the webpage. These **requests** and **responses** that are sent follow a protocol which is called HTTP **(Hypertext Transfer Protocol)** .

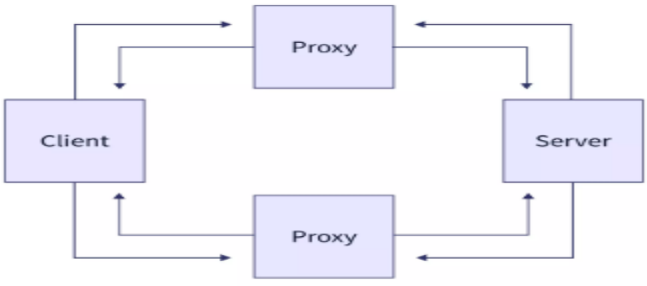
Hence, HTTP is a protocol which is used to transfer hypertext data (example HTML data), plain text data, audio data, video data etc. over the internet.

HTTP relies on the TCP **(Transmission Control Protocol)** and it works on the PORT number 80. Hence, a connection is first established (TCP connection) and then the data is transferred over this connection.

Components of HTTP-based Systems

An HTTP-based system consists of 3 parts i.e. a client, a server and numerous proxies lying in between the client and the server.

1. **Client** A client is the one that sends the HTTP request so as to fetch information from the server at the backend. **For example:** The web browser you are using to surf the internet is the client which is actually requesting the webpages from the web server.



2. **Server** A server is present at the backend which receives requests from the client. It sends an HTTP response back to the client along with the data which is requested by the client.

3. **Proxy** In order to make the process of the request and response faster, we use proxy servers. Proxy servers are generally smaller servers which contain some of the information that is present in the main server.

How Does Hypertext Transfer Protocol Work?

Below is the step by steps discussion of how the Hypertext transfer protocol works.

1. Extracting the IP address of a URL (Uniform Resource Locator)

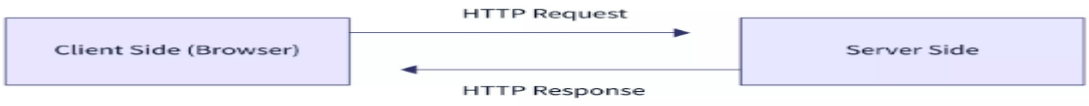
In our browser, we enter the URL address of the website we want to visit. Once we enter the URL address of the website, the browser with the help of the DNS extracts the IP address corresponding to the URL address that is entered.

Domain Name System (DNS)

The DNS contains the mapping of the URLs along with their corresponding IP addresses.

Sending request to the server to access webpage and receiving response

Once we get the IP address of the website, the browser sends an HTTP request to the server to extract the HTML webpage corresponding to the IP address. This request is sent over PORT 80 using TCP. Once the server receives this HTTP request, it responds back with an HTTP response. This HTTP response consists of the information related to the HTML page corresponding to the IP address of the website.



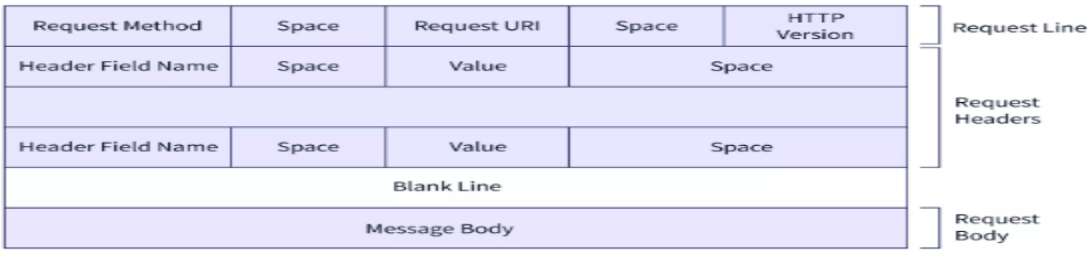
HTTP Request and Response

Receiving HTTP response and displaying the webpage

The browser receives the HTML information for the website along with the response and hence, it processes and displays the HTML page on the browser. Finally, the users can see a HTML page for the URL that they entered.

- **What is in an HTTP Request?**

In order to fetch the webpage corresponding to a given IP address, the browser sends a request (called as HTTP request) to the web server. An HTTP request consists of 3 parts i.e. a request line, request headers, and request body. Given below is the structure of an HTTP request.



Request Line

An HTTP request-line consists of 3 parts which are discussed below.

1. **Request Method** A request method defines what type of request is needed to send to the web server. For example if we want to fetch something from the web server, we use a GET request method. If we want to send something from the client to the web server, we use POST request method.
2. **Request URI** It it the the URI of the website / destination we want to reach. For example "http://scaler.com/".
3. **HTTP Version** It states the version of the HTTP we are using. For example HTTP 1.0, HTTP 1.1.

Request Headers

Request headers contain additional information about the resource/data that is to be fetched from the web server. For example, if we want the data in the plaintext format, we can specify that information in the HTTP headers. We can also specify the information related to the client using headers like the browser it is using etc.

We can send more than one HTTP headers along with an HTTP request to specify the additional information which is to be sent along with an HTTP request.

Request Body

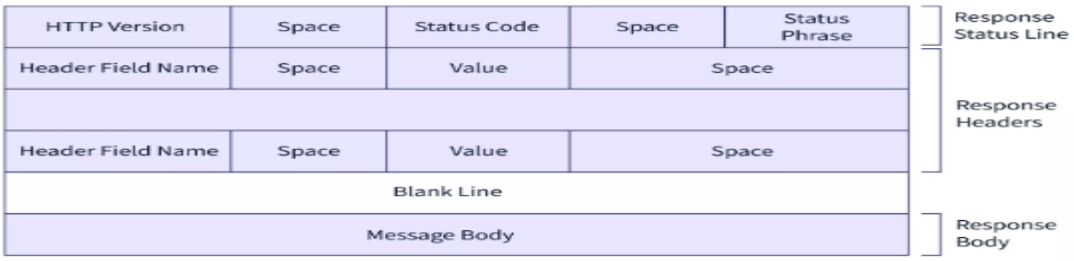
It is an optional part of an HTTP request. If there is a requirement of sending some data along with an HTTP request, we send that data along with the HTTP Body

What is in an HTTP Request Body?

As discussed above, an HTTP body is an optional part of an HTTP request. If there is a requirement of sending some data along with an HTTP request, we send that data along with the HTTP Body. What is in an HTTP Response?

After the client/browser sends an HTTP request to the web server, the server responds back to the browser with an HTTP response. This response consists of all the data that is requested by the client in the HTTP request.

Below is the structure of an HTTP response.



What is an HTTP Status Code?

An HTTP status code is a three-digit code that specifies the status of the HTTP response. Some of the commonly seen HTTP status codes along with their description are given below.

Status Code	Phrase	Description
200	OK	A successful request.
404	Page not found	The requested document is not present.
500	Internal server error	An error such as a server crash has occurred.
400	Bad request	There is some syntax error in the code.