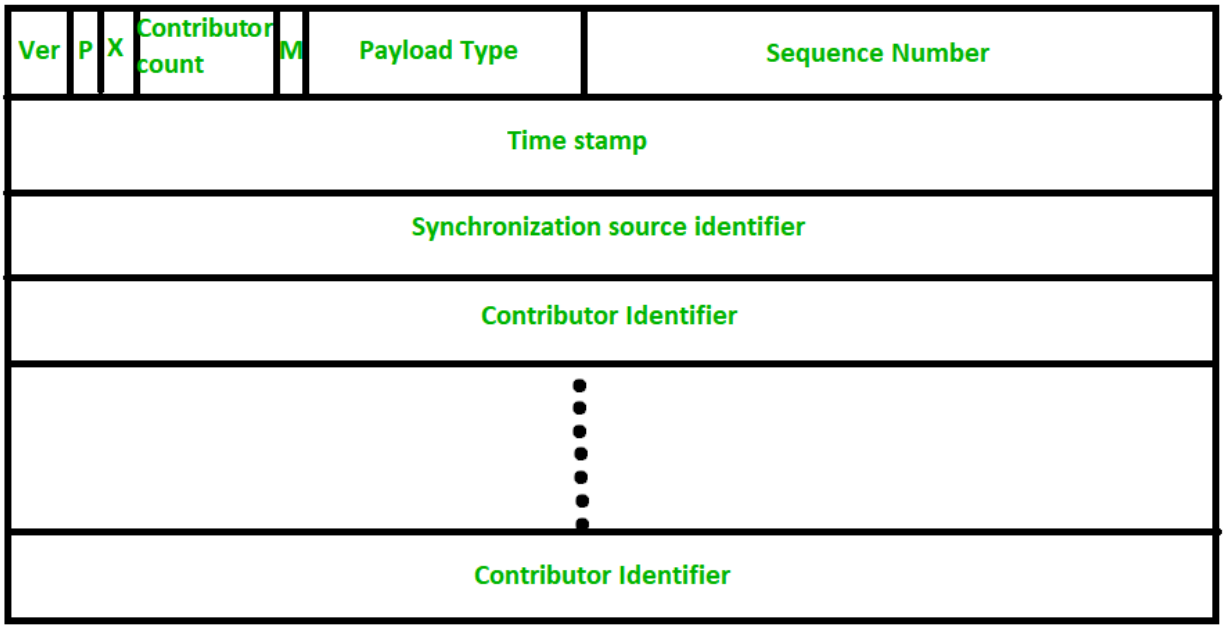
UNIT-5

Real Time Transport Protocol (RTP)

A protocol is designed to handle real-time traffic (like audio and video) of the Internet, is known as Real Time Transport Protocol (RTP). RTP must be used with UDP. It does not have any delivery mechanism like multicasting or port numbers. RTP supports different formats of files like MPEG and MJPEG. It is very sensitive to packet delays and less sensitive to packet loss.



The header format of RTP is very simple and it covers all real-time applications. The explanation of each field of header format is given below:

1. Version: This 2-bit field defines version number. The current version is 2.
2. P – The length of this field is 1-bit. If value is 1, then it denotes presence of padding at end of packet and if value is 0, then there is no padding.
3. X – The length of this field is also 1-bit. If value of this field is set to 1, then its indicates an extra extension header between data and basic header and if value is 0 then, there is no extra extension.
4. Contributor count – This 4-bit field indicates number of contributors. Here maximum possible number of contributor is 15 as a 4-bit field can allows number from 0 to 15.
5. M – The length of this field is 1-bit and it is used as end marker by application to indicate end of its data.
6. Payload types – This field is of length 7-bit to indicate type of payload. We list applications of some common types of payload.
7. Sequence Number – The length of this field is 16 bits. It is used to give serial numbers to RTP packets. It helps in sequencing. The sequence number for first packet is given a random number and then every next packet's sequence number is incremented by 1. This field mainly helps in checking lost packets and order mismatch.
8. Time Stamp – The length of this field is 32-bit. It is used to find relationship between times of different RTP packets. The timestamp for first packet is given randomly and then time stamp for next packets given by sum of previous timestamp and time taken to produce first byte of current packet. The value of 1 clock tick is varying from application to application.
9. Synchronization Source Identifier – This is a 32-bit field used to identify and define the source. The value for this source identifier is a random number that is chosen by source itself.

This mainly helps in solving conflict arises when two sources started with the same sequencing number.

10. Contributor Identifier – This is also a 32-bit field used for source identification where there is more than one source present in session. The mixer source use Synchronization source identifier and other remaining sources (maximum 15) use Contributor identifier.

TRANSMISSION CONTROL PROTOCOL

Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.

Features of TCP protocol

The following are the features of a TCP protocol:

Reliable

TCP is a reliable protocol as it follows the flow and error control mechanism. It also supports the acknowledgment mechanism, which checks the state and sound arrival of the data. In the acknowledgment mechanism, the receiver sends either positive or negative acknowledgment to the sender so that the sender can get to know whether the data packet has been received or needs to resend.

Order of the data is maintained

This protocol ensures that the data reaches the intended receiver in the same order in which it is sent. It orders and numbers each segment so that the TCP layer on the destination side can reassemble them based on their ordering.

Connection-oriented

It is a connection-oriented service that means the data exchange occurs only after the connection establishment. When the data transfer is completed, then the connection will get terminated.

Full duplex

It is a full-duplex means that the data can transfer in both directions at the same time.

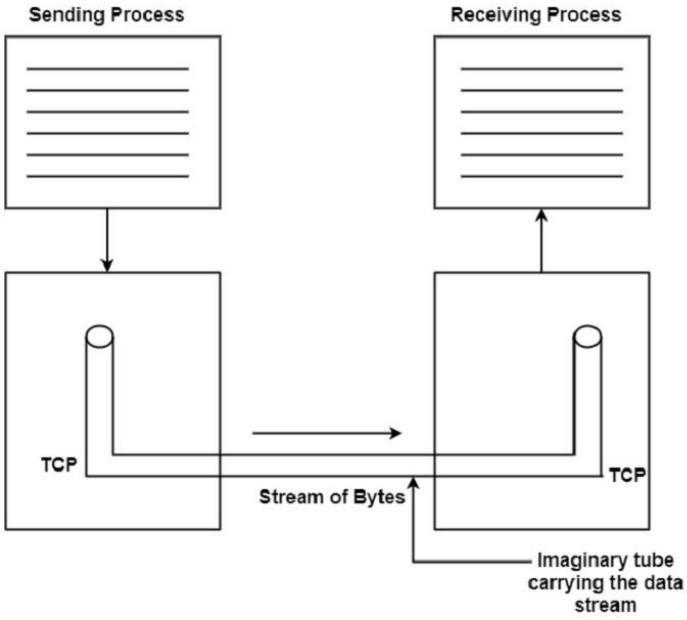
Stream-oriented

TCP is a stream-oriented protocol as it allows the sender to send the data in the form of a stream of bytes and also allows the receiver to accept the data in the form of a stream of bytes. TCP creates an environment in which both the sender and receiver are connected by an imaginary tube known as a virtual circuit. This virtual circuit carries the stream of bytes across the internet.

TCP Services in the Computer Network.

1. Stream Delivery Service

TCP is a stream-oriented protocol. It enables the sending process to deliver data as a stream of bytes and the receiving process to acquire data as a stream of bytes. TCP creates a working environment so that the sending and receiving procedures are connected by an imaginary "tube", as shown in the figure below:



2. Sending and Receiving Buffers

The sending and receiving processes cannot produce and receive data at the same speed. Hence, TCP needs a buffer for storage.

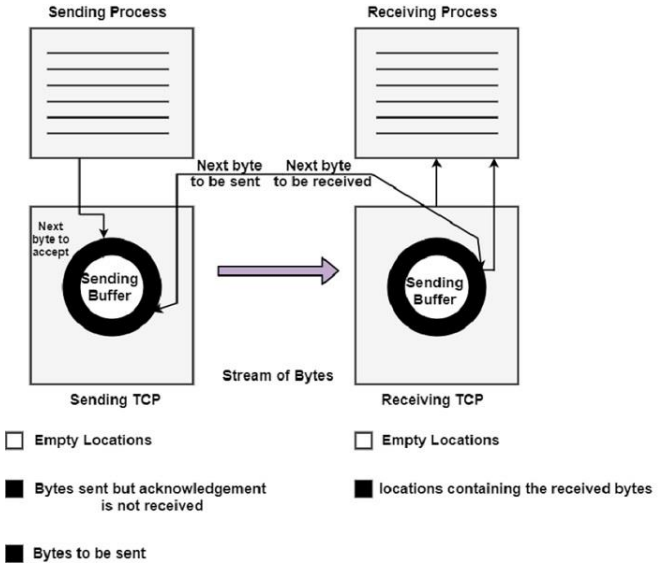
There are two methods of buffers used in each dissection, which are as follows:

- Sending Buffer
- Receiving Buffer

A buffer can be implemented by using a circular array of 1-byte location, as shown in the figure below. The figure shows the movement of data in one direction on the sending side.

The buffer has three types of locations, which are as follows:

- Empty Locations.
- Locations that contain the bytes which have been sent, but not acknowledged. These bytes are kept in the buffer till an acknowledgment is received.
- The location that contains the bytes which are to be sent by the sending TCP



In practice, the TCP may send only a port of data due to the slowness of the receiving process or congestion in the network.

The buffer at the receiver is divided into two parts as mentioned below:

- The part was containing empty locations.
- The part was containing the received bytes, which the sending process can consume.

3. Bytes and Segments

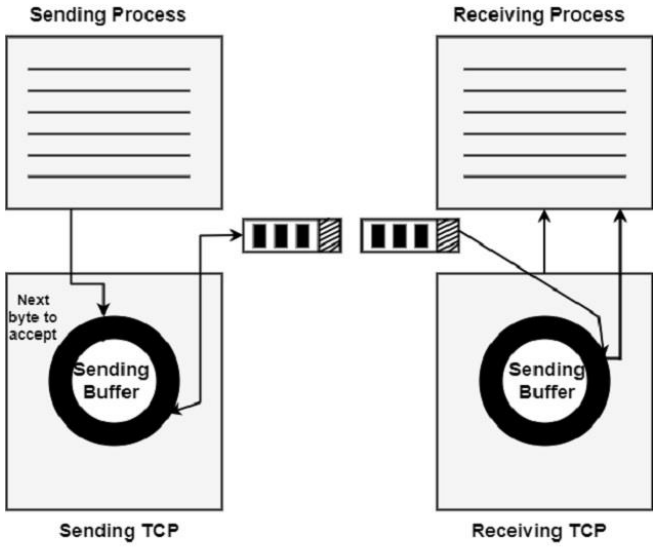
Buffering is used to handle the difference between the speed of data transmission and data consumption. But only buffering is not enough.

We need one more step before sending the data on the Internet Protocol (IP) layer as a TCP service provider. It needs to send data in the form of packets and not as a stream of bytes.

At the transport layer, TCP groups several bytes into a packet and this is called a segment. A header is added to each segment to exercise control.

The segment is encapsulated in an IP diagram and then transmitted. The entire operation is transparent to the receiving process. The segment may be deceived out of order, lost or corrupted when it receives the receiving end.

The figure given below shows how the segments are created from the bytes in the buffers:



The segments are not of the same size. Each segment can carry hundreds of bytes.

4. Full-Duplex Service

TCP offers a full-duplex service where the data can flow in both directions simultaneously. Each TCP will then have a sending buffer and receiving buffer. The TCP segments are sent in both directions.

Connection-Oriented Service

We are already aware that the TCP is a connection-oriented protocol. When a process wants to communicate (send and receive) with another process (process -2), the sequence of operations is as follows:

- TCP of process-1 informs TCP of process-2 and gets its approval.

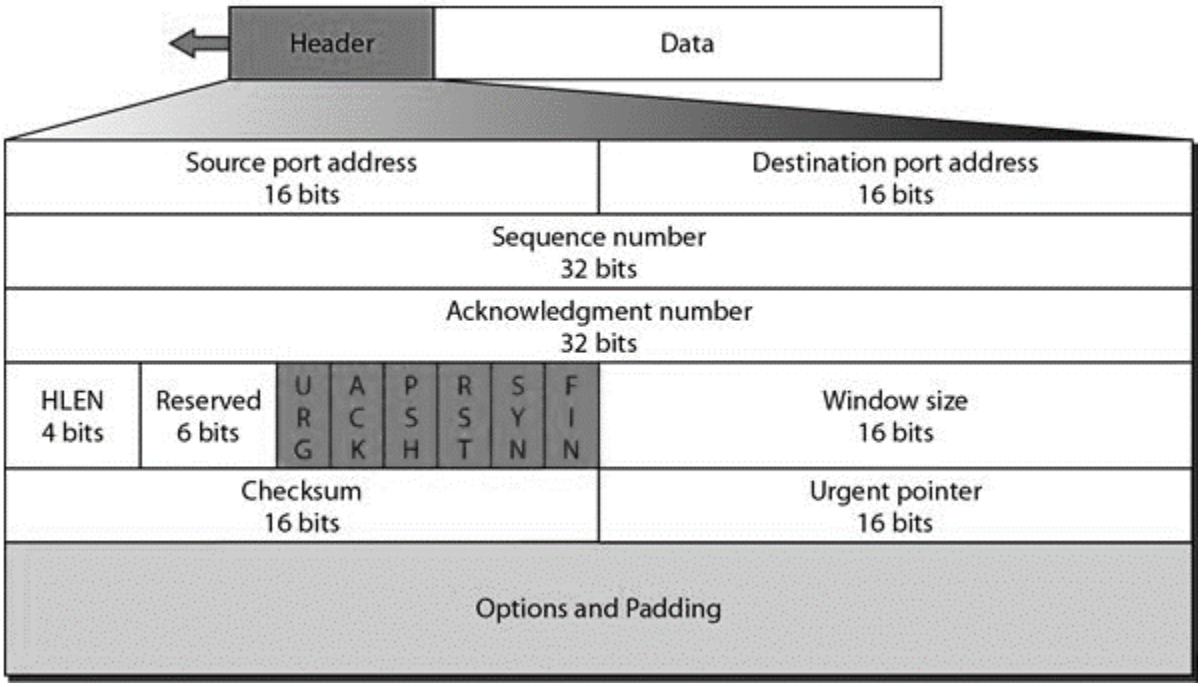
- TCP of process-1 tells TCP of process-2 exchange data in both directions.
- After completing the data exchange, when buffers on both sides are empty, the two TCPs destroy their buffers

The type of connection in TCP is not physical, but it is virtual. The TCP segment encapsulated in an IP datagram can be sent out of order. These segments can get lost or corrupted and may have to be resend. Each segment may take a different path to reach the destination.

5. **Reliable Service**

TCP is a reliable transport protocol. It uses an acknowledgment mechanism for checking the safe and sound arrival of data.

TCP Segment Header Format



Source port: It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.

Destination port: It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.

Sequence number: This field contains the sequence number of data bytes in a particular session.

Acknowledgment number: When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.

HLEN: It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.

Reserved: It is a 4-bit field reserved for future use, and by default, all are set to zero.

Flags

There are six control bits or flags:

URG: It represents an urgent pointer. If it is set, then the data is processed urgently.

ACK: If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.

PSH: If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.

RST: If it is set, then it requests to restart a connection.

SYN: It is used to establish a connection between the hosts.

FIN: It is used to release a connection, and no further data exchange will happen.

Window size

It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.

Checksum

It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.

Urgent pointer

It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.

Options

It provides additional options. The optional field is represented in 32-bits. If this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

TCP Connection Management

TCP communication works in Server/Client model. The client initiates the connection and the server either accepts or rejects it. Three-way handshaking is used for connection management.

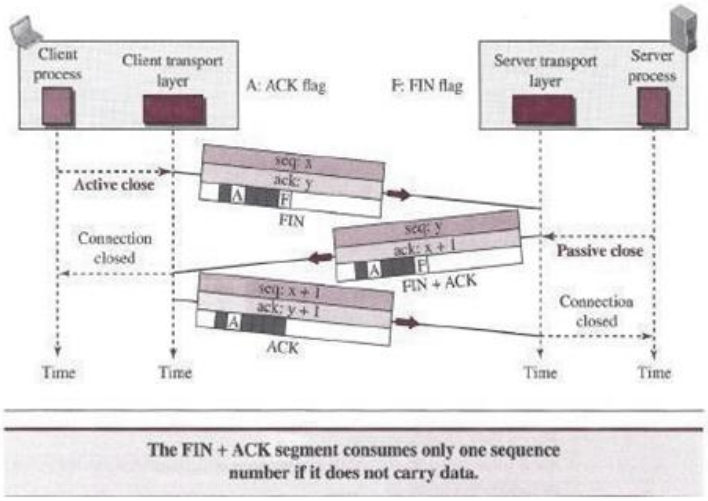
Three- Way Handshaking

The connection establishment in TCP is called *three-way handshaking.* an application program, called the *client,* wants to make a connection with another application program, called the *server,* using TCP as the transport-layer protocol The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a *passive open.* Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.

The client program issues a request for an *active open.* A client that wishes to connect to an open server tells its TCP to connect to a particular server.

- A SYN segment cannot carry data, but it consumes one sequence number.
- A SYN + ACK segment cannot carry data, but it does consume one sequence number.

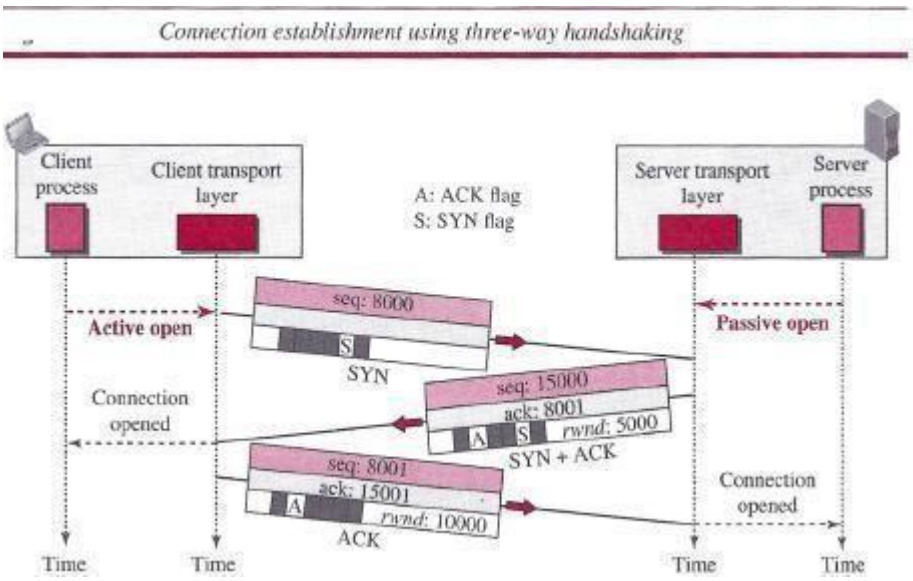
- An ACK segment, if carrying no data, consumes no sequence number.



TCP Connection Establishment

TCP is connection-oriented. a connection-oriented transport protocol establishes a logical path between the source and destination. All of the segments belonging to a message are then sent over this logical path. TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

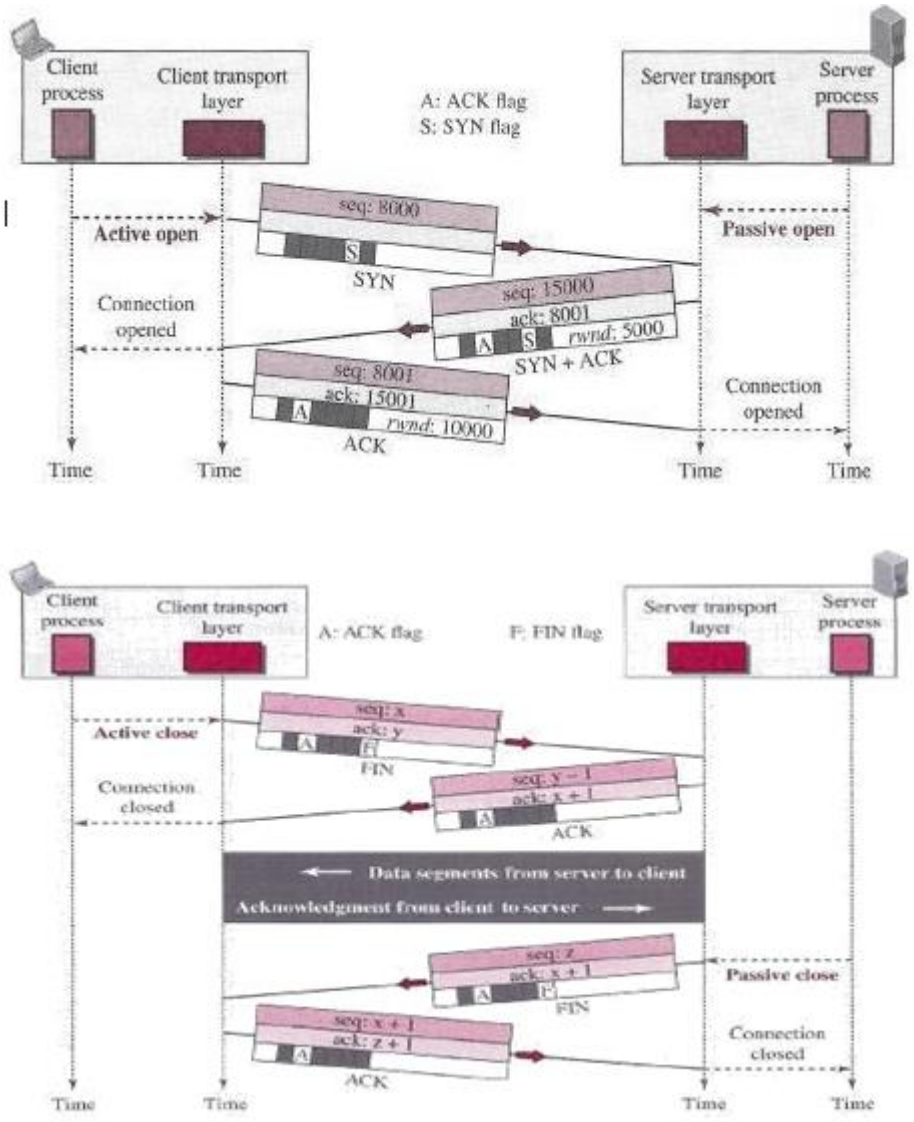
Data transfer:



TCP Connection Release

- Although TCP connections are full duplex, to understand how connections are released it is best to think of them as a pair of simplex connections.
- Each simplex connection is released independently of its sibling. To release a connection, either party can send a TCP segment with the *FIN* bit set, which means that it has no more data to transmit.

- When the *FIN* is acknowledged, that direction is shut down for new data. Data may continue to flow indefinitely in the other direction, however.
- When both directions have been shut down, the connection is released.
- Normally, four TCP segments are needed to release a connection, one *FIN* and one *ACK* for each direction. However, it is possible for the first *ACK* and the second *FIN* to be contained in the same segment, reducing the total count to three.



TCP Connection Management Modeling

The steps required establishing and release connections can be represented in a finite state machine with the 11 states listed in Figure. In each state, certain events are legal. When a legal event happens, some action may be taken. If some other event happens, an error is reported.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

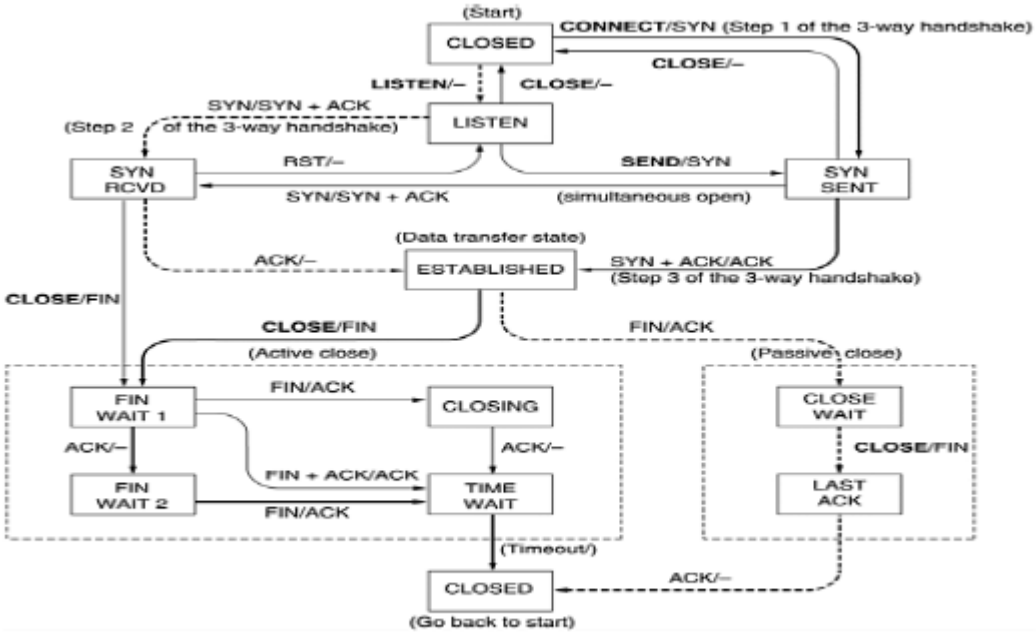


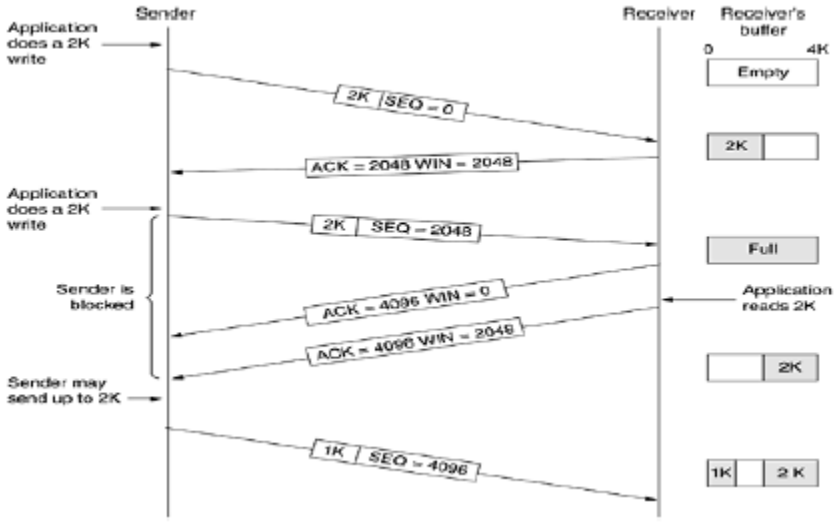
Figure 4.14 - TCP connection management finite state machine.

TCP Connection management from server's point of view:

1. The server does a **LISTEN** and settles down to see who turns up.
2. When a **SYN** comes in, the server acknowledges it and goes to the **SYNRCVD** state
3. When the servers **SYN** is itself acknowledged the 3-way handshake is complete and server goes to the **ESTABLISHED** state. Data transfer can now occur.
4. When the client has had enough, it does a close, which causes a **FIN** to arrive at the server [dashed box marked passive close].
5. The server is then signaled.
6. When it too, does a **CLOSE**, a **FIN** is sent to the client.
7. When the client's acknowledgement shows up, the server releases the connection and deletes the connection record.

TCP Transmission Policy

1. In the above example, the receiver has 4096-byte buffer.
2. If the sender transmits a 2048-byte segment that is correctly received, the receiver will acknowledge the segment.
3. Now the receiver will advertise a window of 2048 as it has only 2048 of buffer space, now.
4. Now the sender transmits another 2048 bytes which are acknowledged, but the advertised window is'0'.
5. The sender must stop until the application process on the receiving host has removed some data from the buffer, at which time TCP can advertise a layer window.



TCP CONGESTION CONTROL:

TCP does to try to prevent the congestion from occurring in the first place in the following way:

When a connection is established, a suitable window size is chosen and the receiver specifies a window based on its buffer size. If the sender sticks to this window size, problems will not occur due to buffer overflow at the receiving end. But they may still occur due to internal congestion within the network. Let's see this problem occurs.

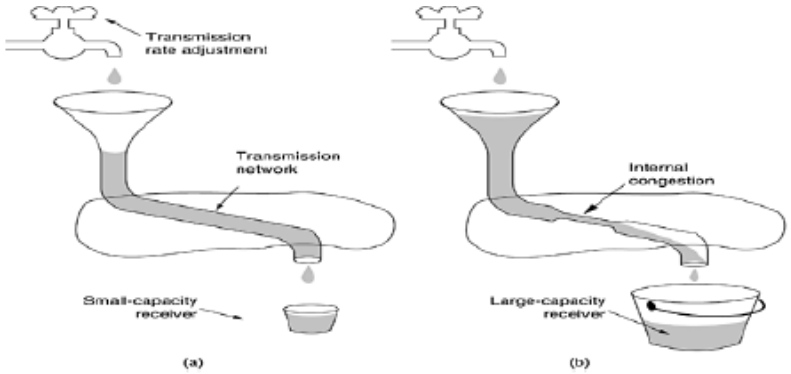


Figure 4 (a) A fast network feeding a low-capacity receiver. (b) A slow network feeding a high-capacity receiver.

In fig (a): We see a thick pipe leading to a small- capacity receiver. As long as the sender does not send more water than the bucket can contain, no water will be lost.
In fig (b): The limiting factor is not the bucket capacity, but the internal carrying capacity of the n/w. if too much water comes in too fast, it will backup and some will be lost.

- When a connection is established, the sender initializes the congestion window to the size of the max segment in use our connection.
- It then sends one max segment .if this max segment is acknowledged before the timer goes off, it adds one segment s worth of bytes to the congestion window to make it two maximum size segments and sends 2 segments.
- As each of these segments is acknowledged, the congestion window is increased by one max segment size.
- When the congestion window is 'n' segments, if all 'n' are acknowledged on time, the congestion window is increased by the byte count corresponding to 'n' segments.

- The congestion window keeps growing exponentially until either a time out occurs or the receiver's window is reached.
- The internet congestion control algorithm uses a third parameter, the **"threshold"** in addition to receiver and congestion windows.

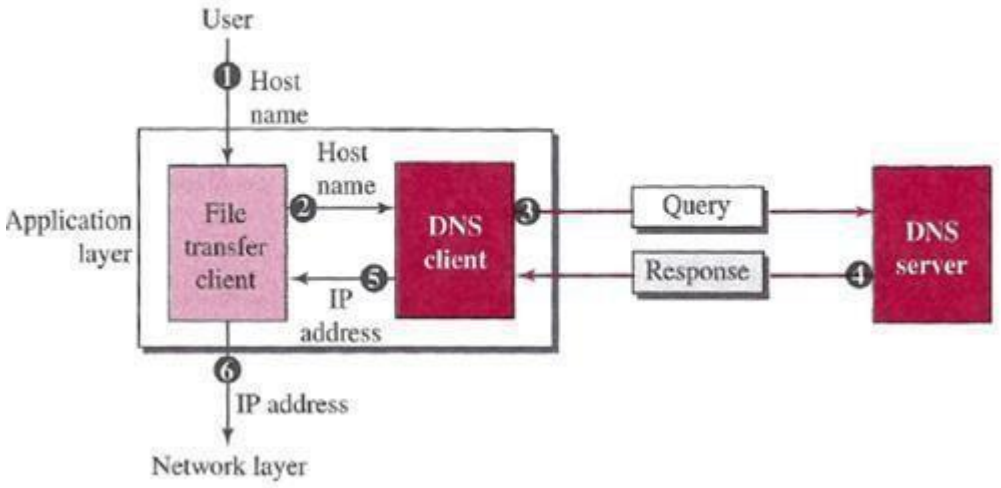
Different congestion control algorithms used by TCP are:

- RTT variance Estimation.
- Exponential RTO back-off Re-transmission Timer Management
- Karn's Algorithm
- Slow Start
- Dynamic window sizing on congestion
- Fast Retransmit Window Management
- Fast Recovery

DOMAIN NAME SYSTEM (DNS)

The host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

Purpose of DNS



A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as *afilesource.com.*

Name Space

A **name** space that maps each address to a unique name can be organized in two ways: flat or hierarchical. In a *flat name space*, a name is assigned to an address. A name in this space is a sequence of characters without structure. In a *hierarchical name space*, each name is made of several parts.

Domain Name Space

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top.

Label

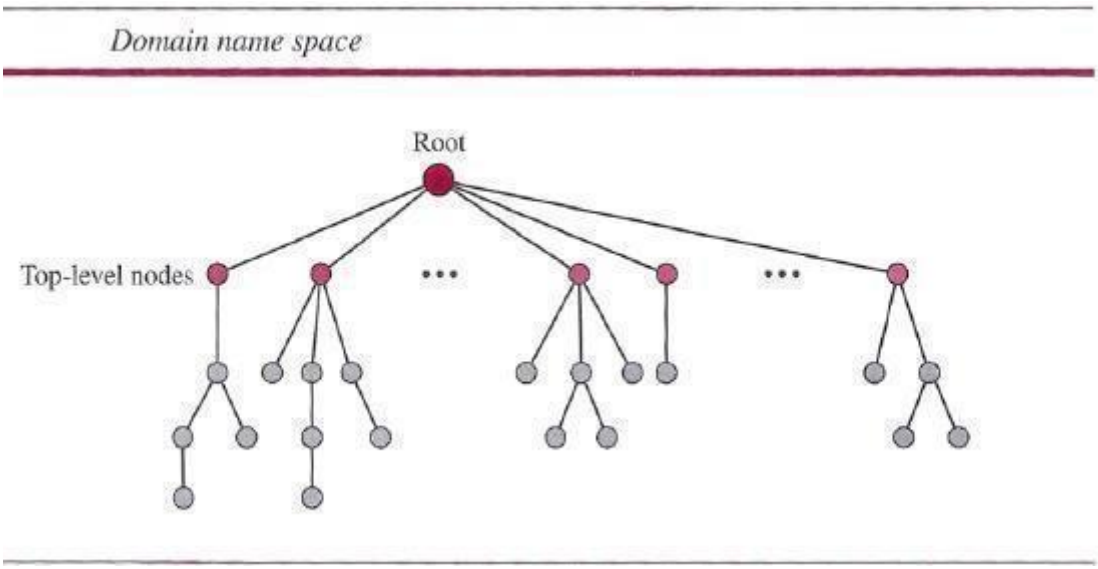
Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string).

Domain Name

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). If a label is not terminated by a null string, it is called a partially qualified domain name PQDN).

Domain

A domain is a sub tree of the domain name space. The name of the domain is the name of the node at the top of the sub tree.



Electronic Mail

Electronic Mail (e-mail) is one of most widely used services of Internet. This service allows an Internet user to send a message in formatted manner (mail) to the other Internet user in any part of world.

Components of E-Mail System

The basic components of an email system are : User Agent (UA), Message Transfer Agent (MTA), Mail Box, and Spool file. These are explained as following below.

1.User Agent (UA) :

The UA is normally a program which is used to send and receive mail. Sometimes, it is called as mail reader. It accepts variety of commands for composing, receiving and replying to messages as well as for manipulation of the mailboxes.

2. Message Transfer Agent (MTA)

MTA is actually responsible for transfer of mail from one system to another. To send a mail, a system must have client MTA and system MTA. It transfer mail to mailboxes of recipients if they are connected in the same machine. It delivers mail to peer MTA if destination

mailbox is in another machine. The delivery from one MTA to another MTA is done by Simple Mail Transfer Protocol.

1. Mailbox :

It is a file on local hard drive to collect mails. Delivered mails are present in this file. The user can read it delete it according to his/her requirement. To use e-mail system each user must have a mailbox . Access to mailbox is only to owner of mailbox.

2. Spool file :

This file contains mails that are to be sent. User agent appends outgoing mails in this file using SMTP. MTA extracts pending mail from spool file for their delivery. E-mail allows one name, an alias, to represent several different e-mail addresses. It is known as mailing list, Whenever user have to sent a message, system checks recipients's name against alias database. If mailing list is present for defined alias, separate messages, one for each entry in the list, must be prepared and handed to MTA. If for defined alias, there is no such mailing list is present, name itself becomes naming address and a single message is delivered to mail transfer entity.

Services provided by E-mail system:

1. Composition

The composition refer to process that creates messages and answers. For composition any kind of text editor can be used.

2. Transfer –

Transfer means sending procedure of mail i.e. from the sender to recipient.

3. Reporting –

Reporting refers to confirmation for delivery of mail. It help user to check whether their mail is delivered, lost or rejected.

4. Displaying –

It refers to present mail in form that is understand by the user.

5. 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.

E-Mail Format

Electronic Mail (e-mail) is one of the most widely used services of the Internet. This service allows an Internet user to send a message in a formatted manner (mail) to other Internet users in any part of the world. Message in the mail not only contain text, but it also contains images, audio and videos data. The person who is sending mail is called sender and person who receives mail is called the recipient. It is just like postal mail service.

Format of E-mail :

An e-mail consists of three parts that are as follows:

1. Envelope

2. Header

3. Body

These are explained as following below.

1. Envelope :

The envelope part encapsulates the message. It contains all information that is required for sending any e-mail such as destination address, priority and security level. The envelope is used by MTAs for routing message.

2. Header :

The header consists of a series of lines. Each header field consists of a single line of ASCII text specifying field name, colon and value. The main header fields related to message transport are :

1.To: It specifies the DNS address of the primary recipient(s).

2.Cc : It refers to carbon copy. It specifies address of secondary recipient(s).

3.BCC: It refers to blind carbon copy. It is very similar to Cc. The only difference between Cc and Bcc is that it allow user to send copy to the third party without primary and secondary recipient knowing about this.

4.From : It specifies name of person who wrote message.

5.Sender : It specifies e-mail address of person who has sent message.

6.Received : It refers to identity of sender's, data and also time message was received. It also contains the information which is used to find bugs in routing system.

7.Return-Path: It is added by the message transfer agent. This part is used to specify how to get back to the sender.

3. Body:- The body of a message contains text that is the actual content/message that needs to be sent, such as "Employees who are eligible for the new health care program should contact their supervisors by next Friday if they want to switch." The message body also may include signatures or automatically generated text that is inserted by the sender's email system.

The above-discussed field is represented in tabular form as follows :

Advantages and Disadvantages of E-mail

1. E-mails provides faster and easy mean of communication. One can send message to any person at any place of world by just clicking mouse.

2. Various folders and sub-folders can be created within inbox of mail, so it provide management of messages.

3. It is effective and cheap means of communication because single message can be send to multiple people at same time.

4. E-mails are very easy to filter. User according to his/her priority can prioritize e-mail by specifying subject of e-mail.

5. E-mail is not just only for textual message. One can send any kind of multimedia within mail.

Disadvantages of E-mail :

1.It is source of viruses. It is capable to harm one's computer and read out user's e-mail address book and send themselves to number of people around the world.

2.It can be source of various spams. These spam mails can fill up inbox and to deletion of these mail consumes lot of time.

3.It is informal method of communication. The documents those require signatures are not managed by e-mail.

HTTP

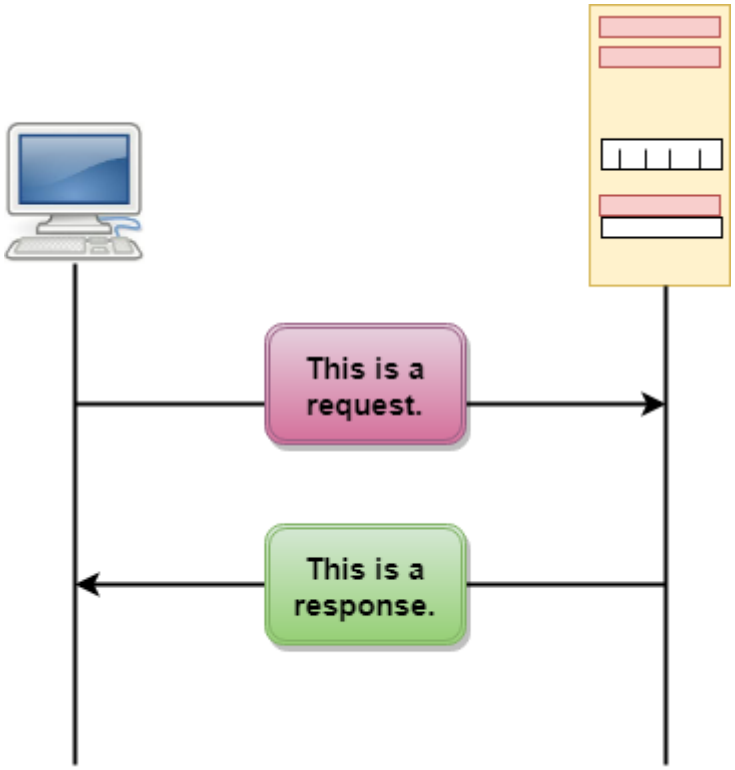
- HTTP stands for **HyperText Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because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

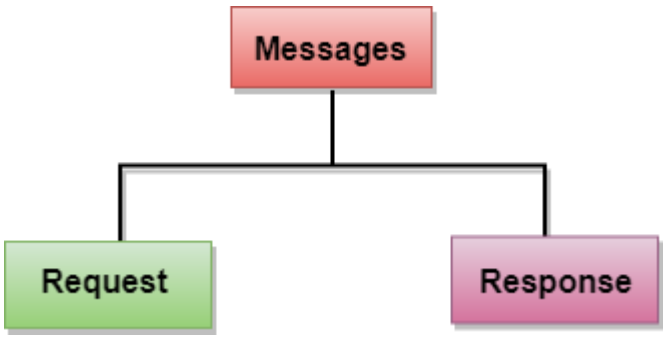
HTTP Transactions



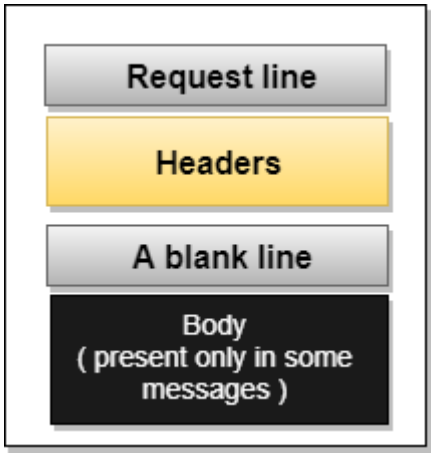
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages

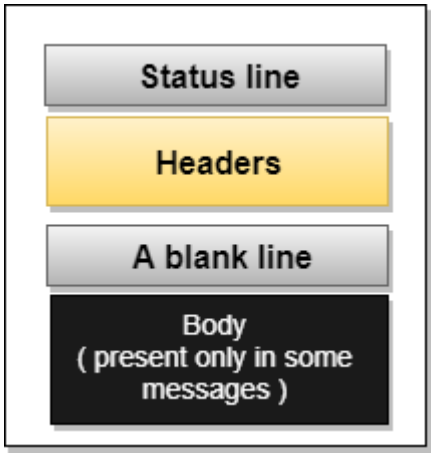
HTTP messages are of two types: request and response. Both the message types follow the same message format.



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.

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.

There are two types of connections in FTP:

Computer Network 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.