**World Wide Web (WWW):**

WWW is one of the services provided by the internet. WWW is "open-ended service". That means it is possible to access the resources from anywhere & anytime.

The **World Wide Web** (WWW) is a **repository** of information linked together from points all over the world. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

**ARCHITECTURE:**

The WWW today is a distributed client/server service, in which a client is using a **web browser** to access the service from server. The service provided by server is distributed and it is available all over many locations called **websites**.
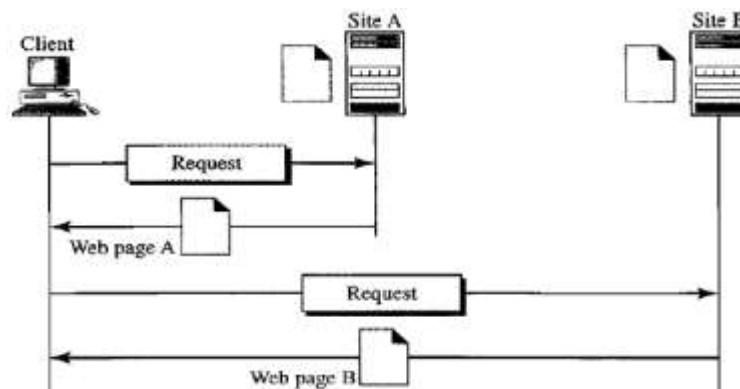


Each website holds one or more documents called as **Web pages.** Each Web page can contain a link to other web pages in the same web site or at other web sites. The pages can be retrieved and viewed by using web browsers.

**Client (Browser):**

At present we are having various commercial browsers used to display a Web document. Al most all web browser's use same architecture.
Each browser usually consists of three parts:
1) Controller.
2) Client protocol.
3) Interpreters.

The **controller receives input** from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to **display** the document on the screen.

The **client protocol** can be one of the protocols **FTP or HTTP**.

The **interpreter** can be **HTML, Java,** or **JavaScript**, depending on the type of document.
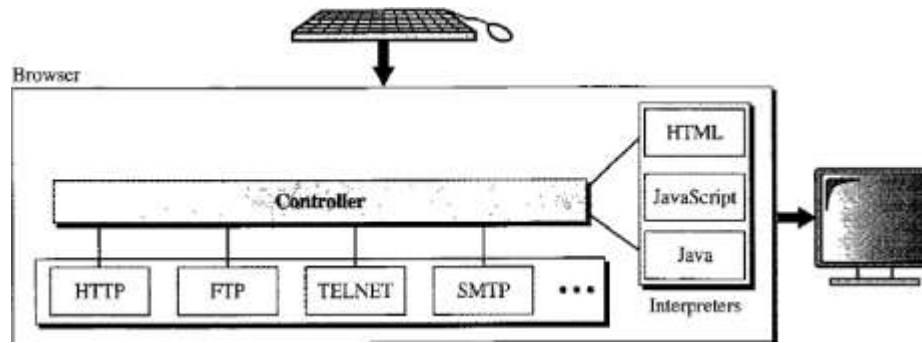
**Server**

The Web pages are stored in server. Each time a client request arrives, the server process the request and the corresponding document is sent from server to client. To improve efficiency, servers normally store mostly requested files in a cache in memory for fast access than disk.
A server can also become more efficient by answering more than one client request at a time.

**Uniform Resource Locator (URL):**

Each webpage or site or document is having an address. That means by using that address it is possible to that anywhere from the world.

Generally it called Locators. Locators are mostly used by HTTP. These Locators are well known as URL. URL is standard to identify and information from the internet.

URL is having four things:



The **protocol** is the client/server program used to retrieve the document. Many different protocols can retrieve a document. Generally we use FTP or HTTP. The most common today is HTTP.
The **host** is the computer on which the information is located. Also we can use the name of the computer as alias.
The **port** indicates the perfect location of the process. It is inserted between the host and the path, and it is separated from the host by a colon.
**Path** is the pathname of the file where the information is located.

**Hypertext Transfer Protocol (HTTP) :**

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP.
HTTP is much simpler than FTP because it uses only one TCP connection, only data are transferred between the client and the server.

HTTP provides request and response approach between client and server.

That means the commands from the client to the server are **embedded** in a **request message**. The contents of the requested file or other information are embedded in a **response message**. HTTP uses the services of TCP on well-known **port 80**. HTTP messages are delivered immediately.

HTTP is a <u>stateless protocol</u>, which means that the connection between the browser and the server is lost once the transaction ends. A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests. HTTP is called a stateless protocol because each command is executed independently, without any knowledge of the commands that came before it.

**HTTP Transaction:**
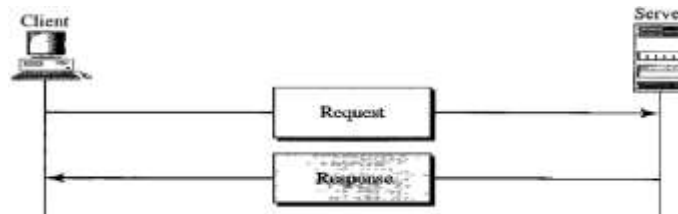


The HTTP transaction between the client and server. Although HTTP uses the services of TCP, HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.

**HTTP Operational model:**

The Hypertext Transfer Protocol is the application-layer protocol that implements the World Wide Web. One basic function of HTTP is to transfer of hypertext documents and other files from Web servers to Web clients. In terms of actual communication, clients are concerned with making requests to servers, and servers are responding to those client requests.

**Basic HTTP Client/Server Communication**

Generally the HTTP uses Client/Server approach. In this the client handles its REQUEST by using its web browser. The HTTP server generally called as WEB SERVER handles the client request and provides the RESPONSE.

To perform Client/Server approach a TCP connection must be created. After connection is established the two steps in communication are as follows:

1. **Client Request:** The HTTP client sends a request message formatted according to the rules of the HTTP standard—an HTTP Request. This message specifies the resource that the client wishes to retrieve, or includes information to be provided to the server.

2. **Server Response:** The server reads and interprets the request. It takes action relevant to the request and creates an HTTP Response message, which it sends back to the client. The response message indicates whether the request was successful, and may also contain the content of the resource that the client requested.

In HTTP the general message format is in two types:
1) Request message.
2) Response message.

The formats of the request and response messages are similar. A request message consists of a request line, a header, and sometimes a body. A response message consists of a status line, a header, and sometimes a body.
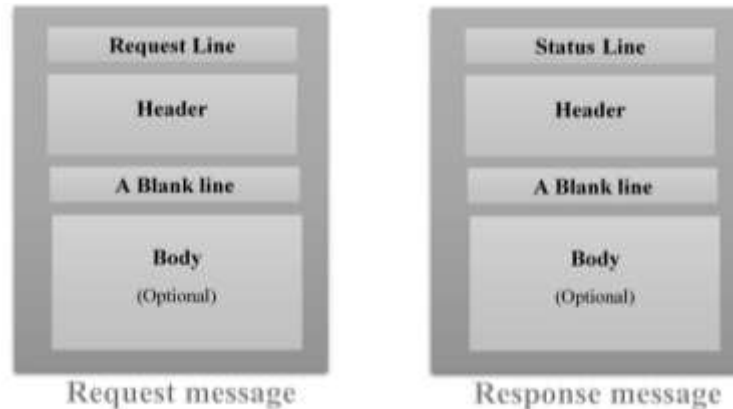


**Request Line**:
The first line in a request message is called a request line; this field is used in the request message. In HTTP, several request types are defined. The request type is categorized into different *methods*.

| GET | Requests a document from the server |
|---|---|
| POST | Sends some information from the client to the server |
| PUT | Sends a document from the server to the client |
| TRACE | Echoes the incoming request |
| CONNECT | Reserved |
| OPTION | Inquires about available options |

**Status line**:
The first line in the response message is called the status line. **Status code** is the field is used in the response message. It consists of three digits.
The codes in the **100** range are only **informational**.
The codes in the **200** range indicate a **successful request**.
The codes in the **300** range **redirect the client to another URL**.
The codes in the **400** range indicate an **error at the client site**.
The codes in the **500** range indicate an **error at the server site**.

| 200 | OK | The request is successful. |
|---|---|---|
| 201 | Created | A new URL is created. |
| 400 | Bad request | There is a syntax error in the request. |
| 401 | Unauthorized | The request lacks proper authorization. |
| 403 | Forbidden | Service is denied. |
| 404 | Not found | The document is not found. |
| 500 | Internal server error | There is an error, such as a crash, at the server site. |

**Header**:

The header exchanges additional information between the client and the server. For example, the client can request that the document be sent in a special format, or the server can send extra information about the document. The header can consist of one or more header lines. Each header line has a header name; a colon, a space, and a header value.
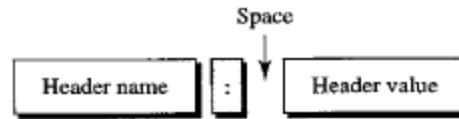


A header line belongs to one of **four** categories:
1) General header.
2) Request header.
3) Response header.
4) Entity header.

A request message can contain only general, request, and entity headers.
A response message contains only general, response, and entity headers.

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

## The Wireless WEB: Wireless Application Protocol.

In this wireless web small portable devices capable of accessing the Web via a wireless link. We are having widely used wide area wireless Web system: WAP.

**Wireless Application Protocol**:

WAP stands for Wireless Application Protocol. WAP is an international standard establishing for how mobile devices such as mobile phones and PDAs can access information on the Internet.

1) The user selects an option on their mobile device that has a URL with Wireless Markup language (WML) content assigned to it. The **phone sends the URL request** via the phone network to a WAP gateway using the binary encoded WAP protocol.

2) The gateway translates this WAP request into a conventional HTTP request for the specified URL and sends it on to the Internet.

3) The appropriate Web server picks up the HTTP request.

4) The server processes the request and the content returned as usual. The Web server adds the HTTP header to the WML content and returns it to the gateway.

5) The WAP gateway compiles the WML into binary form.
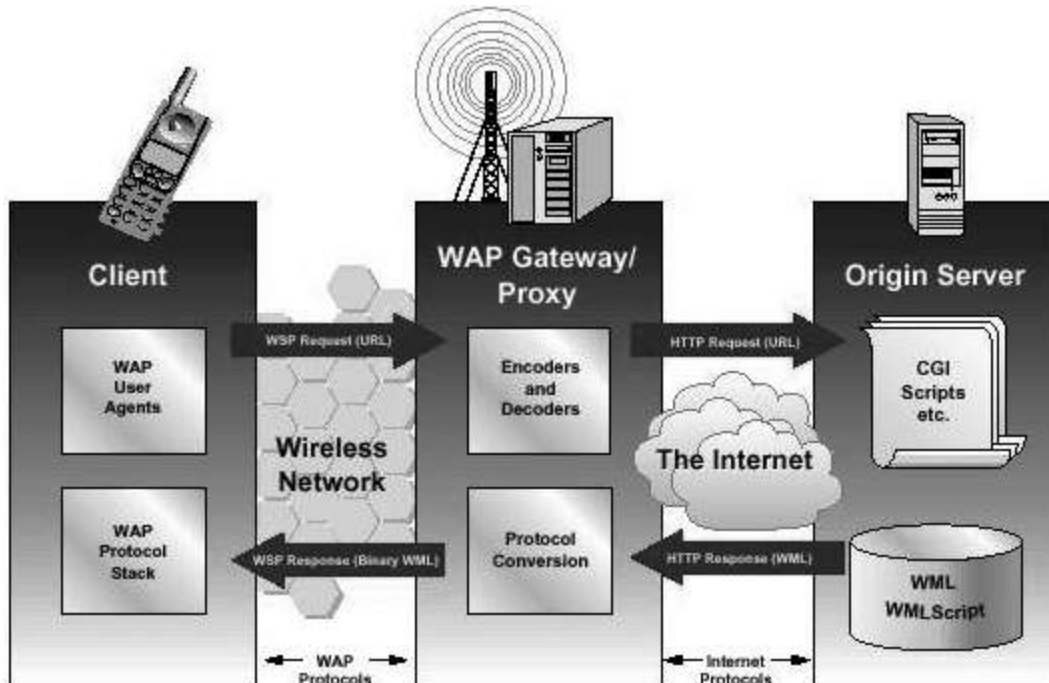
6) The gateway then sends the WML response back to the phone.

7) The phone receives the WML via the WAP protocol.

8) The micro-browser processes the WML and displays the content on the screen.

A WAP device may be an enhanced mobile phone, PDA, or notebook computer. So these devices are known for low-bandwidth connections, slow CPU, little memory, and a small screen. These requirements are different from standard desktop PC scenario. Because of this reason we are having some protocol differences. The layers are:
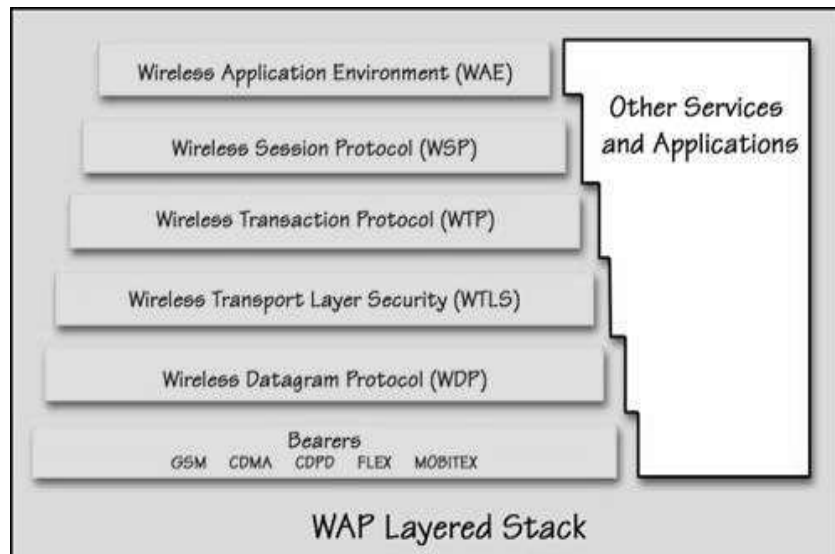
**Wireless Application Environment (WAE):**

The Wireless Application Environment (WAE) is the application layer of the OSI model. It provides the required elements for interaction between Web applications and wireless clients using a WAP micro browser. The micro browser uses the WML (Wireless Markup Language).

**Wireless Session Protocol (WSP)**:

The Wireless Session Protocol is the layer that deals with communication between client and proxy or server.

1. Opens a session of communication between client and server.

2. Establishes a protocol and negotiation between client and server.

3. Exchanges encoded data between client and server.

4. Exchanges request and replies between client and server.

5. Supports several asynchronous transmissions of data.

**Wireless Transaction Protocol (WTP)**:

The Wireless Transaction Protocol, in a simple definition, deals with the transaction of data. WTP handles transactions, re-transmission of data, and with the separation and concatenation of data.

**Wireless Transport Layer Security (WTLS)**:

The Wireless Transport Layer Security is the layer that handles security of data and validity of data between two communicating to manage, start, and finish security issues between two portable devices.

To transport data, WTLS needs: the source address and port number to identify the message creator, and from where the message is being sent, the destination address and port number to which data is being sent, and of course, the data itself.

**Wireless Datagram Protocol (WDP)**:

The Wireless Datagram Protocol allows WAP to be bearer-independent. That means WDP provides an Environment for WAP that can handle any type of bearer. In some areas of WAP, WDP is most similar to UDP.

**Bearers**:

The word Bearers means the carrier or valuator or preventer. The lowest layer supports all the existing mobile phone systems, including GSM, D-AMPS, and CDMA.