

## UNIT I WEB ESSENTIALS




Web Essentials: Clients, Servers, and Communication – The Internet – Basic Internet protocols – World Wide Web – HTTP Request Message – HTTP Response Message – Web Clients – Web Servers

\*\*\*\*\*

**World Wide Web (WWW, W3)**

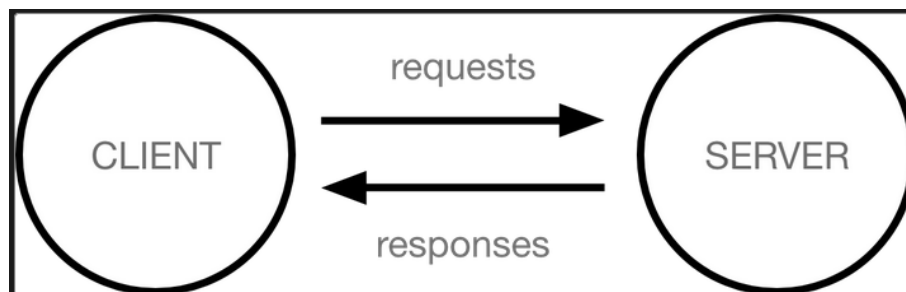
- The World Wide Web—also known as the web, WWW or W3—refers to all the public websites or pages that users can access on their local computers and other devices through the internet.
- British physicist Tim Berners-Lee invented the World Wide Web.
- The *Internet* is an infrastructure, whereas the *Web* is a service built on top of the infrastructure.

## Three iterations of the World Wide Web

	<b>Web 1.0</b> , commonly known as the World Wide Web, enables users to connect to websites and view or download the content. Web 1.0 is considered the genesis of the modern internet, spawned from university experiments and ARPANET to eventually create the internet.
	<b>Web 2.0</b> includes websites and applications that make use of user-generated content, made possible through rich web technologies. This era of the web first materialized in the late 1990s and gave rise to internet giants like Google and Facebook.
	<b>Web 3.0</b> is still being defined at the enterprise level, as the technologies it's based on continue to evolve. Early marketing co-opting the term for consumer technology pushes peer-to-peer, decentralized technologies focused on digital ownership, such as cryptocurrency and nonfungible tokens (NFTs).

**Clients and Servers**

- Computers connected to the internet are called **clients** and **servers**.



- Clients are the typical web user's internet-connected devices (for example, your computer connected to your Wi-Fi, or your phone connected to your mobile network) and web-accessing software available on those devices (usually a web browser like Firefox or Chrome).
- Servers are computers that store webpages, sites, or apps. When a client device wants to access a webpage, a copy of the webpage is downloaded from the server onto the client machine to be displayed in the user's web browser.

### **Basic Terminologies:**

- Internet connection:
  - Allows you to send and receive data on the web. It's basically like the street between your house and the shop.
- TCP/IP:
  - Transmission Control Protocol and Internet Protocol are communication protocols that define how data should travel across the internet. This is like the transport mechanisms that let you place an order, go to the shop, and buy your goods. In our example, this is like a car or a bike (or however else you might get around).
- DNS:
  - Domain Name System is like an address book for websites. When you type a web address in your browser, the browser looks at the DNS to find the website's IP address before it can retrieve the website. The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place (see below). This is like looking up the address of the shop, so you can access it.
- HTTP:
  - Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other. This is like the language you use to order your goods.
- Component files:
  - A website is made up of many files, which are like the different parts of the goods you buy from the shop. These files come in two main types:

- Code files: Websites are built primarily from HTML, CSS, and JavaScript, though you'll meet other technologies a bit later.
  - Assets: This is a collective name for all the other stuff that makes up a website, such as images, music, video, Word documents, and PDFs.
- Web page:
  - A document which can be displayed in a web browser such as Firefox, Google Chrome, Opera, Microsoft Internet Explorer or Edge, or Apple's Safari. These are also often called just “pages.”
- Website:
  - A collection of web pages which are grouped together and usually connected together in various ways. Often called a “website” or a “site.”
- Search engine
  - A web service that helps you find other web pages, such as Google, Bing, Yahoo, or DuckDuckGo.
  - Search engines are normally accessed through a web browser (e.g., you can perform search engine searches directly in the address bar of Firefox, Chrome, etc.) or through a web page (e.g. [bing.com](http://bing.com) or [duckduckgo.com](http://duckduckgo.com)).

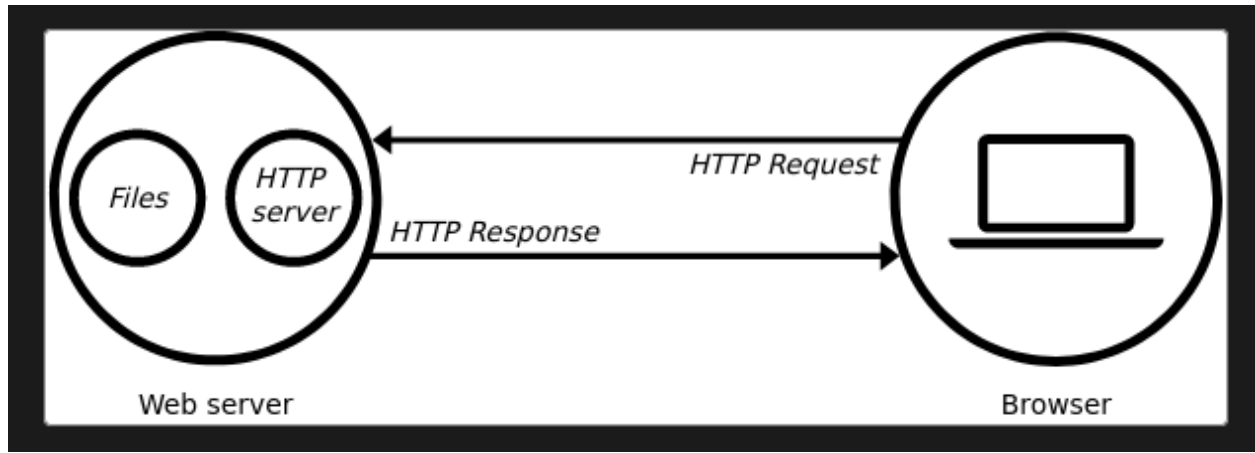
### **Web Client:**

- The web client is the front end or the user side of the web architecture.
- It can be a web browser or a web application that communicates through hypertext transfer protocol (HTTP) to format and transmit data such as documents, images, videos, and audio files from a web server to the end-user.
- Examples:
  - web browsers such as Google Chrome, Internet Explorer, Opera, Firefox, and Safari
  - Zoom

### **Web Server:**

- A web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files).

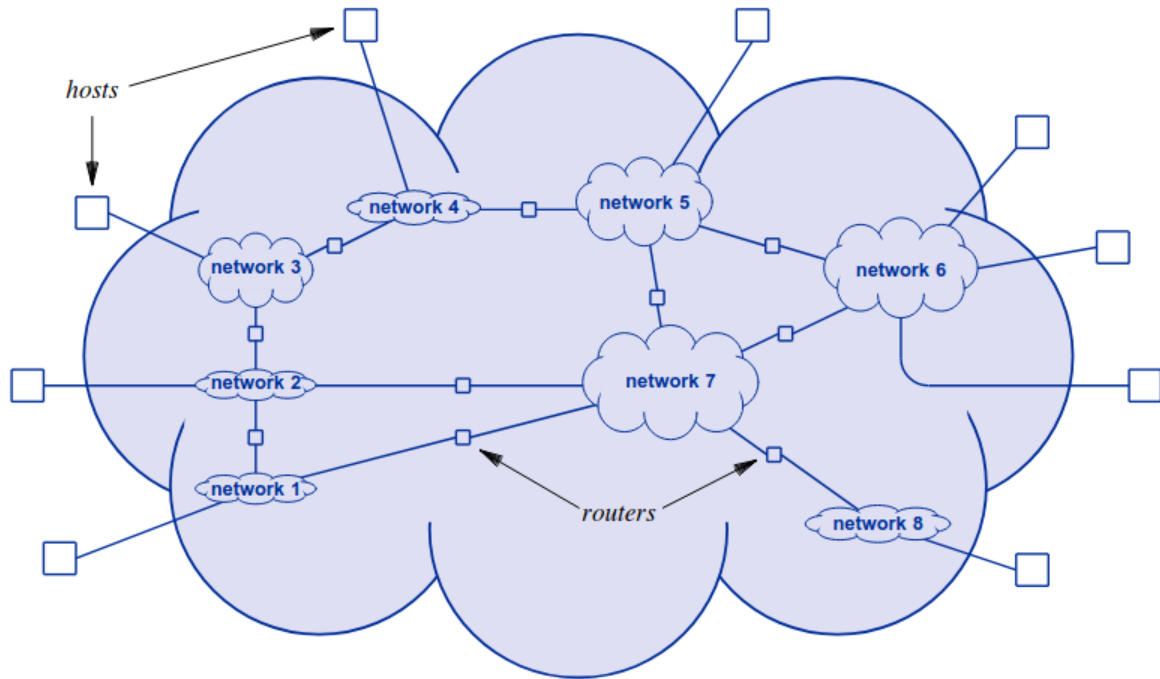
- A web server connects to the Internet and supports physical data interchange with other devices connected to the web.



- A Protocol is a set of rules for communication between two computers. HTTP is a textual, stateless protocol.

### **Internet:**

- It is a global computer communication system that has made all the services possible.
- Internet—the interconnection of networks
- The Internet uses specialized computer systems called routers to interconnect networks.
- A router has multiple network adapters and runs software dedicated to the task of forwarding packets among the networks
- The internet uses a network of networks approach in which thousands of computer networks are interconnected by routers.



**Figure: Illustration of the internal structure of the Internet with many networks interconnected by routers.**

As the figure shows, each host attaches to one of the individual networks. Software on the routers allows a host to exchange packets with any other host. Thus, when a computer on one network communicates with a computer on another network, it sends packets through a router.

### **Internet Protocols**

- Internet Protocols are a set of rules that governs the communication and exchange of data over the internet.
- Both the sender and receiver should follow the same protocols in order to communicate the data.
- In order to understand it better, let's take an example of a language. Any language has its own set of vocabulary and grammar, which we need to know if we want to communicate in that language.
- Similarly, over the internet whenever we access a website or exchange some data with another device, then these processes are governed by a set of rules called the internet protocols.

### **Working on internet protocol:**

- The internet and many other data networks work by organizing data into small pieces called packets. Each large data sent between two network devices is divided into smaller packets by the underlying hardware and software.
- Each network protocol defines the rules for how its data packets must be organized in specific ways according to the protocols the network supports.

### **Why do we need protocols?**

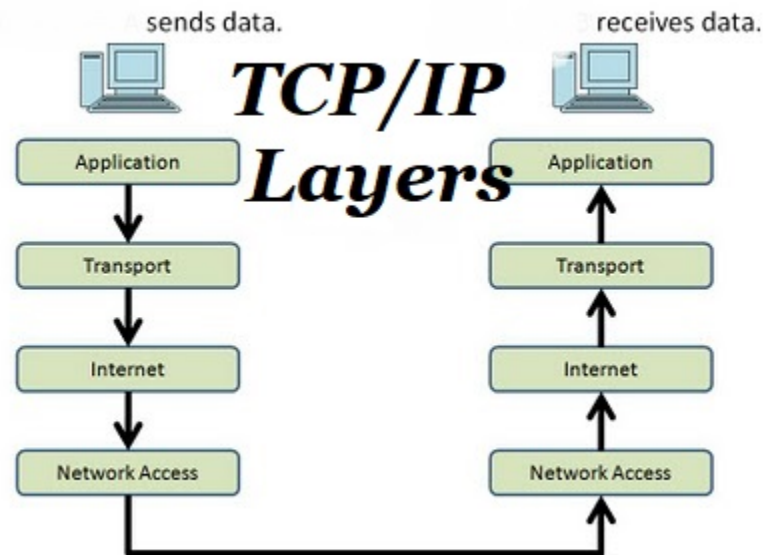
It may be that the sender and receiver of data are parts of different networks, located in different parts of the world, having different data transfer rates. So, we require protocols to manage the flow control of data, and access control of the link being shared in the communication channel. Suppose there is a sender X who has a data transmission rate of 10 Mbps. And, there is a receiver Y who has a data receiving rate of 5Mbps. Since the rate of receiving the data is slow, so some data will be lost during transmission. In order to avoid this, receiver Y needs to inform sender X about the speed mismatch so that sender X can adjust its transmission rate. Similarly, the access control decides the node which will access the link shared in the communication channel at a particular instant of time. If not, the transmitted data will collide if many computers send data simultaneously through the same link, resulting in the corruption or loss of data.

### **Types of internet protocol**

The Internet Protocols are of different types, having different uses:-

**1. TCP/IP (Transmission Control Protocol/ Internet Protocol):** These are a set of standard rules that allows different types of computers to communicate with each other. The IP protocol ensures that each computer that is connected to the Internet is having a specific serial number called the IP address. TCP specifies how data is exchanged over the internet and how it should be broken into IP packets. It also makes sure that the packets have information about the source of the message data, the destination of the message data, the sequence in which the message data should be re-assembled, and checks if the message has been sent correctly to the specific destination.

The functionality of TCP/IP is divided into 4 layers, with each one having specific protocols:



**Application Layer:** The application layer makes sure that the data from the sending end is received in a format that is acceptable and supported at the receiving end.

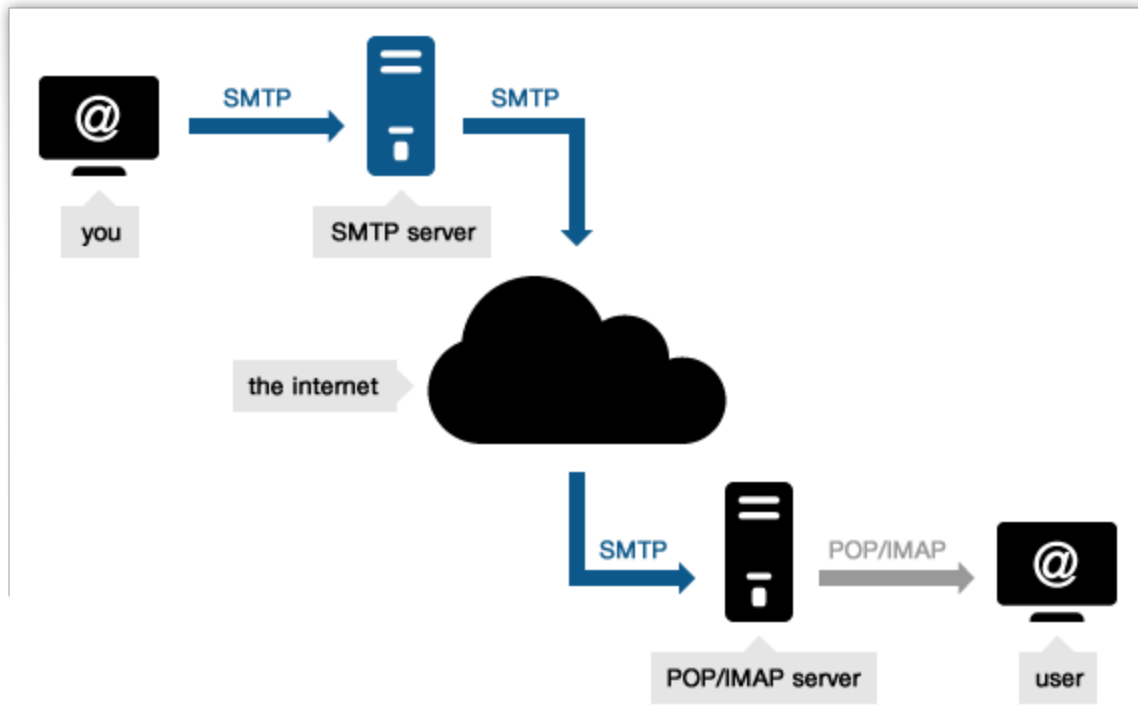
**Transport Layer:** The transport layer is responsible for the smooth transmission of data from one end to the other. It is also responsible for reliable connectivity, error recovery, and flow control of the data.

**Internet Layer:** This Internet Layer moves packets from source to destination by connecting independent networks.

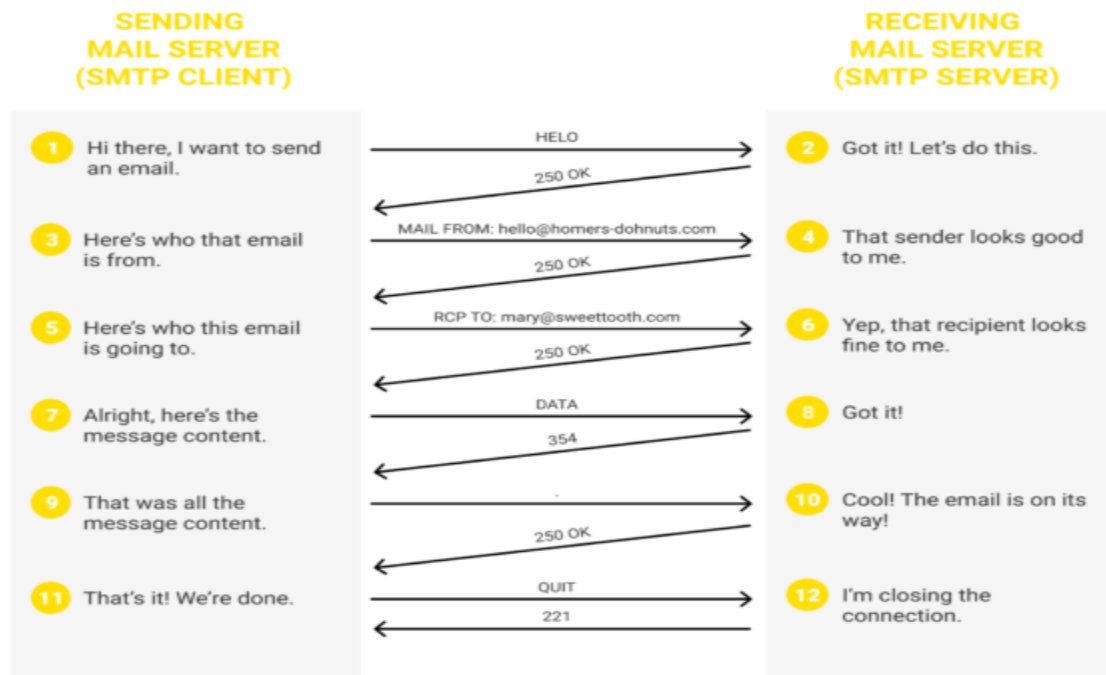
**Network Access Layer:** The Network Access Layer sees how a computer connects to a network.

## 2. SMTP(Simple Mail Transfer Protocol):

- SMTP, which stands for Simple Mail Transfer Protocol, is an email protocol used for sending email messages from one email account to another via the internet.
- Email protocols are sets of rules that let different email clients and accounts easily exchange information, and SMTP is one of the most common ones alongside POP and IMAP.



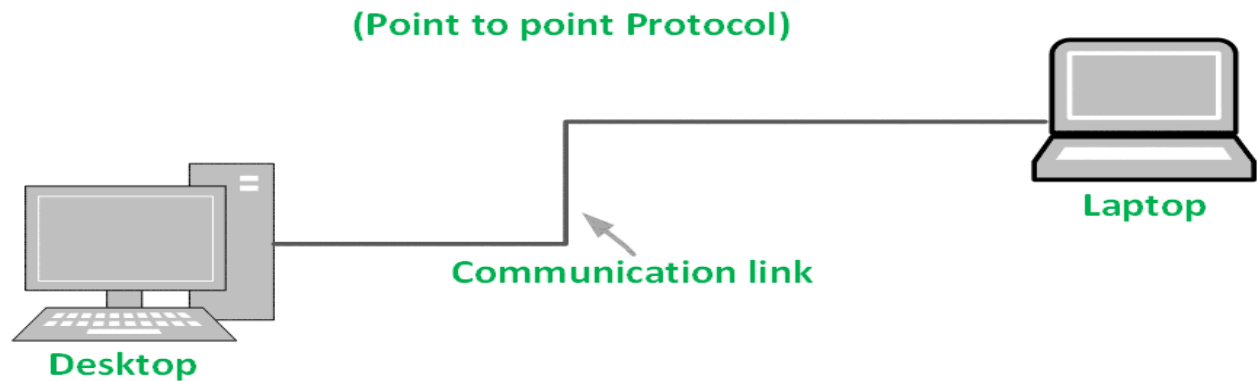
- Most email clients—including [Outlook](#), [Apple Mail](#), Gmail, and Yahoo Mail—rely on SMTP to “push” or send messages from a sender to a recipient.





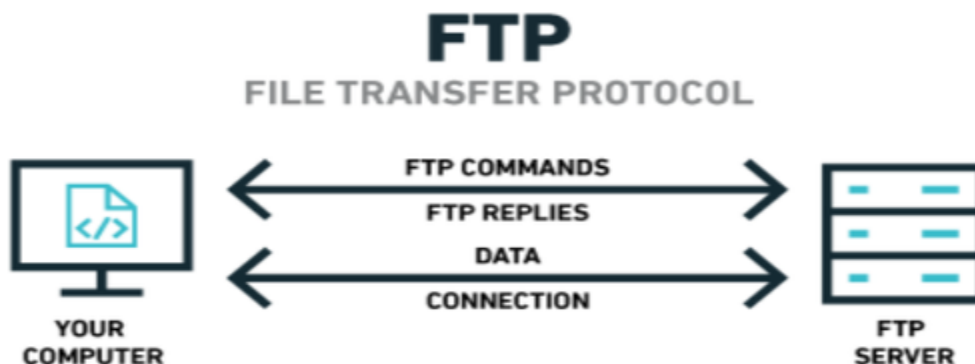
### 3. PPP(Point-to-Point Protocol):

- It is a communication protocol that is used to create a direct connection between two communicating devices.
- This protocol defines the rules using which two devices will authenticate with each other and exchange information with each other.
- For example, A user connects his PC to the server of an Internet Service Provider and also uses PPP.
- Similarly, for connecting two routers for direct communication, it uses PPP.



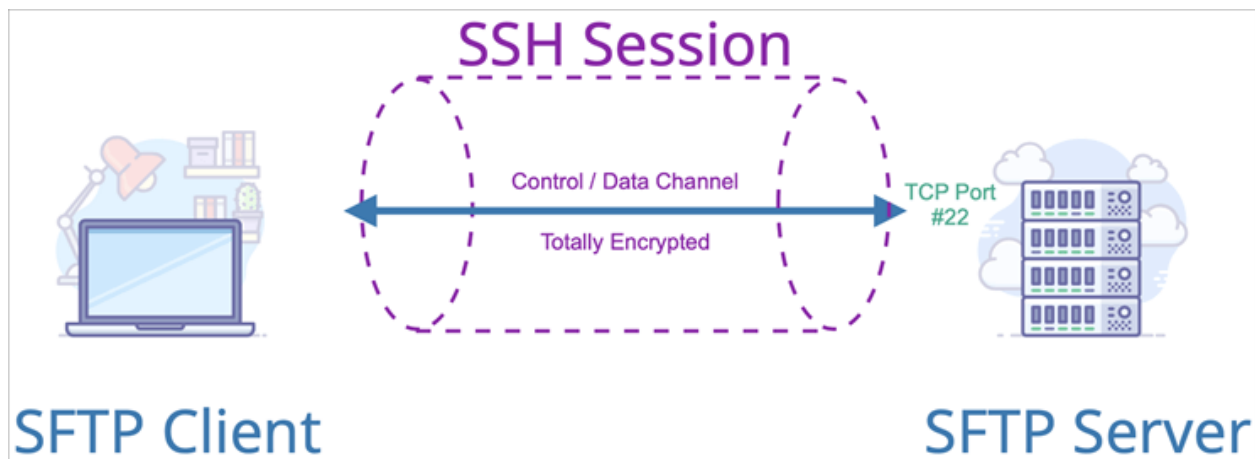
### 4. FTP (File Transfer Protocol):

- This protocol is used for transferring files from one system to the other.
- This works on a client-server model. When a machine requests for file transfer from another machine, the FTO sets up a connection between the two and authenticates each other using their ID and Password. And, the desired file transfer takes place between the machines.



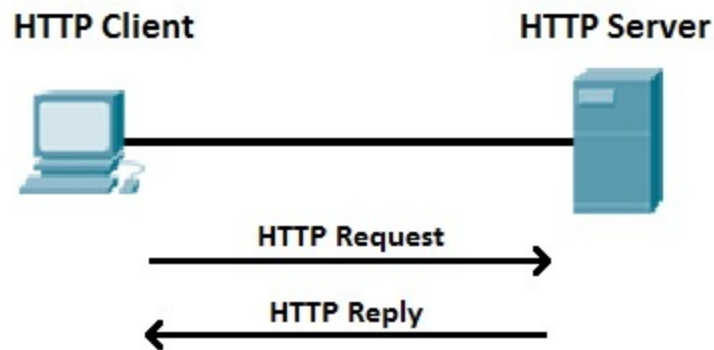
## 5. SFTP(Secure File Transfer Protocol):

- SFTP which is also known as SSH FTP refers to File Transfer Protocol (FTP) over Secure Shell (SSH) as it encrypts both commands and data while in transmission.
- SFTP acts as an extension to SSH and encrypts files and data, then sends them over a secure shell data stream.
- This protocol is used to remotely connect to other systems while executing commands from the command line.



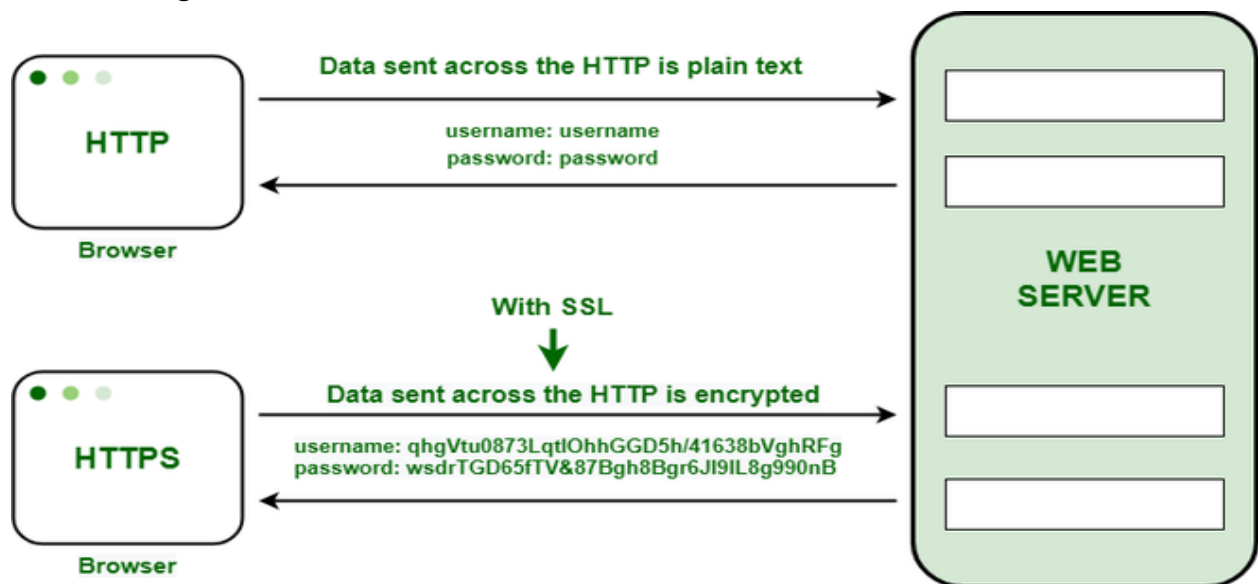
## 6. HTTP (Hypertext Transfer Protocol):

- This protocol is used to transfer hypertext over the internet, and it is defined by the WWW(World Wide Web) for information transfer.
- This protocol defines how the information needs to be formatted and transmitted. And, it also defines the various actions the web browsers should take in response to the calls made to access a particular web page.
- Whenever a user opens their web browser, the user will indirectly use HTTP as this is the protocol that is being used to share text, images, and other multimedia files on the World Wide Web.



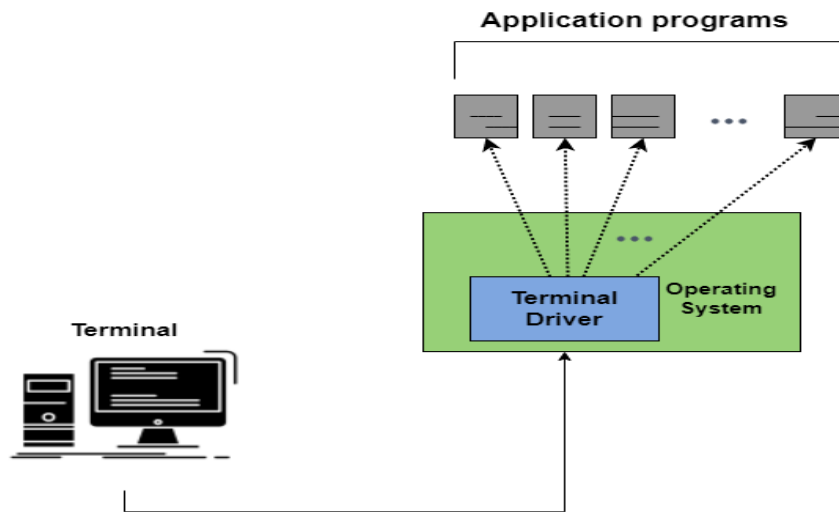
## 7. HTTPS (Hyper Text Transfer Protocol Secure):

- HTTPS is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network with the SSL/TLS protocol for encryption and authentication.
- So, generally, a website has an HTTP protocol but if the website is such that it receives some sensitive information such as credit card details, debit card details, OTP, etc then it requires an SSL certificate installed to make the website more secure.
- So, before entering any sensitive information on a website, we should check if the link is HTTPS or not. If it is not HTTPS, then it may not be secure enough to enter sensitive information.



## 8. TELNET(Terminal Network):

- TELNET is a standard TCP/IP protocol used for virtual terminal service given by ISO. This enables one local machine to connect with another.
- The computer which is being connected is called a remote computer and which is connecting is called the local computer.
- TELNET operation lets us display anything being performed on the remote computer in the local computer.
- This operates on the client/server principle. The local computer uses the telnet client program whereas the remote computer uses the telnet server program.



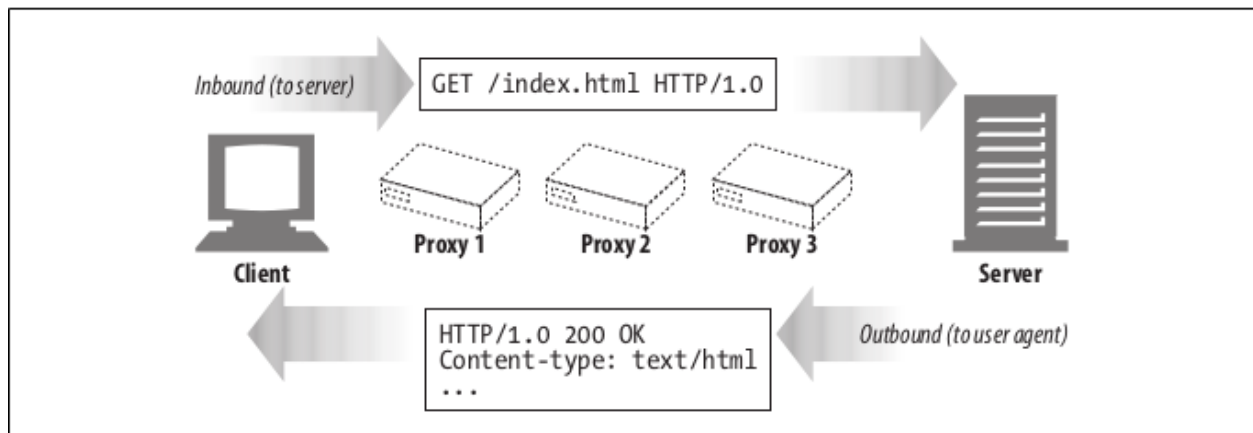
## 9. POP3(Post Office Protocol 3):

- POP3 stands for Post Office Protocol version 3. It has two Message Access Agents (MAAs) where one is client MAA (Message Access Agent) and another is server MAA(Message Access Agent) for accessing the messages from the mailbox.
- This protocol helps us to retrieve and manage emails from the mailbox on the receiver mail server to the receiver's computer. This is implied between the receiver and receiver mail server.



# HTTP Messages

- HTTP messages are how data is exchanged between a server and a client.
- There are two types of messages:
  - HTTP Request Message: **sent by the client to trigger an action on the server**
  - HTTP Response Message: **the answer from the server**



- HTTP messages are composed of textual information encoded in ASCII, and span over multiple lines.
- In HTTP/1.1, and earlier versions of the protocol, these messages were openly sent across the connection.
- In HTTP/2, the once human-readable message is now divided up into HTTP frames, providing optimization and performance improvements.
- Web developers, or webmasters, rarely craft these textual HTTP messages themselves: software, a Web browser, proxy, or Web server, perform this action.
- They provide HTTP messages through config files (for proxies or servers), APIs (for browsers), or other interfaces.

## Three parts of an HTTP message:

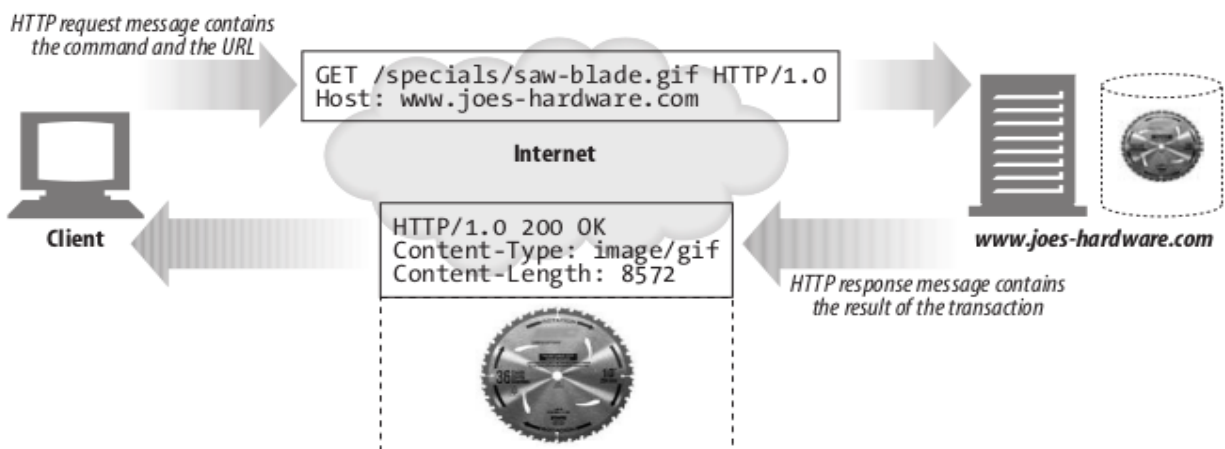


HTTP requests, and responses, share a similar structure and are composed of:

1. A *start-line* describing the requests to be implemented, or its status of whether successful or a failure. This start-line is always a single line.
2. An optional set of *HTTP headers* specifying the request, or describing the body included in the message.
3. A blank line indicating all meta information for the request has been sent.
4. An optional *body* containing data associated with the request (like the content of an HTML form), or the document associated with a response. The presence of the body and its size is specified by the start-line and HTTP headers.

The start-line and HTTP headers of the HTTP message are collectively known as the *head* of the requests, whereas its payload is known as the *body*.

## Example of HTTP Request and Response Message:



# HTTP Request Message

- Request messages ask servers to do something to a resource.
- The start line for a request message, or request line, contains a method describing what operation the server should perform and a request URL describing the resource on which to perform the method.
- The request line also includes an HTTP version which tells the server what dialect of HTTP the client is speaking.
- Message Syntax:

**<method><request-URL> <version>      //start line**

**<headers>**

**<entity-body>**

- Example:

```
GET /test/hi-there.txt HTTP/1.1
```

```
Accept: text/*
```

```
Host: www.joes-hardware.com
```

## Start line

HTTP requests are messages sent by the client to initiate an action on the server. Their *start-line* contains three elements:

1. An *HTTP method*, a verb (like GET, PUT, or POST) or a noun (like HEAD or OPTIONS), that describes the action to be performed. For example, GET indicates that a resource should be fetched or POST means that data is pushed to the server (creating or modifying a resource, or generating a temporary document to send back).

- Common HTTP methods:

Method	Description	Message body?
GET	Get a document from the server.	No
HEAD	Get just the headers for a document from the server.	No
POST	Send data to the server for processing.	Yes
PUT	Store the body of the request on the server.	Yes
TRACE	Trace the message through proxy servers to the server.	No
OPTIONS	Determine what methods can operate on a server.	No
DELETE	Remove a document from the server.	No

- For example, in the line “GET /specials/saw-blade.gif HTTP/1.0,” the method is GET.
2. The *request-target*, usually a URL, or the absolute path of the protocol, port, and domain are usually characterized by the request context. The format of this request target varies between different HTTP methods.
  3. The *HTTP version*, which defines the structure of the remaining message, acts as an indicator of the expected version to use for the response.
    - Example: HTTP/1.0

## Headers

- HTTP header fields add additional information to request and response messages.
- They are basically just lists of name/value pairs.
- For example, the following header line assigns the value 19 to the Content-Length header field:
  - Content-length: 19



## Header classifications

- General headers
  - Can appear in both request and response messages
- Request headers
  - Provide more information about the request
- Response headers
  - Provide more information about the response
- Entity headers
  - Describe body size and contents, or the resource itself
- Extension headers
  - New headers that are not defined in the specification
- **Common header examples**

Header example	Description
Date: Tue, 3 Oct 1997 02:16:03 GMT	The date the server generated the response
Content-length: 15040	The entity body contains 15,040 bytes of data
Content-type: image/gif	The entity body is a GIF image
Accept: image/gif, image/jpeg, text/html	The client accepts GIF and JPEG images and HTML

## Entity Bodies

- The third part of an HTTP message is the optional entity body.
- Entity bodies are the payload of HTTP messages.
- They are the things that HTTP was designed to transport.
- HTTP messages can carry many kinds of digital data: images, video, HTML documents, software applications, credit card transactions, electronic mail, and so on.

Bodies can be broadly divided into two categories:

- Single-resource bodies, consisting of one single file, are defined by the two headers: Content-Type and Content-Length.
- Multiple-resource bodies, consist of a multipart body, each containing a different bit of information. This is typically associated with HTML Forms.

# HTTP Response Message

- Response messages carry status information and any resulting data from an operation back to a client.
- The start line for a **response message, or response line**, contains the **HTTP version** that the response message is using, a numeric **status code**, and a **textual reason phrase** describing the status of the operation.
- Message Syntax:

**<version> <status> <reason-phrase>**

**<headers>**

**<entity-body>**

- Example:

<b>Start line</b>	HTTP/1.0 200 OK
<b>Headers</b>	Content-type: text/plain Content-length: 19
<b>Body</b>	Hi! I'm a message!

## Status line

The start line of an HTTP response, called the *status line*, contains the following information:

1. The *protocol version*, usually HTTP/1.1.
2. A *status code*, indicating success or failure of the request. Common status codes are 200, 404, or 302

- **Status codes**

Overall range	Defined range	Category
100-199	100-101	Informational
200-299	200-206	Successful
300-399	300-305	Redirection
400-499	400-415	Client error
500-599	500-505	Server error

- **Common status codes**

Status code	Reason phrase	Meaning
200	OK	Success! Any requested data is in the response body.
401	Unauthorized	You need to enter a username and password.
404	Not Found	The server cannot find a resource for the requested URL.

3. *A status text.* A brief, purely informational, textual description of the status code to help a human understand the HTTP message.
  - It provides a textual explanation of the status code.
  - For example, in the line “HTTP/1.0 200 OK,” the reason phrase is OK.
  - Reason phrases are paired one-to-one with status codes.
  - The reason phrase provides a human-readable version of the status code that application developers can pass along to their users to indicate what happened during the request.

A typical status line looks like: HTTP/1.1 404 Not Found.

# Headers

- HTTP header fields add additional information to request and response messages.
- They are basically just lists of name/value pairs.
- For example, the following header line assigns the value 19 to the Content-Length header field:
  - Content-length: 19

## Header classifications

- General headers
  - Can appear in both request and response messages
- Request headers
  - Provide more information about the request
- Response headers
  - Provide more information about the response
- Entity headers
  - Describe body size and contents, or the resource itself
- Extension headers
  - New headers that are not defined in the specification

## ● Common header examples

Header example	Description
Date: Tue, 3 Oct 1997 02:16:03 GMT	The date the server generated the response
Content-length: 15040	The entity body contains 15,040 bytes of data
Content-type: image/gif	The entity body is a GIF image
Accept: image/gif, image/jpeg, text/html	The client accepts GIF and JPEG images and HTML

- **Entity Bodies**

- The third part of an HTTP message is the optional entity body.
- Entity bodies are the payload of HTTP messages.
- They are the things that HTTP was designed to transport.
- HTTP messages can carry many kinds of digital data: images, video, HTML documents, software applications, credit card transactions, electronic mail, and so on.

Bodies can be broadly divided into three categories:

- Single-resource bodies, consisting of a single file of known length, defined by the two headers: Content-Type and Content-Length.
- Single-resource bodies, consisting of a single file of unknown length, encoded by chunks with Transfer-Encoding set to chunked.
- Multiple-resource bodies, consisting of a multipart body, each containing a different section of information. These are relatively rare.

