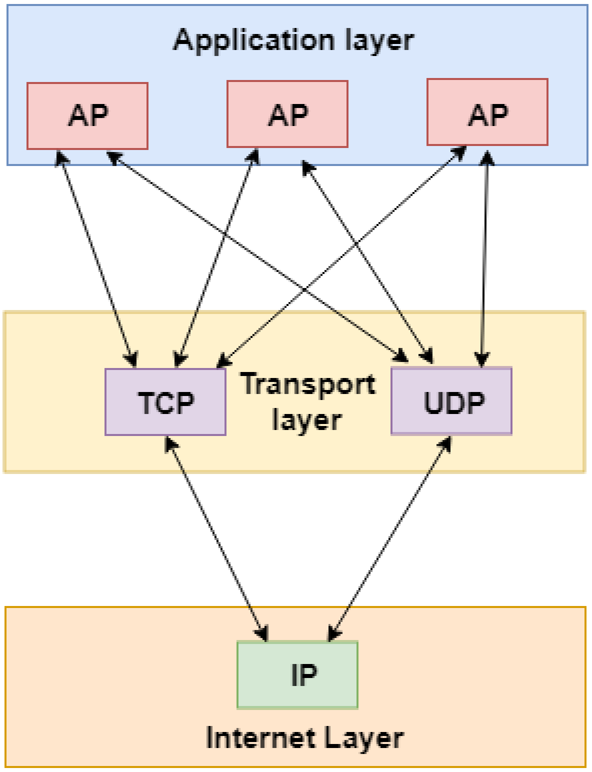
Transport Layer

- The transport layer is a 4th layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.
- The transport layer protocols are implemented in the end systems but not in the network routers.
- A computer network provides more than one protocol to the network applications. For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.
- All transport layer protocols provide multiplexing/demultiplexing service. It also provides other services such as reliable data transfer, bandwidth guarantees, and delay guarantees.
- Each of the applications in the application layer has the ability to send a message by using TCP or UDP. The application communicates by using either of these two protocols. Both TCP and UDP will then communicate with the internet protocol in the internet layer. The applications can read and write to the transport layer. Therefore, we can say that communication is a two-way process.

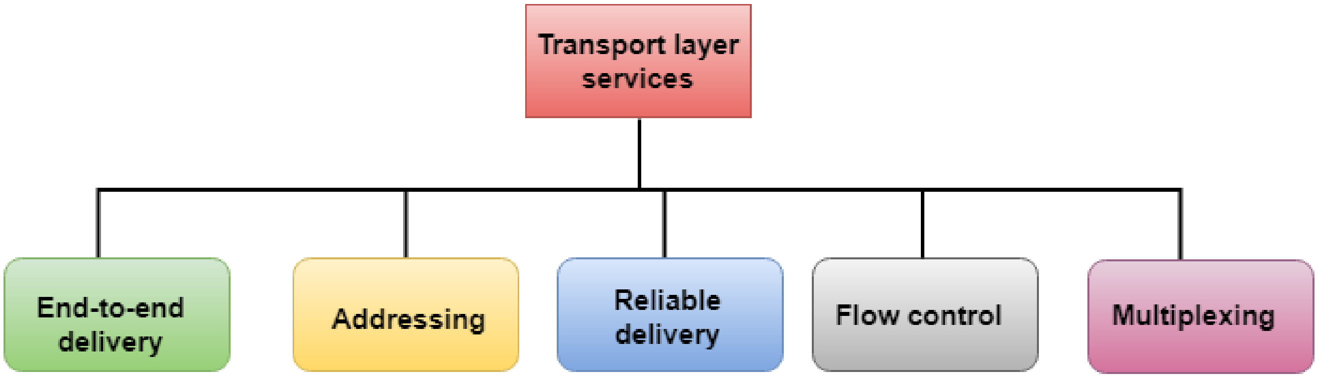


Services provided by the Transport Layer

The services provided by the transport layer are similar to those of the data link layer. The data link layer provides the services within a single network while the transport layer provides the services across an internetwork made up of many networks. The data link layer controls the physical layer while the transport layer controls all the lower layers.

The services provided by the transport layer protocols can be divided into five categories:

- End-to-end delivery
- Addressing
- Reliable delivery
- Flow control
- Multiplexing



End-to-end delivery:

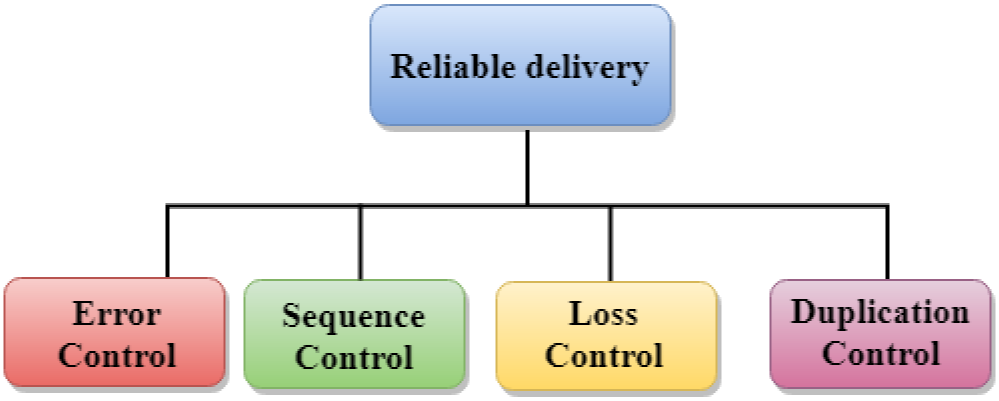
The transport layer transmits the entire message to the destination. Therefore, it ensures the end-to-end delivery of an entire message from a source to the destination.

Reliable delivery:

The transport layer provides reliability services by retransmitting the lost and damaged packets.

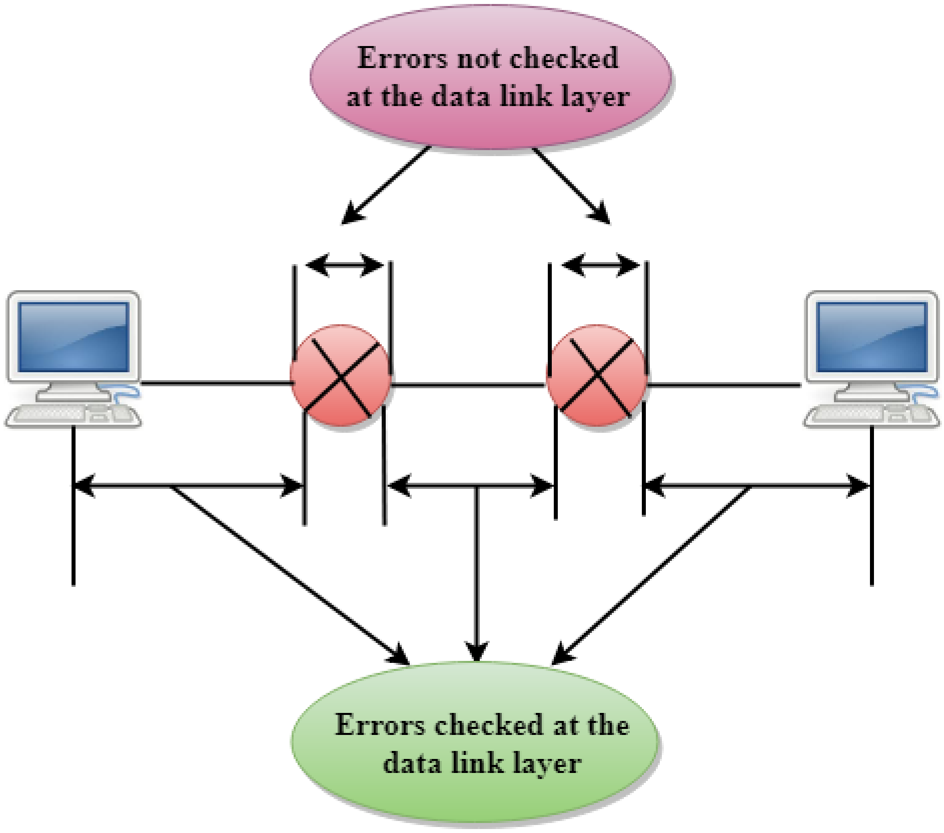
The reliable delivery has four aspects:

- Error control
- Sequence control
- Loss control
- Duplication control



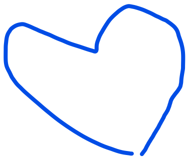
Error Control

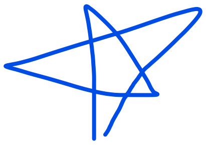
- The primary role of reliability is **Error Control**. In reality, no transmission will be 100 percent error-free delivery. Therefore, transport layer protocols are designed to provide error-free transmission.
- The data link layer also provides the error handling mechanism, but it ensures only node-to-node error-free delivery. However, node-to-node reliability does not ensure the end-to-end reliability.
- The data link layer checks for the error between each network. If an error is introduced inside one of the routers, then this error will not be caught by the data link layer. It only detects those errors that have been introduced between the beginning and end of the link. Therefore, the transport layer performs the checking for the errors end-to-end to ensure that the packet has arrived correctly.


Errors not checked at the data link layer
Errors checked at the data link layer

Sequence Control

- The second aspect of the reliability is sequence control which is implemented at the transport layer.





- On the sending end, the transport layer is responsible for ensuring that the packets received from the upper layers can be used by the lower layers. On the receiving end, it ensures that the various segments of a transmission can be correctly reassembled.

Loss Control

Loss Control is a third aspect of reliability. The transport layer ensures that all the fragments of a transmission arrive at the destination, not some of them. On the sending end, all the fragments of transmission are given sequence numbers by a transport layer. These sequence numbers allow the receiver?s transport layer to identify the missing segment.

Duplication Control

Duplication Control is the fourth aspect of reliability. The transport layer guarantees that no duplicate data arrive at the destination. Sequence numbers are used to identify the lost packets; similarly, it allows the receiver to identify and discard duplicate segments.

Flow Control

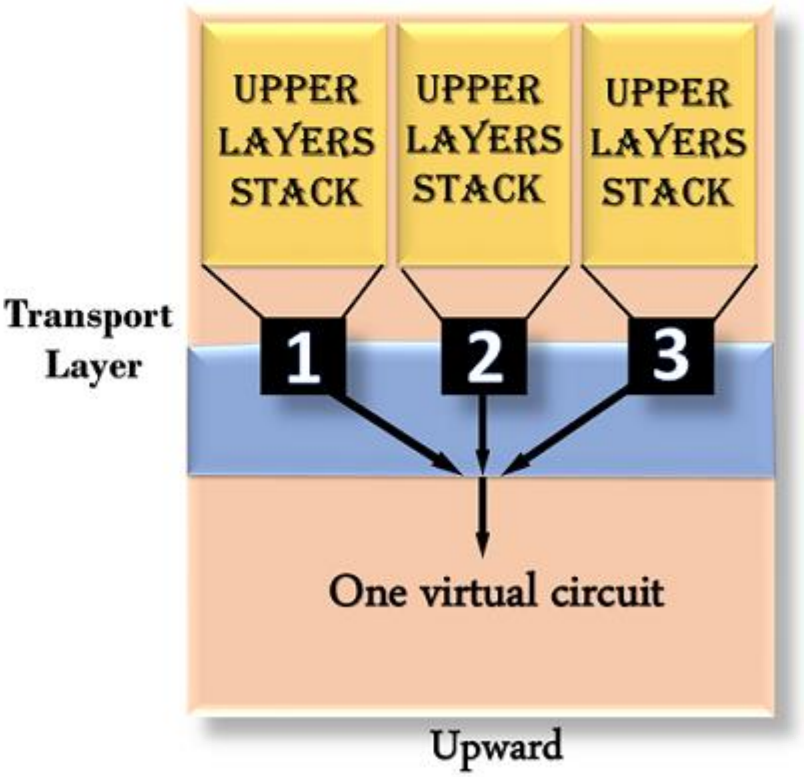
Flow control is used to prevent the sender from overwhelming the receiver. If the receiver is overloaded with too much data, then the receiver discards the packets and asking for the retransmission of packets. This increases network congestion and thus, reducing the system performance. The transport layer is responsible for flow control. It uses the sliding window protocol that makes the data transmission more efficient as well as it controls the flow of data so that the receiver does not become overwhelmed. Sliding window protocol is byte oriented rather than frame oriented.

Multiplexing

The transport layer uses the multiplexing to improve transmission efficiency.

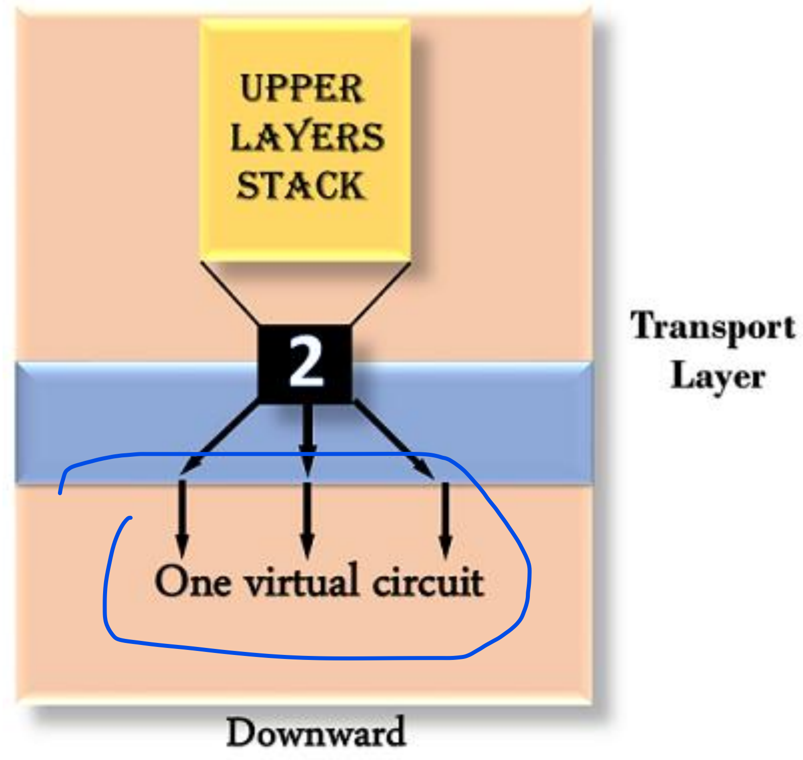
Multiplexing can occur in two ways:

- **Upward multiplexing:** Upward multiplexing means multiple transport layer connections use the same network connection. To make more cost-effective, the transport layer sends several transmissions bound for the same destination along the same path; this is achieved through upward multiplexing.



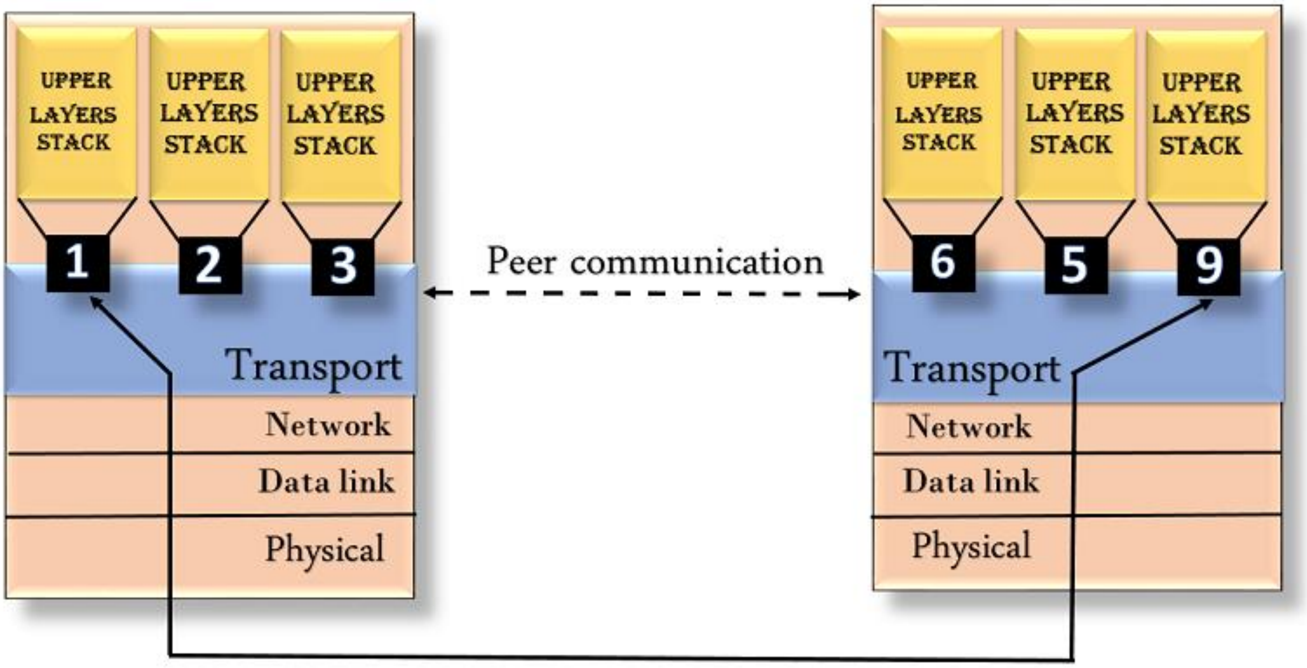
Upward

- **Downward multiplexing:** Downward multiplexing means one transport layer connection uses the multiple network connections. Downward multiplexing allows the transport layer to split a connection among several paths to improve the throughput. This type of multiplexing is used when networks have a low or slow capacity.



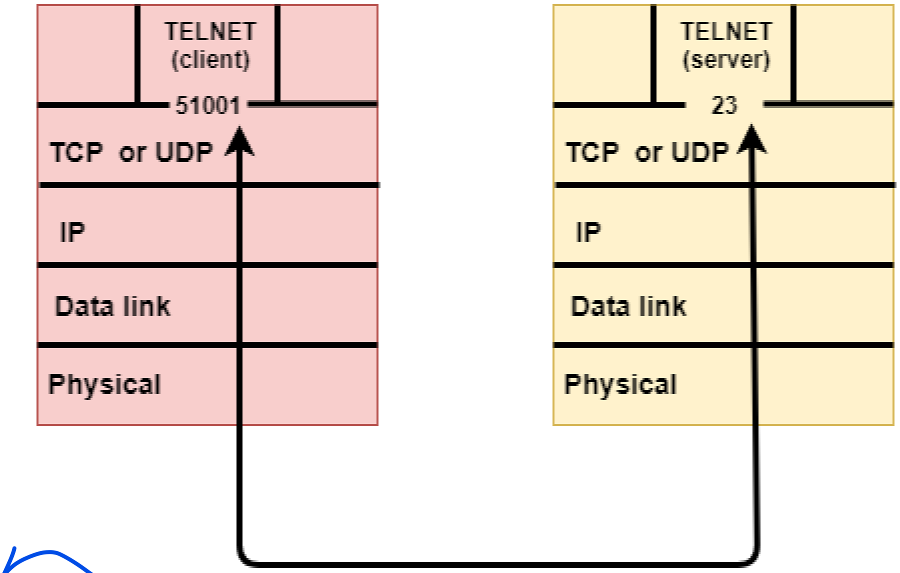
Addressing

- According to the layered model, the transport layer interacts with the functions of the session layer. Many protocols combine session, presentation, and application layer protocols into a single layer known as the application layer. In these cases, delivery to the session layer means the delivery to the application layer. Data generated by an application on one machine must be transmitted to the correct application on another machine. In this case, addressing is provided by the transport layer.
- The transport layer provides the user address which is specified as a station or port. The port variable represents a particular TS user of a specified station known as a Transport Service access point (TSAP). Each station has only one transport entity.
- The transport layer protocols need to know which upper-layer protocols are communicating.



Transport Layer protocols

- The transport layer is represented by two protocols: TCP and UDP.
- The IP protocol in the network layer delivers a datagram from a source host to the destination host.
- Nowadays, the operating system supports multiuser and multiprocessing environments, an executing program is called a process. When a host sends a message to other host means that source process is sending a process to a destination process. The transport layer protocols define some connections to individual ports known as protocol ports.
- An IP protocol is a host-to-host protocol used to deliver a packet from source host to the destination host while transport layer protocols are port-to-port protocols that work on the top of the IP protocols to deliver the packet from the originating port to the IP services, and from IP services to the destination port.
- Each port is defined by a positive integer address, and it is of 16 bits.

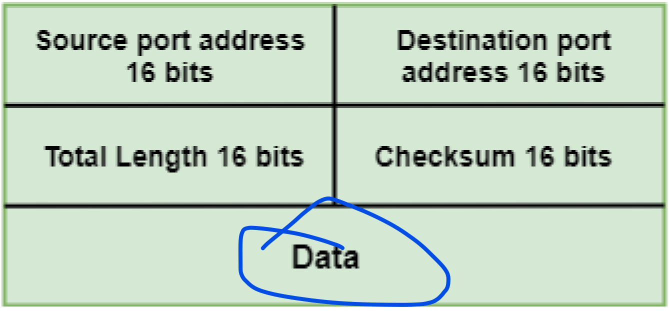


UDP

- UDP stands for **User Datagram Protocol**.
- UDP is a simple protocol and it provides nonsequenced transport functionality.
- UDP is a connectionless protocol.
- This type of protocol is used when reliability and security are less important than speed and size.
- UDP is an end-to-end transport level protocol that adds transport-level addresses, checksum error control, and length information to the data from the upper layer.
- The packet produced by the UDP protocol is known as a user datagram.

User Datagram Format

The user datagram has a 16-byte header which is shown below:



Source port address 16 bits	Destination port address 16 bits
Total Length 16 bits	Checksum 16 bits
Data	

Where,

- **Source port address:** It defines the address of the application process that has delivered a message. The source port address is of 16 bits address.
- **Destination port address:** It defines the address of the application process that will receive the message. The destination port address is of a 16-bit address.
- **Total length:** It defines the total length of the user datagram in bytes. It is a 16-bit field.
- **Checksum:** The checksum is a 16-bit field which is used in error detection.

Disadvantages of UDP protocol

- UDP provides basic functions needed for the end-to-end delivery of a transmission.
- It does not provide any sequencing or reordering functions and does not specify the damaged packet when reporting an error.
- UDP can discover that an error has occurred, but it does not specify which packet has been lost as it does not contain an ID or sequencing number of a particular data segment.

TCP

- TCP stands for Transmission Control Protocol.
- It provides full transport layer services to applications.
- It is a connection-oriented protocol means the connection established between both the ends of the transmission. For creating the connection, TCP generates a virtual circuit between sender and receiver for the duration of a transmission.

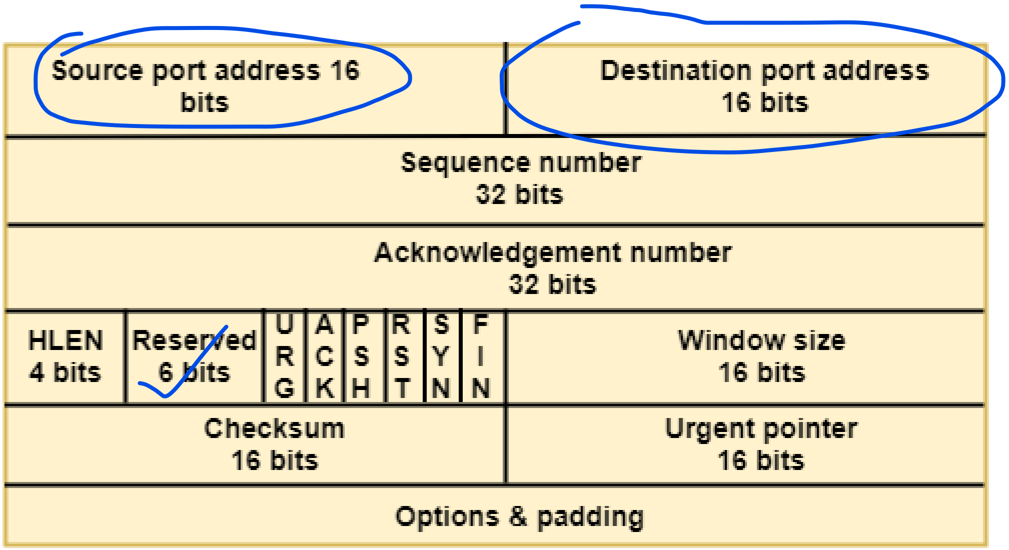
Features Of TCP protocol

- **Stream data transfer:** TCP protocol transfers the data in the form of contiguous stream of bytes. TCP group the bytes in the form of TCP segments and then passed it to the IP layer for transmission to the destination. TCP itself segments the data and forward to the IP.
- **Reliability:** TCP assigns a sequence number to each byte transmitted and expects a positive acknowledgement from the receiving TCP. If ACK is not received within a timeout interval, then the data is retransmitted to the destination.

The receiving TCP uses the sequence number to reassemble the segments if they arrive out of order or to eliminate the duplicate segments.

- **Flow Control:** When receiving TCP sends an acknowledgement back to the sender indicating the number the bytes it can receive without overflowing its internal buffer. The number of bytes is sent in ACK in the form of the highest sequence number that it can receive without any problem. This mechanism is also referred to as a window mechanism.
- **Multiplexing:** Multiplexing is a process of accepting the data from different applications and forwarding to the different applications on different computers. At the receiving end, the data is forwarded to the correct application. This process is known as demultiplexing. TCP transmits the packet to the correct application by using the logical channels known as ports.
- **Logical Connections:** The combination of sockets, sequence numbers, and window sizes, is called a logical connection. Each connection is identified by the pair of sockets used by sending and receiving processes.
- **Full Duplex:** TCP provides Full Duplex service, i.e., the data flow in both the directions at the same time. To achieve Full Duplex service, each TCP should have sending and receiving buffers so that the segments can flow in both the directions. TCP is a connection-oriented protocol. Suppose the process A wants to send and receive the data from process B. The following steps occur:
    - Establish a connection between two TCPs.
    - Data is exchanged in both the directions.
    - The Connection is terminated.

TCP Segment Format



Source port address 16 bits					Destination port address 16 bits			
Sequence number 32 bits								
Acknowledgement number 32 bits								
HLEN 4 bits	Reserved 6 bits	URG	ACK	PSH	RST	SYN	FIN	Window size 16 bits
Checksum 16 bits					Urgent pointer 16 bits			
Options & padding								

Where,

- **Source port address:** It is used to define the address of the application program in a source computer. It is a 16-bit field.
- **Destination port address:** It is used to define the address of the application program in a destination computer. It is a 16-bit field.
- **Sequence number:** A stream of data is divided into two or more TCP segments. The 32-bit sequence number field represents the position of the data in an original data stream.
- **Acknowledgement number:** A 32-field acknowledgement number acknowledge the data from other communicating devices. If ACK field is set to 1, then it specifies the sequence number that the receiver is expecting to receive.
- **Header Length (HLEN):** It specifies the size of the TCP header in 32-bit words. The minimum size of the header is 5 words, and the maximum size of the header is 15 words. Therefore, the maximum size of the TCP header is 60 bytes, and the minimum size of the TCP header is 20 bytes.
- **Reserved:** It is a six-bit field which is reserved for future use.
- **Control bits:** Each bit of a control field functions individually and independently. A control bit defines the use of a segment or serves as a validity check for other fields.

There are total six types of flags in control field:

- **URG:** The URG field indicates that the data in a segment is urgent.
- **ACK:** When ACK field is set, then it validates the acknowledgement number.

- **PSH:** The PSH field is used to inform the sender that higher throughput is needed so if possible, data must be pushed with higher throughput.
- **RST:** The reset bit is used to reset the TCP connection when there is any confusion occurs in the sequence numbers.
- **SYN:** The SYN field is used to synchronize the sequence numbers in three types of segments: connection request, connection confirmation (with the ACK bit set), and confirmation acknowledgement.
- **FIN:** The FIN field is used to inform the receiving TCP module that the sender has finished sending data. It is used in connection termination in three types of segments: termination request, termination confirmation, and acknowledgement of termination confirmation.
  - **Window Size:** The window is a 16-bit field that defines the size of the window.
  - **Checksum:** The checksum is a 16-bit field used in error detection.
  - **Urgent pointer:** If URG flag is set to 1, then this 16-bit field is an offset from the sequence number indicating that it is a last urgent data byte.
  - **Options and padding:** It defines the optional fields that convey the additional information to the receiver.

Differences b/w TCP & UDP

Basis for Comparison	TCP	UDP
Definition	TCP establishes a virtual circuit before transmitting the data.	UDP transmits the data directly to the destination computer without verifying whether the receiver is ready to receive or not.
Connection Type	It is a Connection-Oriented protocol	It is a Connectionless protocol
Speed	slow	high
Reliability	It is a reliable protocol.	It is an unreliable protocol.
Header size	20 bytes	8 bytes
acknowledgement	It waits for the acknowledgement of data and has the ability to resend the lost packets.	It neither takes the acknowledgement, nor it retransmits the damaged frame.

Congestion Control and Quality of Service

Congestion Control:

Congestion Control is a type of network layer issue, and it is concerned with what happens when there is more data in the network than can be sent with reasonable packet delays, and there are no lost packets.

Causes of Congestion:

The main cause of congestion is a huge amount of data traffic. But other factors are equally important for making congestion as given below:

1. **Sudden arrival of large data** (called burst data) from many input lines and trying to access a single output line of a router. In this case, the particular output line is blocked if its bandwidth isn't sufficiently high.
2. **Low bandwidth** line will produce congestion even if the data rate isn't too high.
3. **Mismatch between the speeds of different components** of the system may also produce congestion.

Principle of Congestion Control:

Congestion can be controlled in either of two ways:

1. Open-Loop Control
2. Closed-Loop Control

Open-Loop Control:

Open-Loop Control always involves good design such that congestion doesn't occur. So while designing the system all possible causes of congestion and the remedy for each of them are to be incorporated before installation. Some of the concerned points are:

- When to accept new traffic
- When to discard packets
- Which packet to discard
- Making scheduling decisions at various points of the network.

Closed-Loop Control:

Closed-Loop Control is always based on the concept of feedback from the network. It contains the following parts:

i. Monitor the system to detect when and where congestion occurs.

ii. Pass this information to the places where actions can be taken.

iii. Adjust system operation to correct the problem.

Quality of Service (QoS):

With the increasing use of networks assuring the good **Quality of Service** (QoS) became another important requirement of networks. Four characteristics and parameters contribute to the quality of service are:

1. Reliability
2. Delay
3. Jitter
4. Bandwidth

Techniques for Achieving Good QoS:

A single technique can’t provide good quality of service, rather a combination of techniques is implemented by the system designers to get better QoS. Some of the techniques are given below:

Over Provisioning:

It is the easiest solution. A sufficient number of routers with sufficient capacity, buffer space and bandwidth are used in the network such that it can always provide good QoS. But it is an expensive solution.

Buffering:

In this technique, packets are buffered before delivery. Packets deliver at regular intervals of time. This method doesn’t improve reliability and bandwidth but it increases the delay and improves jitter.

Traffic Shaping:

In the previous technique, hosts are designed to transmit at a uniform rate. But sometimes it's not possible to meet regularity in transmission because of various reasons. Traffic shaping smoothens the traffic on the server side rather than on the client side. The steps are given below:

1. Service level agreement is done between the user and subnet.
2. As long as the user sends packets according to the agreed contract, the carrier promises to deliver them all in a timely fashion.
3. It is highly important for real-time data transfer.
4. Monitoring the traffic flow is called Traffic Policing.
5. Leaky Bucket and Token Bucket Algorithm is used for traffic shaping.

Resource Reservation:

Traffic shaping improves the QoS but it maintains that some predefined path must be set before sending data packets. For a virtual-circuit connection, the path is set before data transfer and resources can be reserved for the specific circuit. Kinds of resources that can be preserved are :

i. **Bandwidth:** High Bandwidth can handle a large data rate. So, it provides high bandwidth that will ensure good quality of service.
ii. **Buffer Space:** High Buffer Space is reserved for accepting and temporarily storing packets such that traffic shaping can be done and good QoS is provided.
iii. **CPU Cycles:** For timely processing of packets CPU cycles are also reserved thereby providing good QoS.

Application Layer

The application layer in the OSI model is the closest layer to the end user which means that the application layer and end user can interact directly with the software application. The application layer programs are based on client and servers.

The Application layer includes the following functions:

- **Identifying communication partners:** The application layer identifies the availability of communication partners for an application with data to transmit.
- **Determining resource availability:** The application layer determines whether sufficient network resources are available for the requested communication.
- **Synchronizing communication:** All the communications occur between the applications requires cooperation which is managed by an application layer.

Services of Application Layers

- **Network Virtual terminal:** An application layer allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal, which in turn, talks to the host. The remote host thinks that it is communicating with one of its own terminals, so it allows the user to log on.
- **File Transfer, Access, and Management (FTAM):** An application allows a user to access files in a remote computer, to retrieve files from a computer and to manage files in a remote computer. FTAM defines a hierarchical virtual file in terms of file structure, file attributes and the kind of operations performed on the files and their attributes.
- **Addressing:** To obtain communication between client and server, there is a need for addressing. When a client made a request to the server, the request contains the server address and its own address. The server response to the client request, the request contains the destination address, i.e., client address. To achieve this kind of addressing, DNS is used.
- **Mail Services:** An application layer provides Email forwarding and storage.
- **Directory Services:** An application contains a distributed database that provides access for global information about various objects and services.

  Authentication: It authenticates the sender or receiver's message or both.

Network Application Architecture

Application architecture is different from the network architecture. The network architecture is fixed and provides a set of services to applications. The application architecture, on the other hand, is designed by the application developer and defines how the application should be structured over the various end systems.

Application architecture is of two types:

- **Client-server architecture:** An application program running on the local machine sends a request to another application program is known as a client, and a program that serves a request is known as a server. For example, when a web server receives a request from the client host, it responds to the request to the client host.

Characteristics Of Client-server architecture:

- In Client-server architecture, clients do not directly communicate with each other. For example, in a web application, two browsers do not directly communicate with each other.
- A server is fixed, well-known address known as IP address because the server is always on while the client can always contact the server by sending a packet to the sender's IP address.

Disadvantage Of Client-server architecture:

It is a single-server based architecture which is incapable of holding all the requests from the clients. For example, a social networking site can become overwhelmed when there is only one server exists.

- **P2P (peer-to-peer) architecture:** It has no dedicated server in a data center. The peers are the computers which are not owned by the service provider. Most of the peers reside in the homes, offices, schools, and universities. The peers communicate with each other without passing the information through a dedicated server, this architecture is known as peer-to-peer architecture. The applications based on P2P architecture includes file sharing and internet telephony.

Features of P2P architecture

- **Self scalability:** In a file sharing system, although each peer generates a workload by requesting the files, each peer also adds a service capacity by distributing the files to the peer.

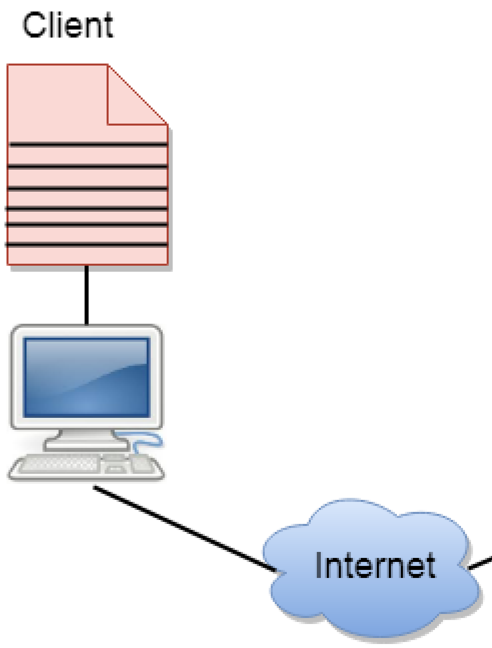
- **Cost-effective:** It is cost-effective as it does not require significant server infrastructure and server bandwidth.

Client and Server processes

- A network application consists of a pair of processes that send the messages to each other over a network.
- In P2P file-sharing system, a file is transferred from a process in one peer to a process in another peer. We label one of the two processes as the client and another process as the server.
- With P2P file sharing, the peer which is downloading the file is known as a client, and the peer which is uploading the file is known as a server. However, we have observed in some applications such as P2P file sharing; a process can be both as a client and server. Therefore, we can say that a process can both download and upload the files.

Client and Server model

- A client and server networking model is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.
- The application programs using the client-server model should follow the given below strategies:



- An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.
- A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.
- A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.
- Services are required frequently, and many users have a specific client-server application program. For example, the client-server application program allows the user to access the files, send e-mail, and so on. If the services are more customized, then we should have one generic application program that allows the user to access the services available on the remote computer.

Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

Advantages of Client-server networks:

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

Disadvantages of Client-Server network:

- **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.
- A client/server network is very decisive. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.
- Sometimes the resources exist in the server but may not exist in the client. For example, If the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

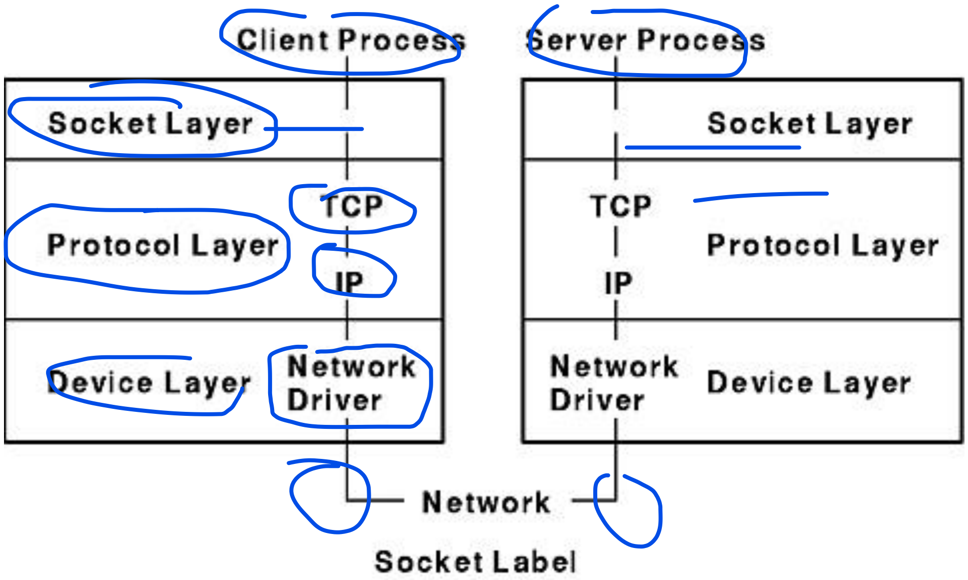
Sockets Interface

The kernel structure consists of three layers: the socket layer, the protocol layer, and the device layer.

- The *socket layer* supplies the interface between the subroutines and lower layers, the *protocol layer* contains the protocol modules used for communication, and the *device layer* contains the device drivers that control the network devices. Protocols and drivers are dynamically loadable. The Socket Label figure (Figure 1) illustrates the relationship between the layers.

C-S P+P

- Figure 1. Socket Label



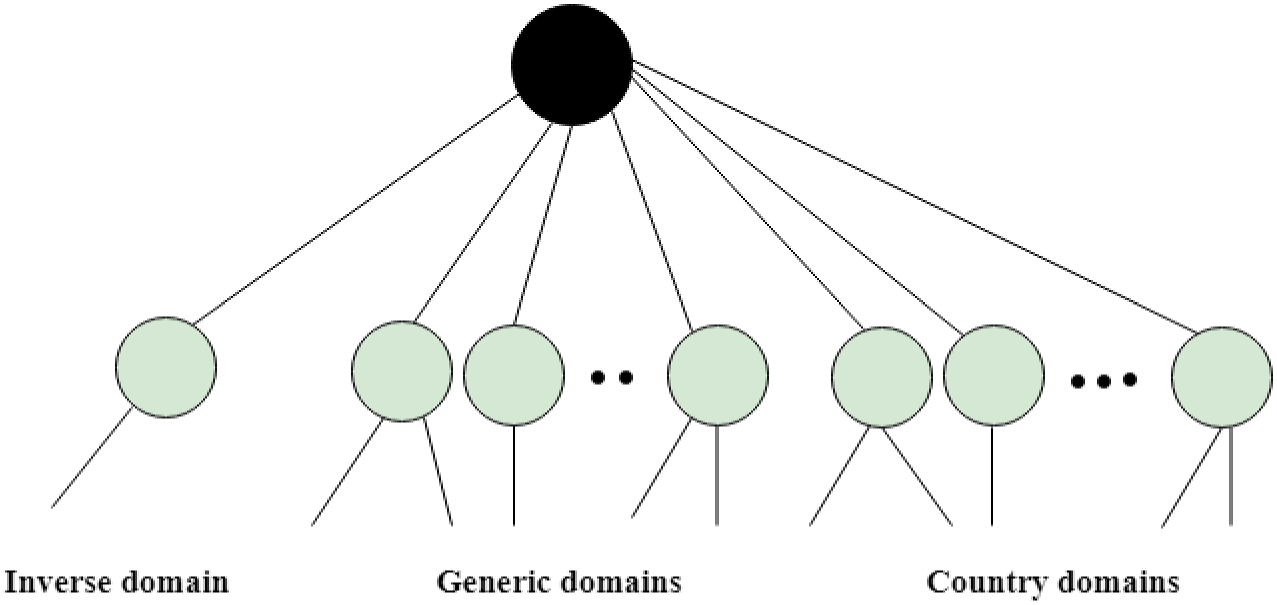
- Processes communicate using the client and server model. In this model, a server process, one end point of a two-way communication path, listens to a socket. The client process, the other end of the communication path, communicates to the server process over another socket. The client process can be on another machine. The kernel maintains internal connections and routes data from client to server.
- Within the socket layer, the socket data structure is the focus of activity. The system-call interface subroutines manage the activities related to a subroutine, collecting the subroutine parameters and converting program data into the format expected by second-level subroutines.
- Most of the socket facilities are implemented within second-level subroutines. These second-level subroutines directly manipulate socket data structures and manage the synchronization between asynchronous activities.

DNS

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

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

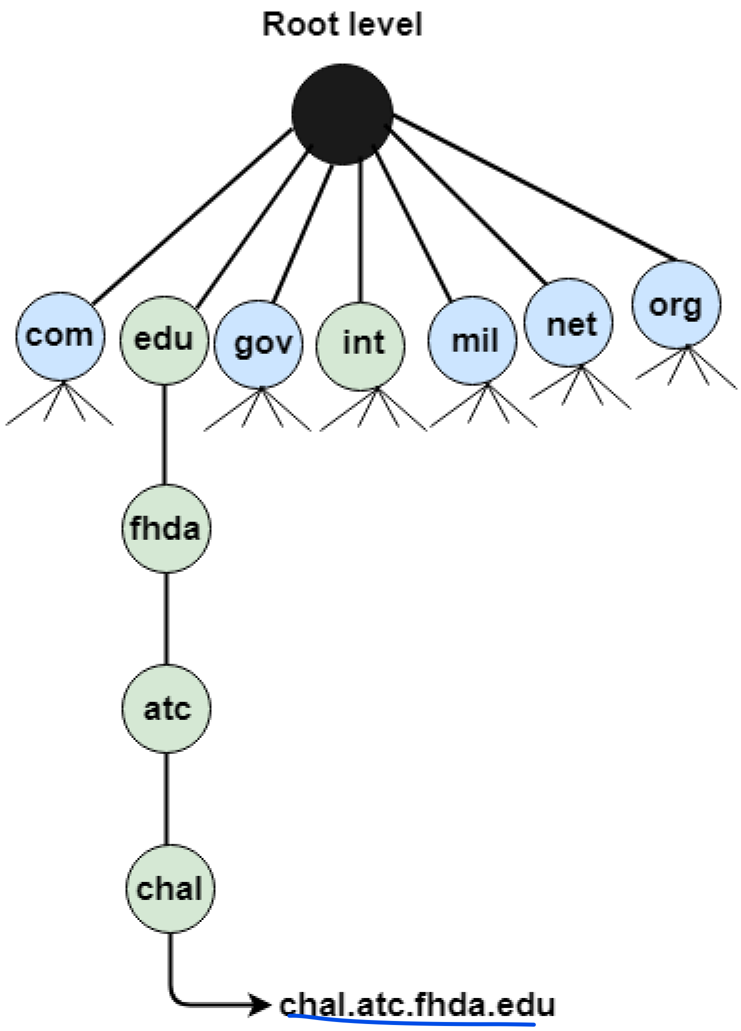
DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



Country Domain

.in

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

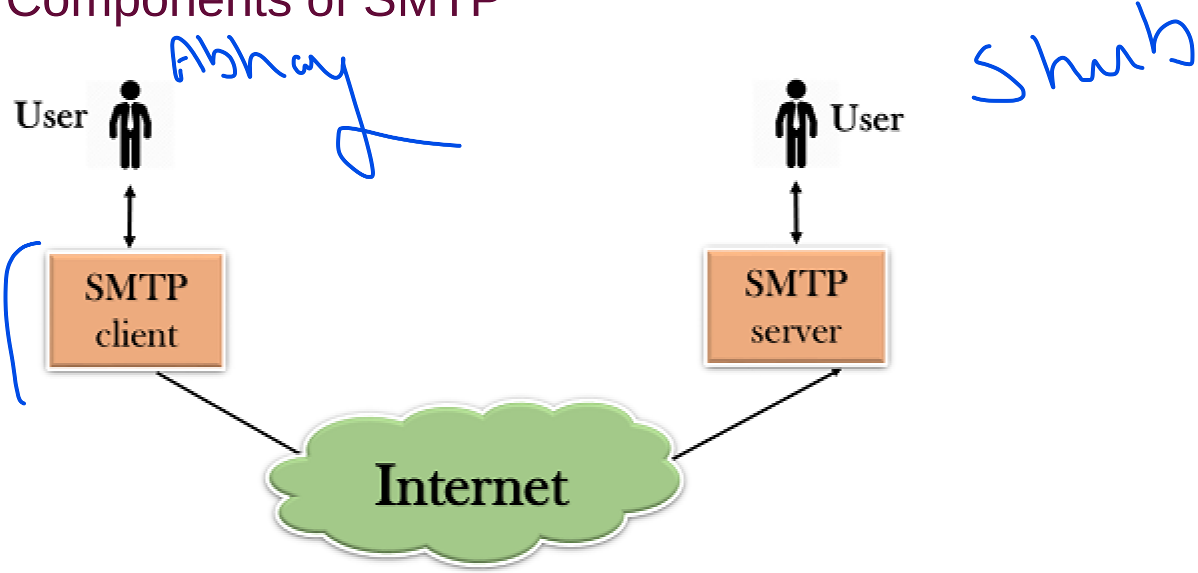
Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

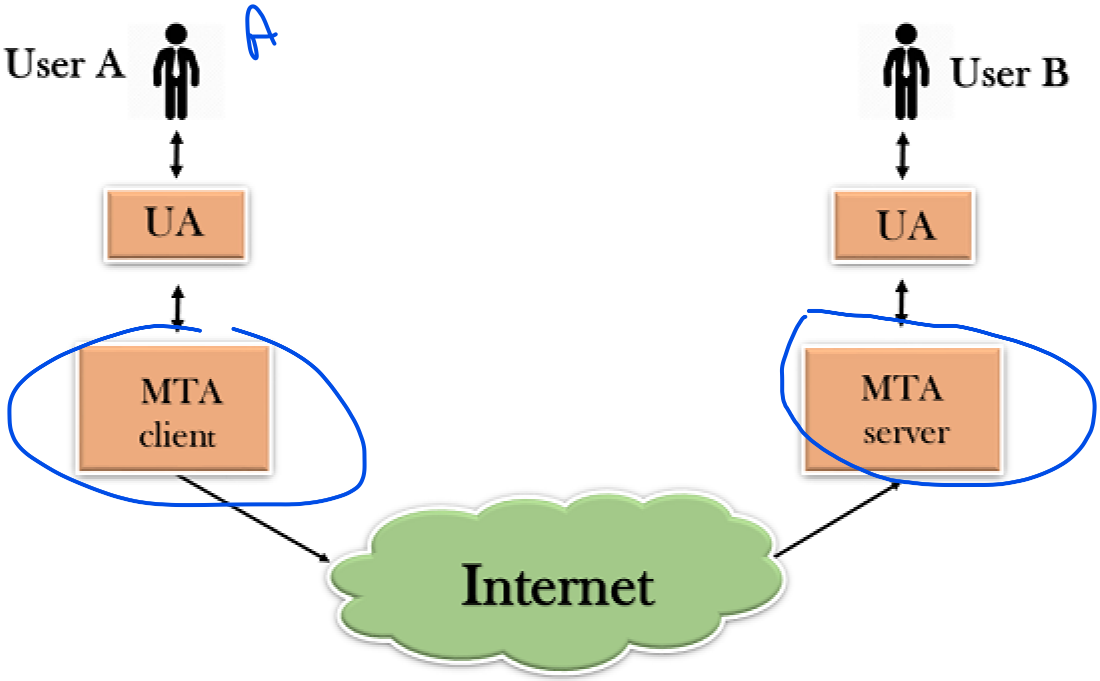
SMTP

- SMTP stands for Simple Mail Transfer Protocol.
- SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called **Simple Mail Transfer Protocol**.
- It is a program used for sending messages to other computer users based on e-mail addresses.
- It provides a mail exchange between users on the same or different computers, and it also supports:
  - It can send a single message to one or more recipients.
  - Sending message can include text, voice, video or graphics.
  - It can also send the messages on networks outside the internet.
- The main purpose of SMTP is used to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address. For example, if the recipient address is wrong, then receiving server reply with an error message of some kind.

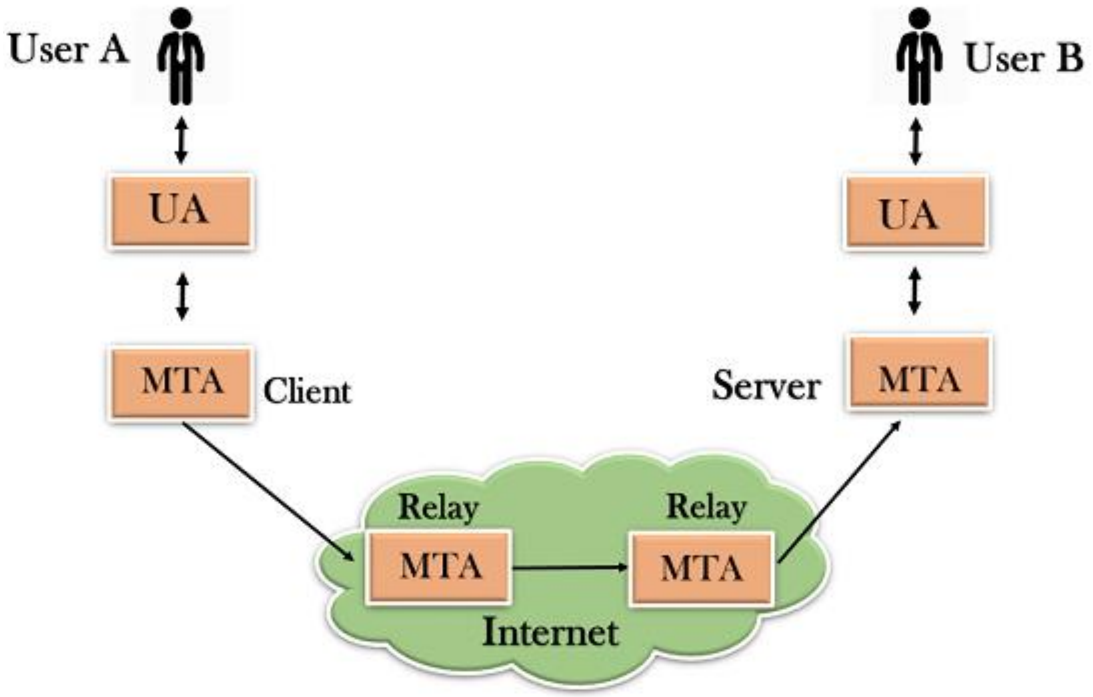
Components of SMTP



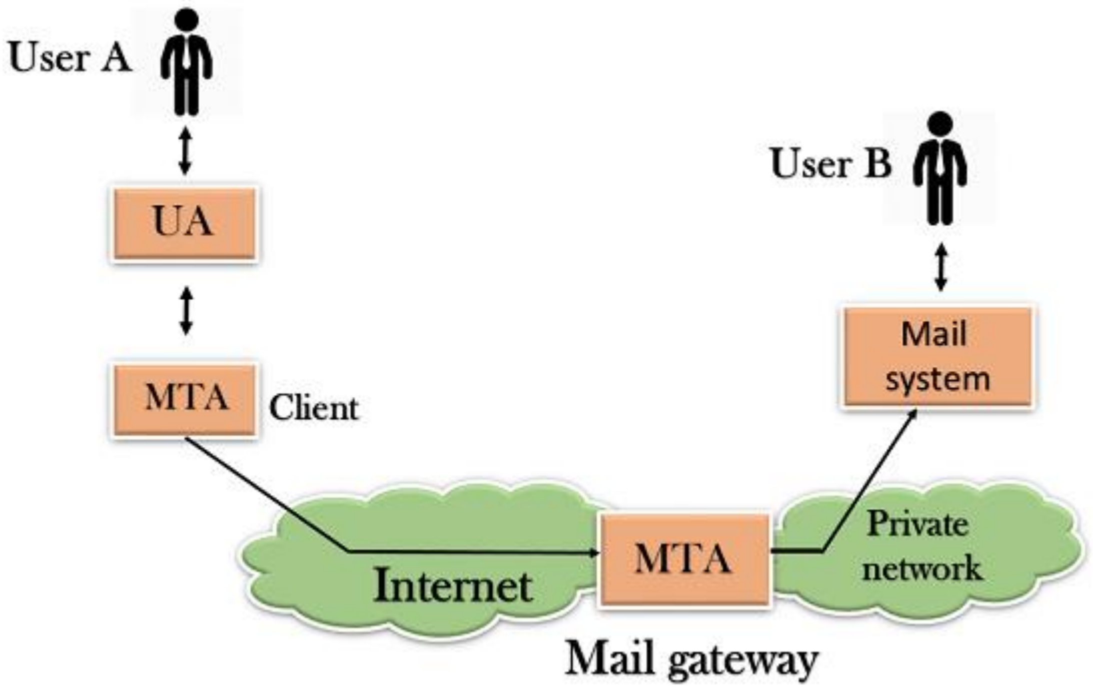
- First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.



- SMTP allows a more complex system by adding a relaying system. Instead of just having one MTA at sending side and one at receiving side, more MTAs can be added, acting either as a client or server to relay the email.



- The relaying system without TCP/IP protocol can also be used to send the emails to users, and this is achieved by the use of the mail gateway. The mail gateway is a relay MTA that can be used to receive an email.



Working of SMTP

1. **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and header is like an envelope that contains the recipient's address.
2. **Submission of Mail:** After composing an email, the mail client then submits the completed e-mail to the SMTP server by using SMTP on TCP port 25.
3. **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name. For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name.
   If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.
4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

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.

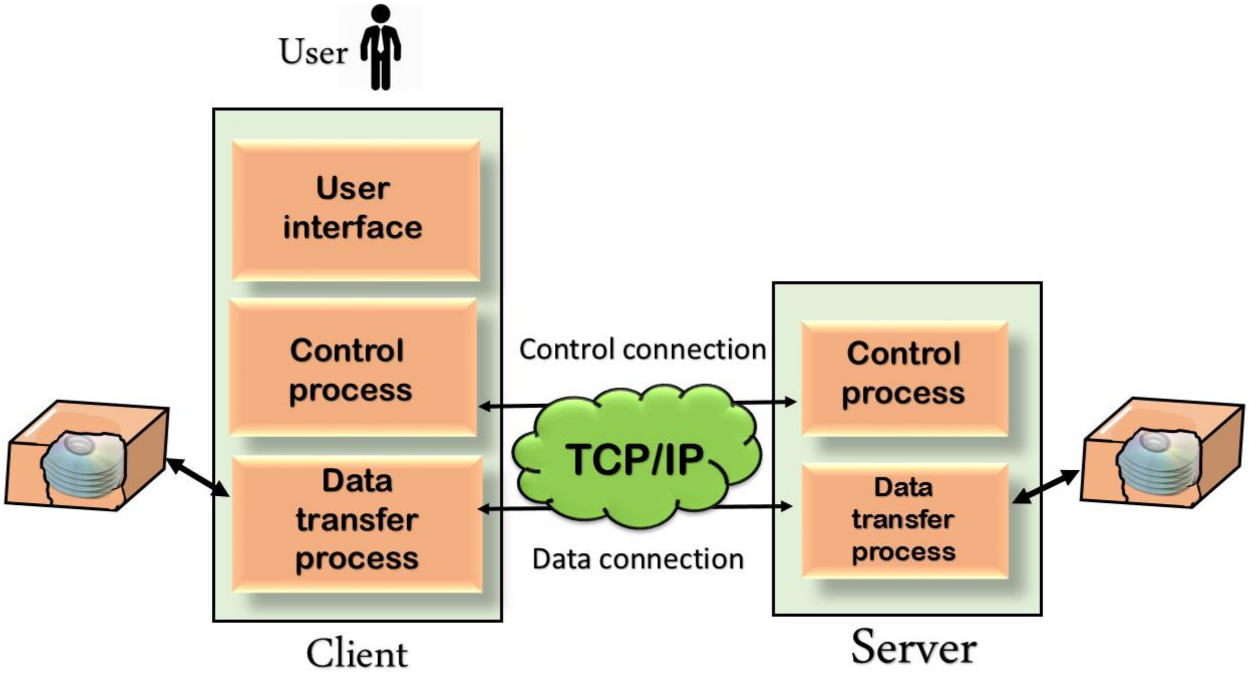
Objectives of FTP

- 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?

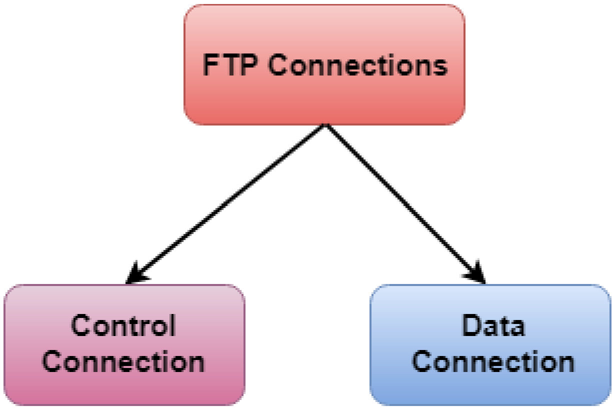
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures.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

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.

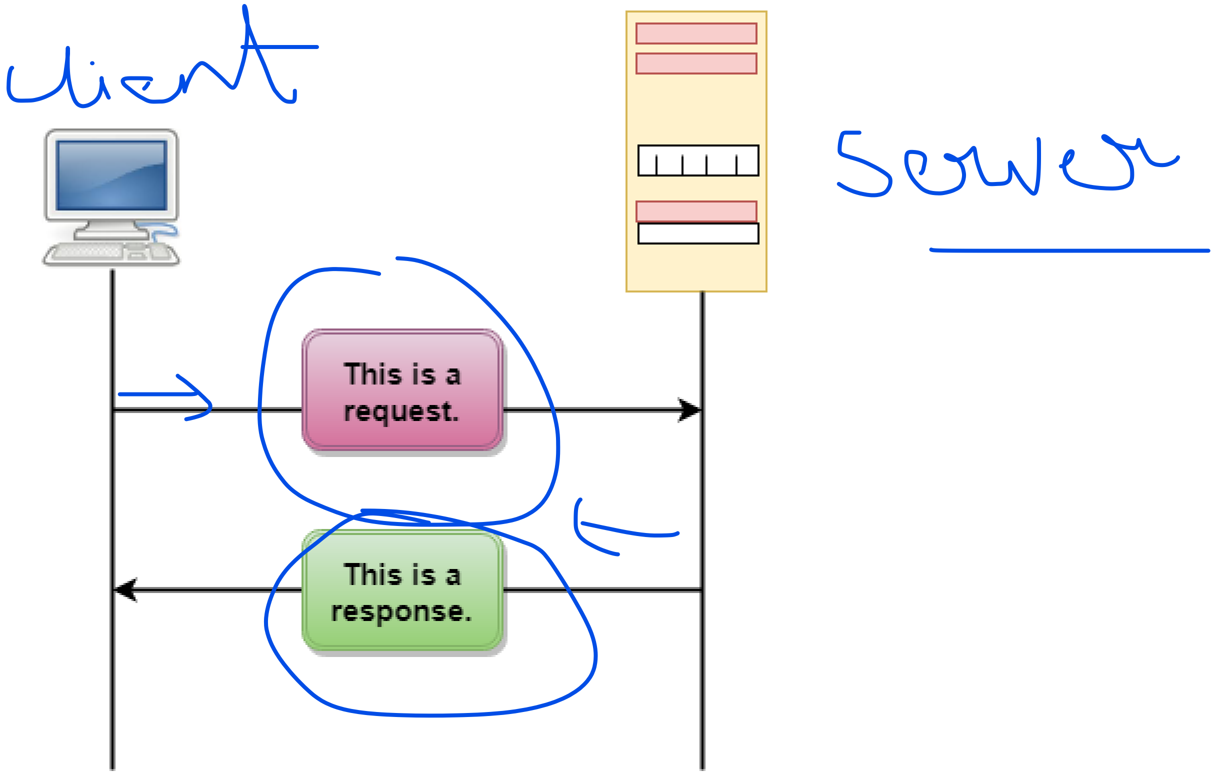
Features of HTTP:

- **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.

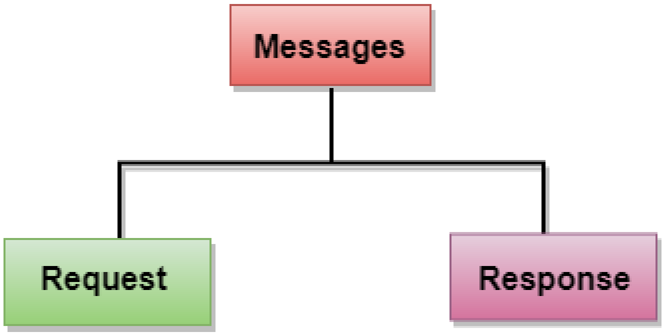
HTTP Transactions



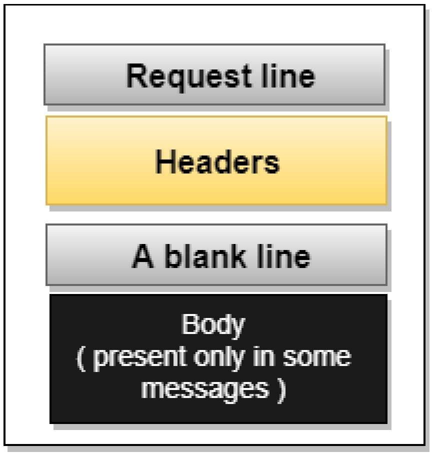
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages

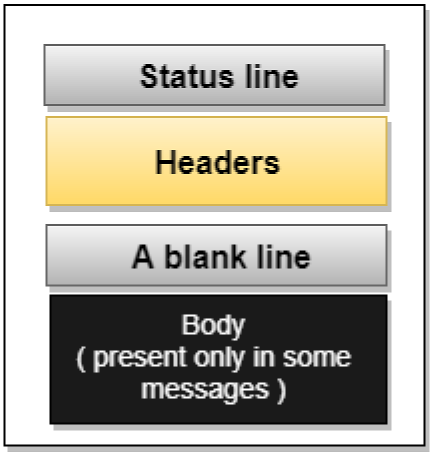
HTTP messages are of two types: request and response. Both the message types follow the same message format.



Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

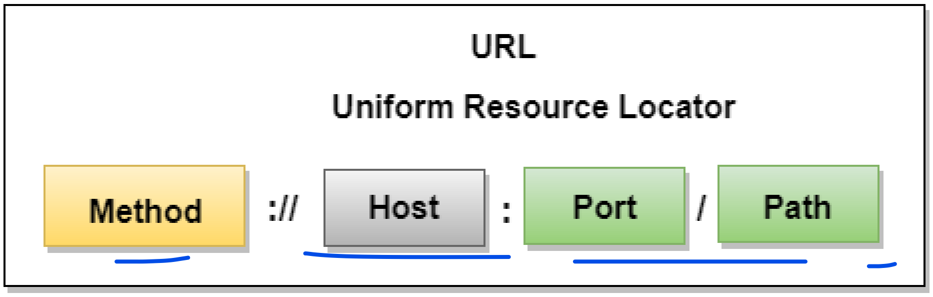


Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



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.



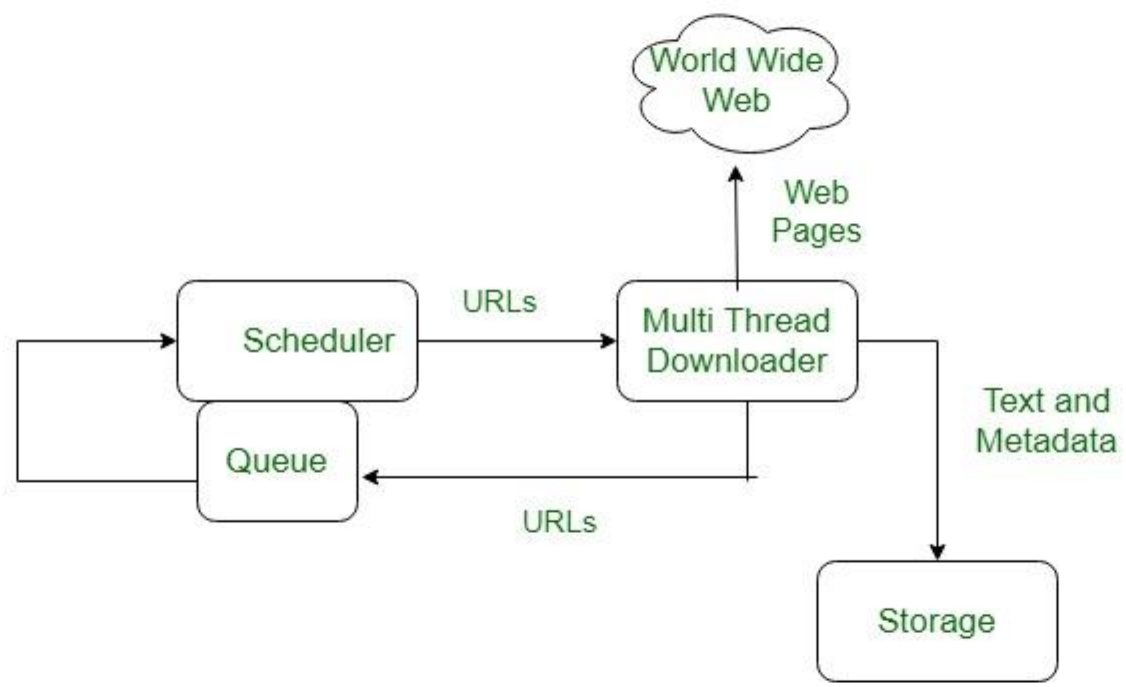
- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name 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.

World Wide Web (WWW)

-

The **World Wide Web** is abbreviated as WWW and is commonly known as the web. The WWW was initiated by CERN (European library for Nuclear Research) in 1989.
WWW can be defined as the collection of different websites around the world, containing different information shared via local servers(or computers).

History:
It is a project created, by Timothy Berner Lee in 1989, for researchers to work together effectively at CERN. is an organization, named the World Wide Web Consortium (W3C), which was developed for further development of the web. This organization is directed by Tim Berner's Lee, aka the father of the web.
System Architecture:
From the user's point of view, the web consists of a vast, worldwide connection of documents or web pages. Each page may contain links to other pages anywhere in the world. The pages can be retrieved and viewed by using browsers of which internet explorer, Netscape Navigator, Google Chrome, etc are the popular ones. The browser fetches the page requested interprets the text and formatting commands on it, and displays the page, properly formatted, on the screen.
The basic model of how the web works are shown in the figure below. Here the browser is displaying a web page on the client machine. When the user clicks on a line of text that is linked to a page on the abd.com server, the browser follows the hyperlink by sending a message to the abd.com server asking it for the page.



Here the browser displays a web page on the client machine when the user clicks on a line of text that is linked to a page on abd.com, the browser follows the hyperlink by sending a message to the abd.com server asking for the page.

Working of WWW:

The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP).

A Web browser is used to access web pages. Web browsers can be defined as programs which display text, data, pictures, animation and video on the Internet. Hyperlinked resources on the World Wide Web can be accessed using software interfaces provided by Web browsers. Initially, Web browsers were used only for surfing the Web but now they have become more universal. Web browsers can be used for several tasks including conducting searches, mailing, transferring files, and much more. Some of the commonly used browsers are Internet Explorer, Opera Mini, and Google Chrome.

Features of WWW:

- HyperText Information System
- Cross-Platform
- Distributed
- Open Standards and Open Source
- Uses Web Browsers to provide a single interface for many services
- Dynamic, Interactive and Evolving.
- “Web 2.0”

Components of the Web: There are 3 components of the web:

1. **Uniform Resource Locator (URL):** serves as a system for resources on the web.
2. **HyperText Transfer Protocol (HTTP):** specifies communication of browser and server.
3. **Hyper Text Markup Language (HTML):** defines the structure, organisation and content of a webpage.