What is a protocol?

A Protocol is a set of rules that we use for specific purposes. In the current scenario, when we are talking about *protocols,* it is about communication- the way we talk to each other. For instance, a newsreader speaks in English and because you understand English, you are able to understand. English is the protocol. The moment the newsreader starts speaking in a language that you don't understand, the protocol beats its purpose. Thus, we need both the parties to agree to a set of rules for the communication to take place. The protocol, in this case, is for communication.

Now, talking about the web, in particular, multiple protocols are used to communicate. Primarily for end users, the most important and visible protocols are HTTP and HTTPS. Though there are many other protocols as well, HTTP and HTTPS protocols cater to most of the population.

What is HTTP?

HTTP is Hypertext transfer protocol. Simply put - Rules to sending and receiving text-based messages. As we all know, computers work in a language of 1's and 0's i.e. Binary language. Therefore, potentially every set of 1's and 0's construct something, it could be a word.

Let's say I want to write 'a'. Now, if 0 stands for 'a', 1 stands for 'b', and 01 stands for 'c', I can infer that a combination of 0's and 1's can construct a word as well. In this case, the text is already constructed and is being sent on the wire. The computer works on many languages - pure binary, text and some other formats like byte codes. Here, what is being transferred is text. I am emphasizing on 'text' because this text is interpreted by the browser and the moment browser interprets it, it becomes hypertext, and the protocol that transfers the text is referred to as *hypertext transfer protocol - HTTP.*

Using HTTP, you can definitely transfer images and text and even sound, but not videos.

What is HTTPS?

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms.

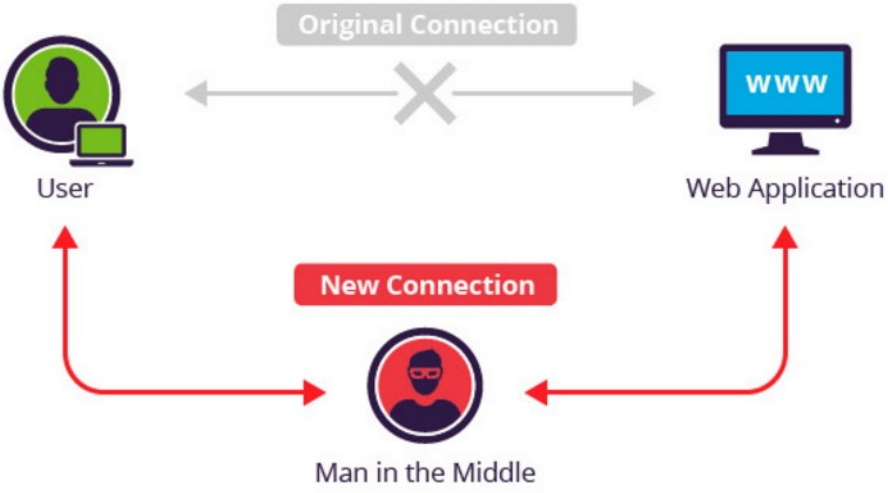
What is the importance of HTTPS?

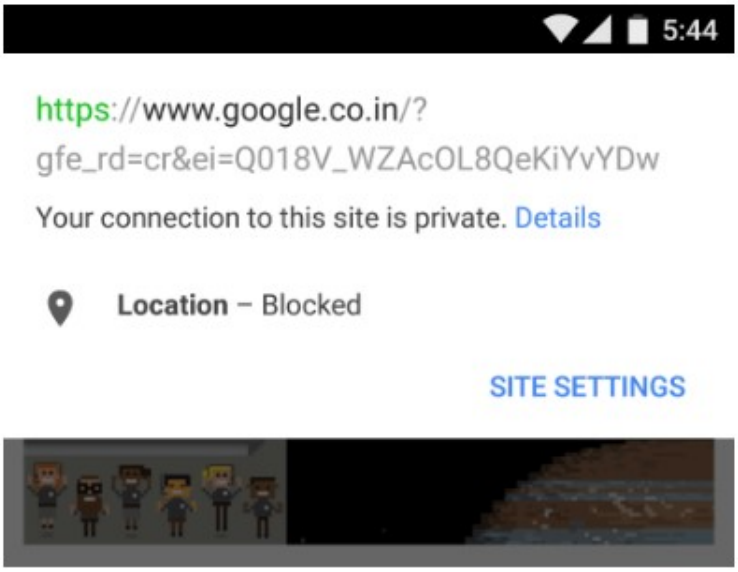
We agreed upon the fact that what is being transferred from one point to another is text. To understand why HTTPS protocol, we first should know how wi-fi routers function. Let's say you are at an airport and you are connecting to the wi-fi which is the property of a third party. Now, when you are communicating over HTTP, the text is being transferred by their router. And if I go to a low version of the router, I can comfortably check and read the text that is being transferred. There could be a password that I can use to login to your bank site and do a fraudulent

transaction! Point being - this is fundamentally insecure. This is called the *man in the middle attack.*

And this why do we need https when HTTP seems to suffice.

Now, to save our data from such attacks, we need to encrypt that data.





This is what HTTPS site looks like.

Encryption and Encryption Levels

Encryption is simple terms is a hiding information. There are various ways to do so. You must have heard these terms - 128 bit encrypt HTTPS and 64 bit encrypt HTTPS. 128-bit Encrypt is a high encryption technique and it's very difficult to decrypt (decode). In the case of HTTPS when the data is being transferred on the wires, the man in the middle may still know what is being transferred, but can not make sense out of it as the data is encrypted. Only the browser will decrypt it and show it, and the server will decrypt it and use it for transactions.

How does this happens when you request to open a site in a browser?

To understand this, let us imagine that there is one *Server that resides somewhere* serving all the request for one domain. Now, when I type xyz.com, it's a server that I am connecting to, taking data from and rendering it in the browser.

To simplify further, imagine a domain name google.com being broadcasted from one server. There resides one machine somewhere connected to the internet and the moment you say google.com in your browser, you connect to that machine, pick data from that machine and show it in the browser. If you have saved your picture, it gets uploaded to that machine. Now, if you want to see that picture, you go to google.com/show-me-my-picture, which transfers the picture from the machine to the browser to be shown to you.

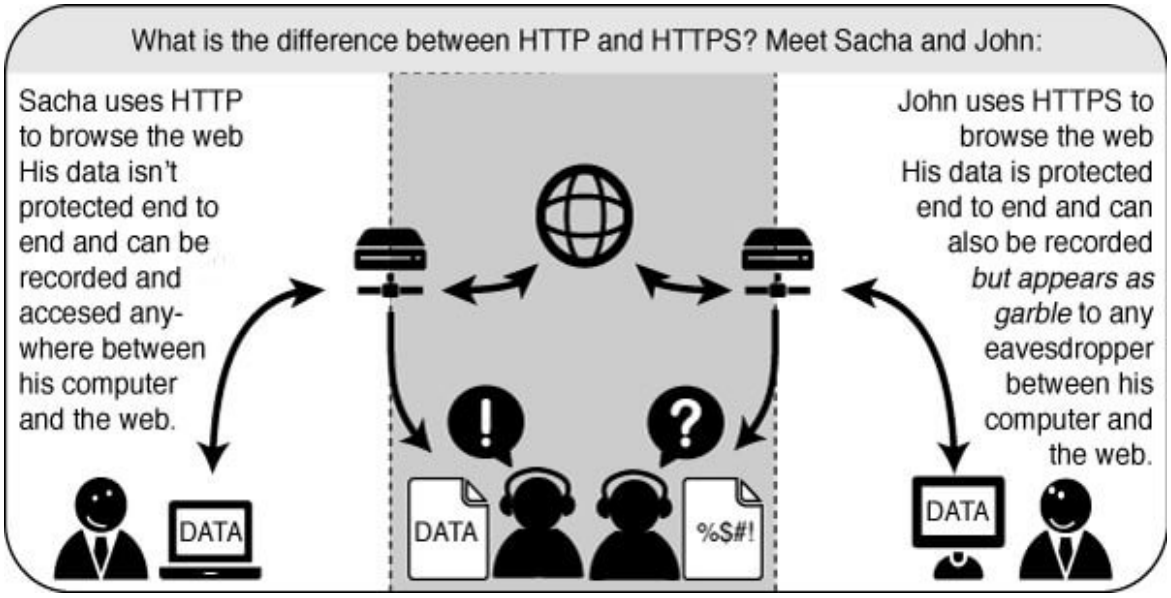
This process cannot be completed if I am not able to reach that particular machine. For this to happen, every machine has an address (the way we have a mobile number), it is called the IP address and every domain has an IP map. The moment you enter this user-friendly URL - google.com, it converts this username into IP and connects to the router to reach out to that particular service line associated with this URL. Once it reaches to the server, it raises a request of what is needed. It is represented as 'google.com/s=', helping the user understand the request made by him. As a result, the server gives him the results according to your request, which gets rendered to the browser.

What happens when a request for a website URL is made which is on HTTP protocol?

As the first step, it is the job of HTTP to find out the server and once the communication route is established, the server sends a text to the browser. This text could either be in its pure form or encrypted form, which is then rendered by the browser or used for whatever purpose it has to it has to be used.

As there should be a measurement of this difficulty quotient, we interpret that, higher the number of bits, more difficult it is to decrypt. However, it only increases the level of complexity making it very difficult to decrypt, but not impossible.

Deciding between HTTP and HTTPS



Source: http://en.flossmanuals.net/basic-internet-security/_all/

Anything and everything is personal. If you are searching for "How to install SSL Certificate", that search would be private to you, isn't it? Whether you are browsing or looking for a product, reading an article, you generally do not want others to know about it. As an end user, I would want to keep it as private. There are things I might not want to keep private and for those, I can use HTTP. However, for personal information, banks and transnational information, HTTPS has become a standard.

HTTPS sounds great. What else should you know about it?

There is no denying to the fact that privacy has a cost to it. There are a couple of *cons*-

1. HTTPS requests take more time to process.

2. Because it needs more time to process, it needs more hardware - the server that you are utilizing. This also means additional cost

Whereas, for HTTP you use lesser energy as compared to HTTPS as the communication happens faster (without encryption and decryption).

- HTTP stands for **HyperText Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.

- This protocol is known as HyperText Transfer Protocol because of its efficiency 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:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.
- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

HTTP Request

HTTP request comprises of lines which contains:

- Request line
- Header Fields
- Message body

Key Points

- The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post.**
- The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

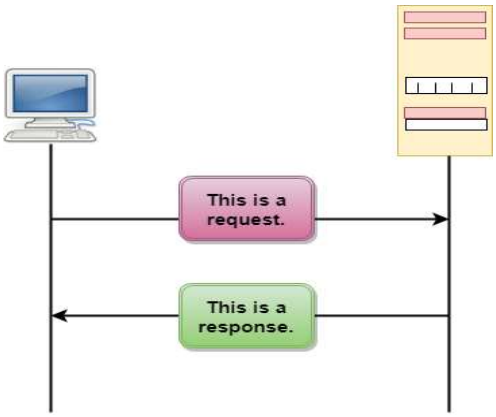
HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains:

- Status line

- Headers
- Message body

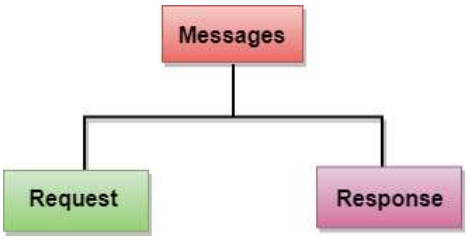
HTTP Transactions



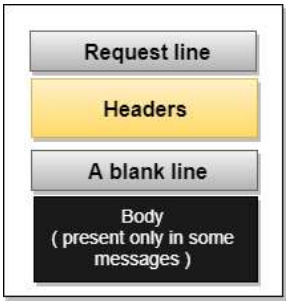
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages

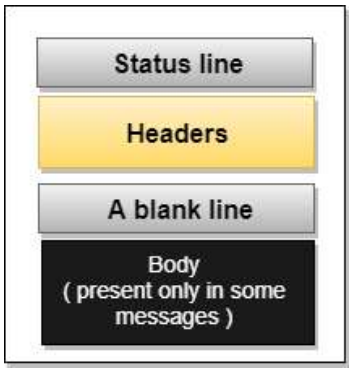
HTTP messages are of two types: request and response. Both the message types follow the same message format.



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)

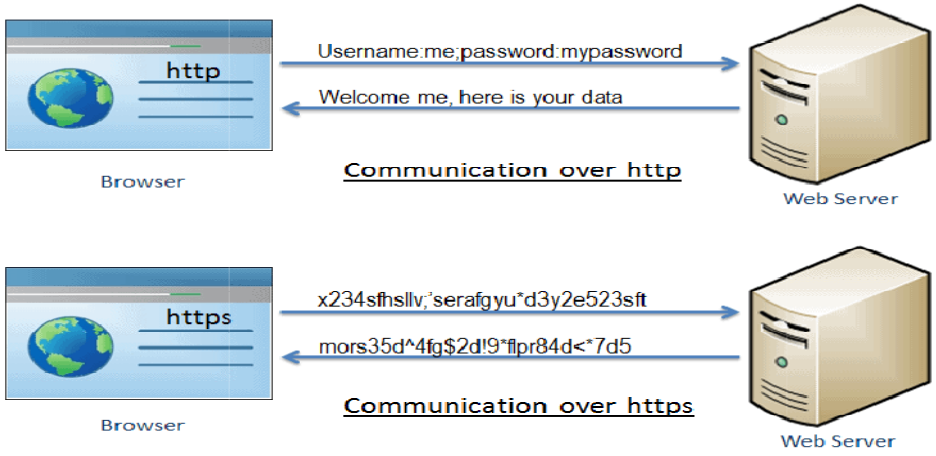
- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL 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 contain slashes that separate the directories from the subdirectories and files.

HTTPS

HTTPS stands for Hyper Text Transfer Protocol Secure. It is a protocol for securing the communication between two systems e.g. the browser and the web server. The following figure illustrates the difference between communication over http and https:



Communication over https and http

As you can see in the above figure, http transfers data between the browser and the web server in the hypertext format, whereas https transfers data in the encrypted format. Thus, https prevents hackers from reading and modifying the data during the transfer between the browser and the web server. Even if hackers manage to intercept the communication, they will not be able to use it because the message is encrypted.

HTTPS established an encrypted link between the browser and the web server using the Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols. TLS is the new version of SSL.

http vs https

http	https
Transfers data in hypertext (structured text) format	Transfers data in encrypted format
Uses port 80 by default	Uses port 443 by default
Not secure	Secured using SSL technology
Starts with http://	Starts with https://

Advantage of https

- **Secure Communication:** https makes a secure connection by establishing an encrypted link between the browser and the server or any two systems.
- **Data Integrity:** https provides data integrity by encrypting the data and so, even if hackers manage to trap the data, they cannot read or modify it.
- **Privacy and Security:** https protects the privacy and security of website users by preventing hackers to passively listen to communication between the browser and the server.
- **Faster Performance:** https increases the speed of data transfer compared to http by encrypting and reducing the size of the data.
- **SEO:** Use of https increases SEO ranking. In Google Chrome, Google shows the **Not Secure** label in the browser if users' data is collected over http.
- **Future:** https represents the future of the web by making internet safe for users and website owners.

FTP

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- 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.

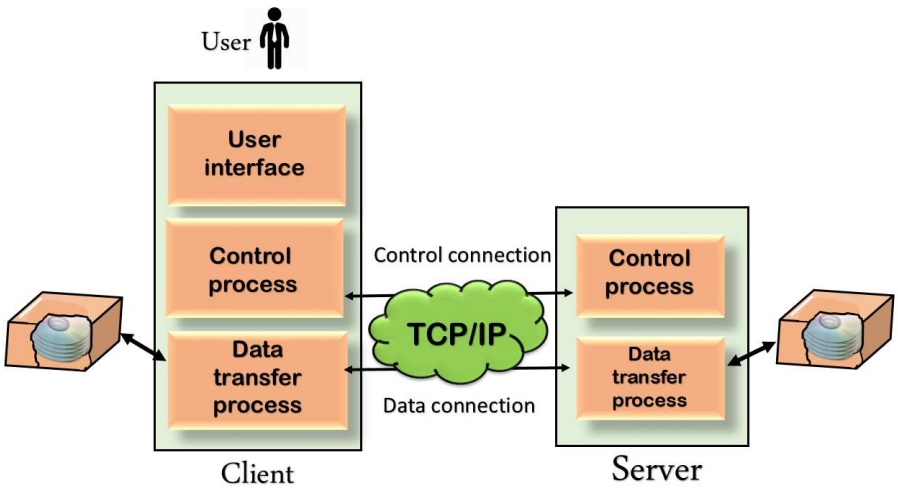
Objectives of FTP

- 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?

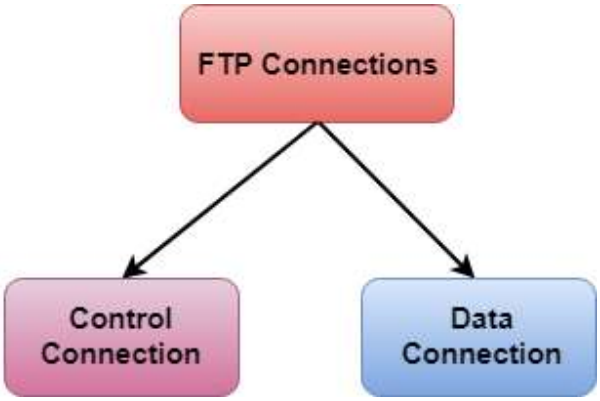
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. 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:



- **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.

Advantages of FTP:

- **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way 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.

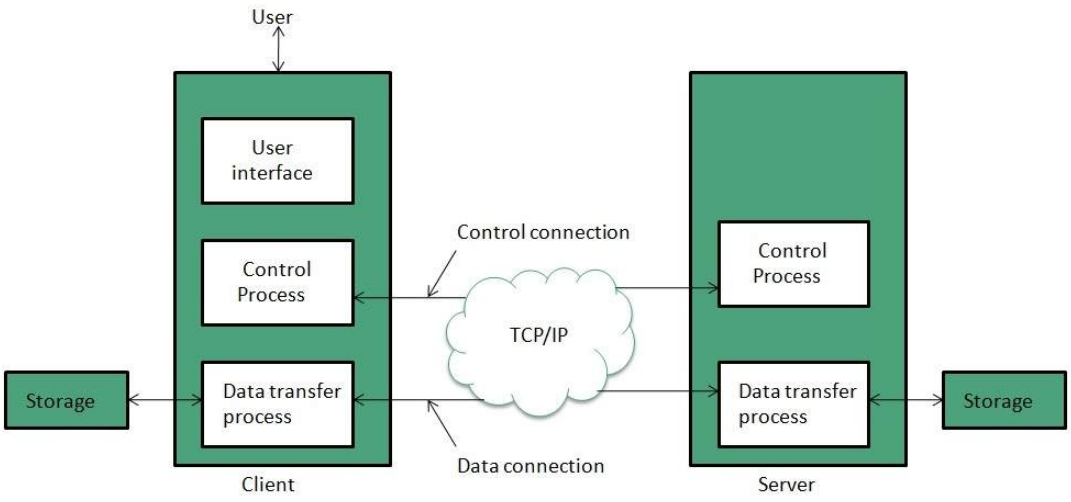
Disadvantages of FTP:

- 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 provides 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.

File Transfer Protocol (FTP)

FTP is used to copy files from one host to another. FTP offers the mechanism for the same in following manner:

- FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.
- FTP establishes two different connections: one is for data transfer and other is for control information.
- **Control connection** is made between **control processes** while **Data Connection** is made between<="" b="" style="user-select: auto;">
- FTP uses **port 21** for the control connection and **Port 20** for the data connection.



Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol is also used to transfer the files but it transfers the files without authentication. Unlike FTP, TFTP does not separate control and data information. Since there is no authentication exists, TFTP lacks in security features therefore it is not recommended to use TFTP.

Key points

- TFTP makes use of UDP for data transport. Each TFTP message is carried in separate UDP datagram.
- The first two bytes of a TFTP message specify the type of message.
- The TFTP session is initiated when a TFTP client sends a request to upload or download a file.
- The request is sent from an ephemeral UDP port to the **UDP port 69** of an TFTP server.

Difference between FTP and TFTP

S.N.	Parameter	FTP	TFTP
1	Operation	Transferring Files	Transferring Files
2	Authentication	Yes	No
3	Protocol	TCP	UDP
4	Ports	21 – Control, 20 – Data	Port 3214, 69, 4012
5	Control and Data	Separated	Separated
6	Data Transfer	Reliable	Unreliable

E-mail Protocols

E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMPTP

SMTP stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.

- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

SMTP Commands

The following table describes some of the SMTP commands:

S.N.	Command Description
1	**HELLO** This command initiates the SMTP conversation.
2	**EHELLO** This is an alternative command to initiate the conversation. ESMTP indicates that the sender server wants to use extended SMTP protocol.
3	**MAIL FROM** This indicates the sender's address.
4	**RCPT TO** It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.
5	**SIZE** This command let the server know the size of attached message in bytes.
6	**DATA** The **DATA** command signifies that a stream of data will follow. Here stream of data refers to the body of the message.
7	**QUIT** This commands is used to terminate the SMTP connection.
8	**VERFY** This command is used by the receiving server in order to verify whether the given username is valid or not.
9	**EXPN** It is same as VRFY, except it will list all the users name when it used with a distribution list.

IMAP

IMAP stands for **Internet Message Access Protocol.** It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

Key Points:

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

IMAP Commands

The following table describes some of the IMAP commands:

S.N.	Command Description
1	**IMAP_LOGIN** This command opens the connection.
2	**CAPABILITY** This command requests for listing the capabilities that the server supports.
3	**NOOP** This command is used as a periodic poll for new messages or message status updates during a period of inactivity.
4	**SELECT** This command helps to select a mailbox to access the messages.
5	**EXAMINE** It is same as SELECT command except no change to the mailbox is permitted.
6	**CREATE** It is used to create mailbox with a specified name.
7	**DELETE** It is used to permanently delete a mailbox with a given name.
8	**RENAME** It is used to change the name of a mailbox.
9	**LOGOUT**

	This command informs the server that client is done with the session. The server must send BYE untagged response before the OK response and then close the network connection.

POP

POP stands for Post Office Protocol. It is generally used to support a single client. There are several versions of POP but the POP 3 is the current standard.

Key Points

- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.
- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.

POP Commands

The following table describes some of the POP commands:

S.N.	Command Description
1	**LOGIN** This command opens the connection.
2	**STAT** It is used to display number of messages currently in the mailbox.
3	**LIST** It is used to get the summary of messages where each message summary is shown.
4	**RETR** This command helps to select a mailbox to access the messages.
5	**DELE** It is used to delete a message.
6	**RSET** It is used to reset the session to its initial state.
7	**QUIT** It is used to log off the session.

Comparison between POP and IMAP

S.N.	POP	IMAP
1	Generally used to support single client.	Designed to handle multiple clients.
2	Messages are accessed offline.	Messages are accessed online although it also supports offline mode.
3	POP does not allow search facility.	It offers ability to search emails.
4	All the messages have to be downloaded.	It allows selective transfer of messages to the client.
5	Only one mailbox can be created on the server.	Multiple mailboxes can be created on the server.
6	Not suitable for accessing non-mail data.	Suitable for accessing non-mail data i.e. attachment.
7	POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.	IMAP commands are not abbreviated, they are full. Eg. STATUS.
8	It requires minimum use of server resources.	Clients are totally dependent on server.
9	Mails once downloaded cannot be accessed from some other location.	Allows mails to be accessed from multiple locations.
10	The e-mails are not downloaded automatically.	Users can view the headings and sender of e-mails and then decide to download.
10	POP requires less internet usage time.	IMAP requires more internet usage time.