# UNIT-V

## Transport Layer and Application Layer

### Transport Layer Introduction:

The network layer provides end-to-end packet delivery using data-grams or virtual circuits. The transport layer builds on the network layer to provide data transport from a process on a source machine to a process on a destination machine with a desired level of reliability that is independent of the physical networks currently in use. It provides the abstractions that applications need to use the network.

**Transport Entity**: The hardware and/or software which make use of services provided by the network layer, (within the transport layer) is called transport entity.

**Transport Service Provider:** Layers 1 to 4 are called Transport Service Provider.

**Transport Service User**: The upper layers i.e., layers 5 to 7 are called Transport Service User.

**Transport Service Primitives**: Which allow transport users (application programs) to access the transport service.

**TPDU (Transport Protocol Data Unit)**: Transmissions of message between 2 transport entities are carried out by TPDU. The transport entity carries out the transport service primitives by blocking the caller and sending a packet the service. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity. The task of the transport layer is to provide reliable, cost-effective data transport from the source machine to the destination machine, independent of physical network or networks currently in use.

### Services Provided to the Upper Layers

The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective data transmission service to its users, normally processes in the application layer. To achieve this, the transport layer makes use of the services provided by the network layer. The software and/or hardware within the transport layer that does the work is called the **transport entity**. The transport entity can be located in the operating system kernel, in a library package bound into network applications, in a separate user process, oreven on the network interface card.

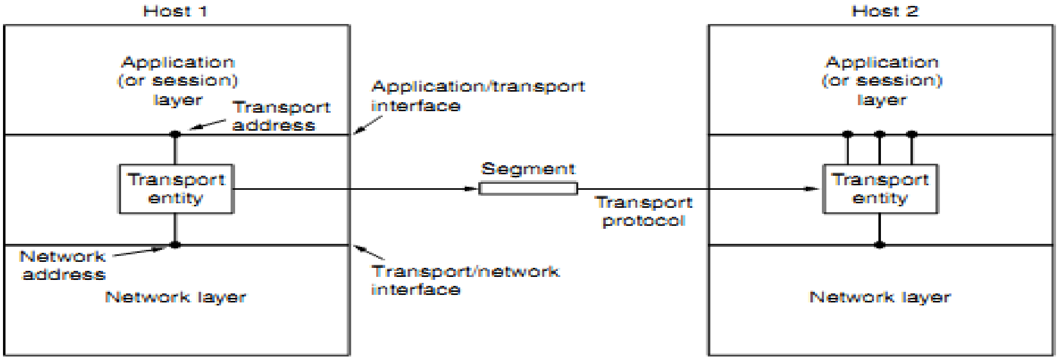


Fig 4.1: The network, Application and transport layer

**There are two types of network service**

- Connection-oriented
- Connectionless

Similarly, there are also two types of transport service. The connection-oriented transport service is similar tothe connection-oriented network service in many ways.

**In both cases, connections have three phases:**

- Establishment
- Data transfer
- Release.

**Transport Service Primitives**

To allow users to access the transport service, the transport layer must provide some operations to application programs, that is, a transport service interface. Each transport service has its own interface.

The transport service is similar to the network service, but there are also some important differences.

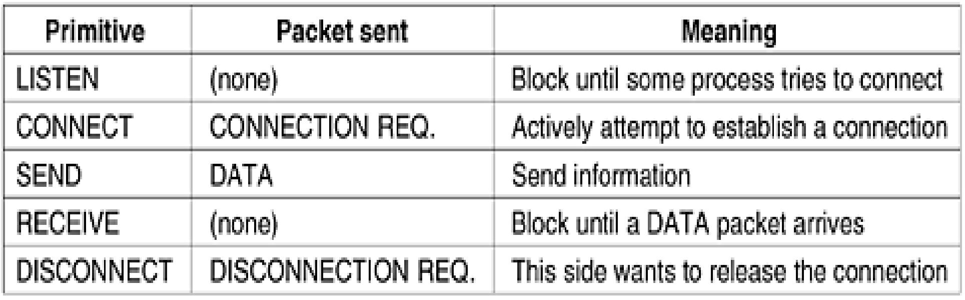
The main difference is that the network service is intended to model the service offered by real networks. Real networks can lose packets, so the network service is generally unreliable.

The (connection-oriented) transport service, in contrast, is reliable

As an example, consider two processes connected by pipes in UNIX. They assume the connection between them is perfect. They do not want to know about acknowledgements, lost packets, congestion, or anything like that. What they want is a 100 percent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other.

A second difference between the network service and transport service is whom the services are intended for. The network service is used only by the transport entities. Consequently, the transport service must be convenient and easy to use.

Table:4.1 - The primitives for a simple transport service.



| Primitive | Packet sent | Meaning |
|---|---|---|
| LISTEN | (none) | Block until some process tries to connect |
| CONNECT | CONNECTION REQ. | Actively attempt to establish a connection |
| SEND | DATA | Send information |
| RECEIVE | (none) | Block until a DATA packet arrives |
| DISCONNECT | DISCONNECTION REQ. | This side wants to release the connection |

Eg: Consider an application with a server and a number of remote clients.

The server executes a “LISTEN” primitive by calling a library procedure that makes a System call to block the server until a client turns up.

When a client wants to talk to the server, it executes a “CONNECT” primitive, with “CONNECTION REQUEST” TPDU sent to the server.

When it arrives, the TE unblocks the server and sends a “CONNECTION ACCEPTED” TPDU back to the client.

When it arrives, the client is unblocked and the connection is established. Data can now be exchanged using “SEND” and “RECEIVE” primitives.

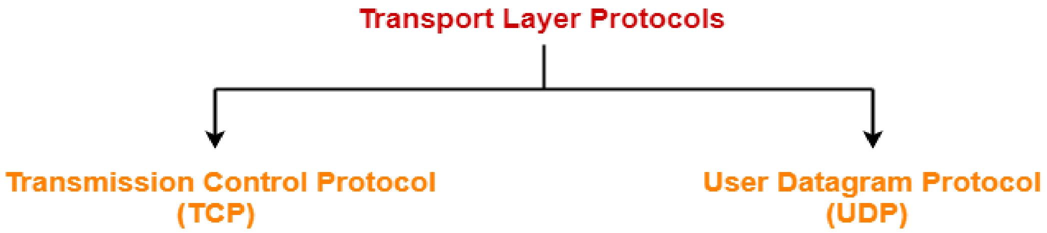
When a connection is no longer needed, it must be released to free up table space within the 2 transport entries, which is done with “DISCONNECT” primitive by sending “DISCONNECTION REQUEST”

**The elements of transport protocols are:**

ADDRESSING
Connection Establishment.
Connection Release.
Error control and flow control
Multiplexing.

Transport Layer Protocols-

There are mainly two transport layer protocols that are used on the Internet-


Transport Layer Protocols
Transmission Control Protocol (TCP)
User Datagram Protocol (UDP)

**Transmission Control Protocol-**

- TCP is short for Transmission Control Protocol.
- It is a transport layer protocol.
- It has been designed to send data packets over the Internet.
- It establishes a reliable end to end connection before sending any data.

**Characteristics Of TCP**-

- TCP is a reliable protocol.
- TCP is a connection oriented protocol.
- TCP handles both congestion and flow control.
- TCP ensures in-order delivery.
- TCP connections are full duplex.
- TCP works in collaboration with Internet Protocol.
- TCP can use both selective & cumulative acknowledgements.

- TCP is a byte stream protocol.
- TCP provides error checking & recovery mechanism.

## 3 Way Handshakes in TCP

- Three Way Handshake is a process used for establishing a TCP connection.

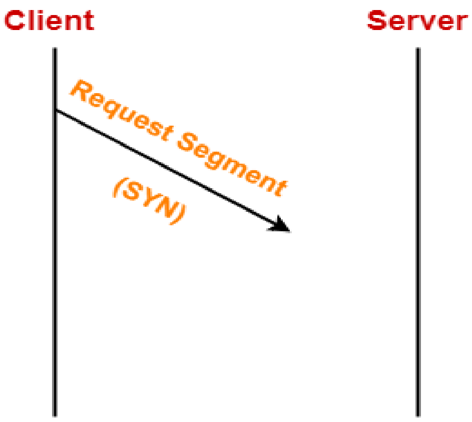
Consider-

- Client wants to establish a connection with the server.
- Before Three Way Handshake, both client and server are in closed state.

**TCP Handshake involves the following steps in establishing the connection-**

### Step-01: SYN-

- For establishing a connection,
- Client sends a request segment to the server.
- Request segment consists only of TCP Header with an empty payload.
- Then, it waits for a reply segment from the server.



**Request segment contains the following information in TCP header-**

- Initial sequence number
- SYN bit set to 1
- Maximum segment size
- Receiving window size

### 1. Initial Sequence Number-

- Client sends the initial sequence number to the server.
- It is contained in the sequence number field.
- It is a randomly chosen 32 bit value.

### 2. SYN Bit Set To 1-

- Client sets SYN bit to 1 which indicates the server-
- This segment contains the initial sequence number used by the client.
- It has been sent for synchronizing the sequence numbers.

### 3. Maximum Segment Size (MSS)-

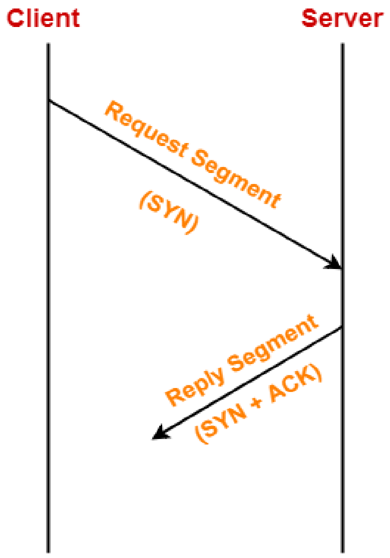
- Client sends its MSS to the server.
- It dictates the size of the largest data chunk that client can send and receive from the server.
- It is contained in the Options field.

### 4. Receiving Window Size-

- Client sends its receiving window size to the server.
- It dictates the limit of unacknowledged data that can be sent to the client.
- It is contained in the window size field.

### Step-02: SYN + ACK-

- After receiving the request segment,
- Server responds to the client by sending the reply segment.
- It informs the client of the parameters at the server side.



**Reply segment contains the following information in TCP header-**

- Initial sequence number
- SYN bit set to 1
- Maximum segment size
- Receiving window size
- Acknowledgment number
- ACK bit set to 1

### 1. Initial Sequence Number-

- Server sends the initial sequence number to the client.
- It is contained in the sequence number field.
- It is a randomly chosen 32 bit value.

### 2. SYN Bit Set To 1-

- Server sets SYN bit to 1 which indicates the client-
- This segment contains the initial sequence number used by the server.
- It has been sent for synchronizing the sequence numbers.

### 3. Maximum Segment Size (MSS)-

- Server sends its MSS to the client.
- It dictates the size of the largest data chunk that server can send and receive from the client.
- It is contained in the Options field.

### 4. Receiving Window Size-

- Server sends its receiving window size to the client.
- It dictates the limit of unacknowledged data that can be sent to the server.
- It is contained in the window size field.

### 5. Acknowledgement Number-

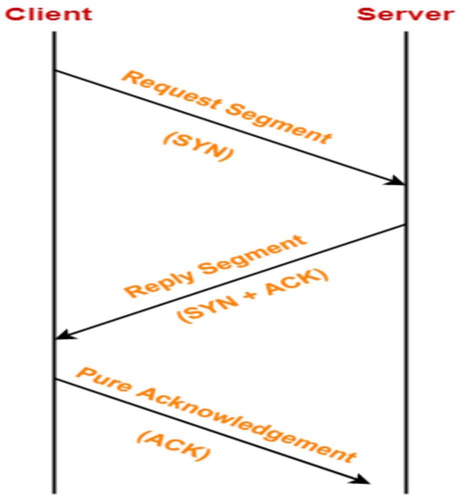
- Server sends the initial sequence number incremented by 1 as an acknowledgement number.
- It dictates the sequence number of the next data byte that server expects to receive from the client.

### 6. ACK Bit Set To 1-

- Server sets ACK bit to 1.
- It indicates the client that the acknowledgement number field in the current segment is valid.

### Step-03: ACK-

- After receiving the reply segment,
- Client acknowledges the response of server.
- It acknowledges the server by sending a pure acknowledgement.



3 Way Handshake

## TCP Congestion Control-

- TCP reacts to congestion by reducing the sender window size.

**The size of the sender window is determined by the following two factors-**

- Receiver window size
- Congestion window size

## 1. Receiver Window Size-

- Receiver window size is an advertisement of-

"How much data (in bytes) the receiver can receive without acknowledgement?"

- Sender should not send data greater than receiver window size.

Otherwise, it leads to dropping the TCP segments which causes **TCP Retransmission**.

So, sender should always send data less than or equal to receiver window size. Receiver dictates its window size to the sender through **TCP Header**.

## 2. Congestion Window-

- Sender should not send data greater than congestion window size.
- Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.

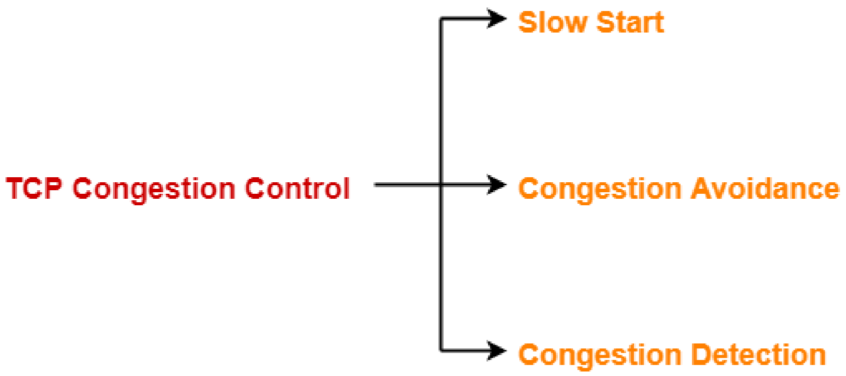
So, sender should always send data less than or equal to congestion window size. Different variants of TCP use different approaches to calculate the size of congestion window.

Congestion window is known only to the sender and is not sent over the links. So, always-

Sender window size = Minimum (Receiver window size, Congestion window size)

## TCP Congestion Policy-

TCP's general policy for handling congestion consists of following three phases-



- ➢ Slow Start
- ➢ Congestion Avoidance
- ➢ Congestion Detection

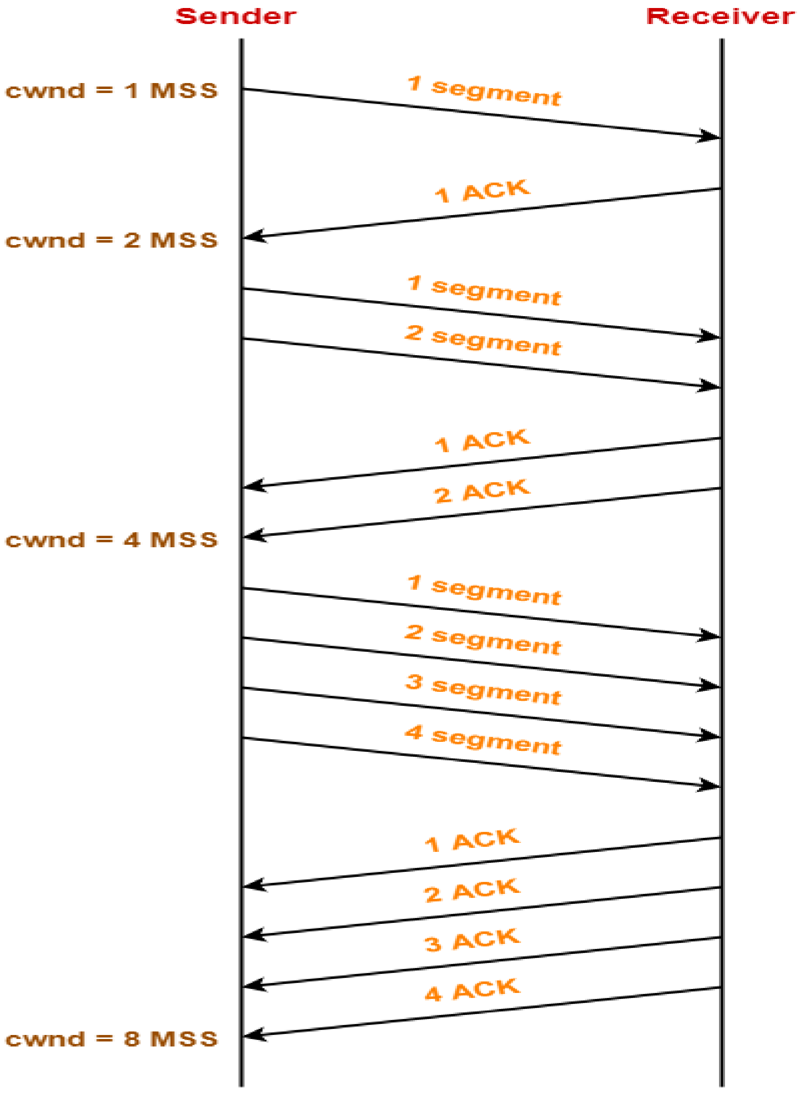
## 1. Slow Start Phase-

- ✓ Initially, sender sets congestion window size = Maximum Segment Size (1 MSS).
- ✓ After receiving each acknowledgment, sender increases the congestion window size by 1 MSS.
- ✓ In this phase, the size of congestion window increases exponentially.

The followed formula is-

| Congestion window size = Congestion window size + Maximum segment size |
|---|

This is shown below-



(cwnd = congestion window size)

After 1 round trip time, congestion window size = $(2)^1$ = 2 MSS

After 2 round trip time, congestion window size = $(2)^2$ = 4 MSS

After 3 round trip time, congestion window size = $(2)^3$ = 8 MSS and so on.

This phase continues until the congestion window size reaches the slow start threshold.

Threshold= Maximum number of TCP segments that receiver window can accommodate / 2

= (Receiver window size / Maximum Segment Size) / 2

## 2. Congestion Avoidance Phase-

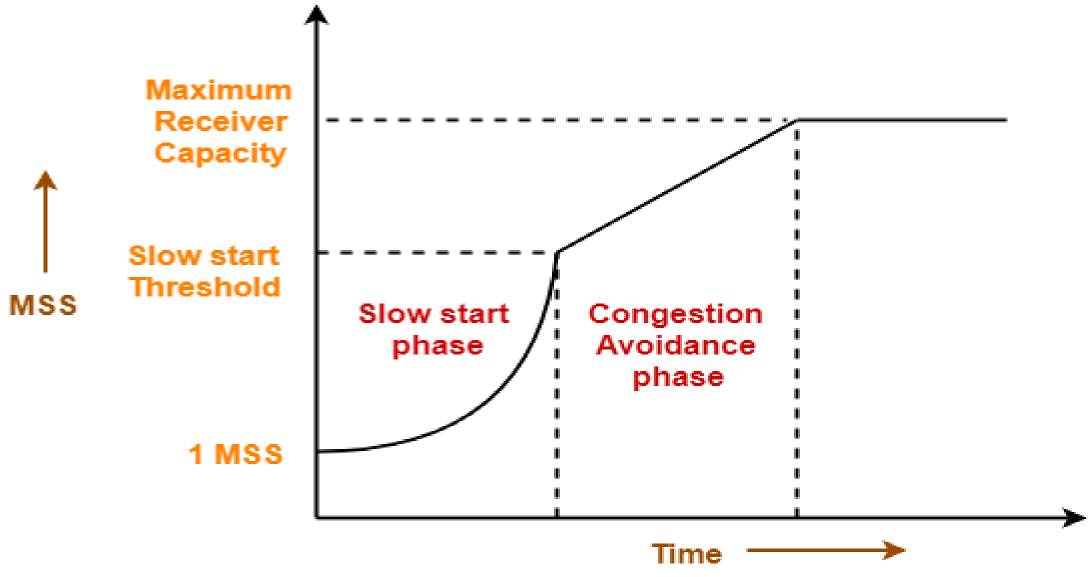
After reaching the threshold,

- ✓ Sender increases the congestion window size linearly to avoid the congestion.
- ✓ On receiving each acknowledgement, sender increments the congestion window size by 1.

The followed formula is-

Congestion window size = Congestion window size + 1

- ✓ This phase continues until the congestion window size becomes equal to the receiver window size.



## 3. Congestion Detection Phase-

When sender detects the loss of segments, it reacts in different ways depending on how the loss is detected-

**Case-01: Detection On Time Out-**

- ✓ Time Out Timer expires before receiving the acknowledgement for a segment.
- ✓ This case suggests the stronger possibility of congestion in the network.
- ✓ There are chances that a segment has been dropped in the network.

**Reaction-**

In this case, sender reacts by-

- ✓ Setting the slow start threshold to half of the current congestion window size.
- ✓ Decreasing the congestion window size to 1 MSS.
- ✓ Resuming the slow start phase.

**Case-02: Detection On Receiving 3 Duplicate Acknowledgements-**

- ✓ Sender receives 3 duplicate acknowledgements for a segment.
- ✓ This case suggests the weaker possibility of congestion in the network.
- ✓ There are chances that a segment has been dropped but few segments sent later may have reached.

**Reaction-**

In this case, sender reacts by-

- ✓ Setting the slow start threshold to half of the current congestion window size.
- ✓ Decreasing the congestion window size to slow start threshold.
- ✓ Resuming the congestion avoidance phase.

**Error control in TCP**

- ✓ Acknowledgement number
- ✓ Re transmission
- ✓ Checksum
- ✓ Sequence number
- ✓ Acknowledgment Number:

    - ➢ In TCP for every data/segment send to the other end, it requires an acknowledgment in return. The acknowledgment number is nothing but the sequence number of the next bytes the receiver expects to receive.
    - ➢ In the case of TCP, there is a cumulative acknowledgment number that is acknowledgment number is not send for each byte rather it is sent for a group of bytes that is called a segment.

For example:

If the **acknowledgment number** is 1635, means all the bytes before this number are reached and the receiver expects bytes with 1635 as the next sequence number.

If an acknowledgment number is not received, TCP automatically re-transmits the data(segment) and waits a longer amount of time.

Note:The maximum time it can keep trying re transmission is 4 to 10 mins, depending upon implementation.

TCP does not guarantee that data will be received at other end, its just that it provides reliable delivery of data or reliable notification of failure.

There is nothing called segment number in TCP, rather each segment is a collection of bytes and each bytes is associated with sequence number.

Thus acknowledgment number provides reliability to the TCP

**Retransmission:**

- ✓ This is the heart of the TCP when it comes to reliability that is error control mechanism. If the packets is lost or damaged or corrupted or the ack itself is lost, TCP retransmits the data.
- ✓ Retransmission takes place in two scenarios:
- ✓ Re transmission timer expires: that is it does not get the ack for the send bytes within stipulated time.
- ✓ Fast re transmission: This happens when the sender receives three duplicate ACK, in this segment is re transmitted even before RTO.
- ✓ Checksum: This is one of the features of TCP along with acknowledgment and retransmission which is used for error control mechanisms in TCP.

**The checksum** is calculated on three fields:

- ➢ TCP header
- ➢ TCP Body
- ➢ Pseudo IP header

The most surprising field out of the above three is the pseudo IP header because the IP header is below the transport header and the values of its fields keep changing when the packet traverses the network. So the IP fields are used for checksum are those which are constant in the network that is:
Source IP address

- ✓ Destination IP address
- ✓ Protocol
- ✓ TCP segment size
- ✓ Fixed of 8 bits
- ✓ The total size of the pseudo-header is 12 bytes.

Once the checksum is calculated with all the three fields, it is placed in the checksum field of TCP header and send to the receiver side and even calculates the checksum on the same fields and compares with what it received from the sender.

If the checksum happens not to be same, segment is considered as corrupted and ack is not being send and TCP autocratically retransmits the same.

Note: Pseudo header is not forwarded along with the network, it is discarded once checksum is calculated.

**Sequence number:**

TCP associates a sequence number with each bytes it send. For example if the application writes a data of size say 2048 bytes, TCP would send this in two segments where the first segment carries bytes ranging from 1-1024 and the second 1025-2048.

**Significance of sequence number:**

Reassembly of packet at receiver side:

If the segments arrive out of order, the receiving TCP will reorder the two segments on the basis of sequence number before passing it to the application. Hence in TCP segments never reach out of order.

**Discard of duplicate data:**

If TCP receives duplicate data may be because of lost acknowledgment or delay in receiving ack because of congestion, the receiving TCP can detect the duplicate data with the help of sequence number and discards the data.

Retransmission of lost or corrupted or for damaged data:

Segments which are lost or damaged are re-send on the basis of the sequence number.

**Flow Control in TCP**

TCP provides a mechanism called flow control by which it always tells its peer how many bytes of data it is willing to accept. This is called advertised window which reflects the buffer size of the receiver side so that sender cannot overflow the receiver buffer.

This is also called **windowing mechanism.**

**UDP Protocol-**

- UDP is short for **User Datagram Protocol**.
- It is the simplest transport layer protocol.
- It has been designed to send data packets over the Internet.
- It simply takes the datagram from the network layer, attaches its header and sends it to the user.

**Characteristics of UDP-**

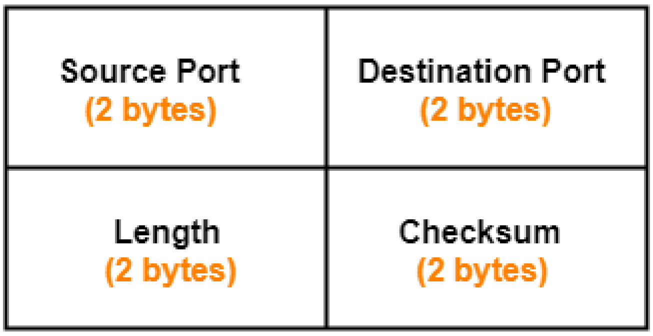
- ✓ It is a connectionless protocol.
- ✓ It is a stateless protocol.
- ✓ It is an unreliable protocol.
- ✓ It is a fast protocol.
- ✓ It offers the minimal transport service.
- ✓ It is almost a null protocol.
- ✓ It does not guarantee in order delivery.
- ✓ It does not provide congestion control mechanism.
- ✓ It is a good protocol for data flowing in one direction.

## Need of UDP-

- ✓ TCP proves to be an overhead for certain kinds of applications.
- ✓ The **Connection Establishment** Phase, **Connection Termination** Phase etc of TCP are time consuming.
- ✓ To avoid this overhead, certain applications which require fast speed and less overhead use UDP.

## UDP Header-

The following diagram represents the UDP Header Format-



| | |
|---|---|
| Source Port<br>(2 bytes) | Destination Port<br>(2 bytes) |
| Length<br>(2 bytes) | Checksum<br>(2 bytes) |

UDP Header

### 1. Source Port-

- ✓ Source Port is a 16 bit field.
- ✓ It identifies the port of the sending application.

### 2. Destination Port-

- ✓ Destination Port is a 16 bit field.
- ✓ It identifies the port of the receiving application.

### 3. Length-

- ✓ Length is a 16 bit field.
- ✓ It identifies the combined length of UDP Header and Encapsulated data.

Length = Length of UDP Header + Length of encapsulated data

### 4. Checksum-

- ✓ **Checksum** is a 16 bit field used for error control.
- ✓ It is calculated on UDP Header, encapsulated data and IP pseudo header.
- ✓ Checksum calculation is not mandatory in UDP.

Applications Using UDP-

Following applications use UDP-

- ✓ Applications which require one response for one request use UDP.
  Example- **DNS**.
- ✓ Routing Protocols like RIP and OSPF use UDP because they have very small amount of data to be transmitted.
- ✓ Trivial **File Transfer Protocol** (TFTP) uses UDP to send very small sized files.
- ✓ Broadcasting and multicasting applications use UDP.
- ✓ Streaming applications like multimedia, video conferencing etc use UDP since they require speed over reliability.
- ✓ Real time applications like chatting and online games use UDP.
- ✓ Management protocols like SNMP (Simple Network Management Protocol) use UDP.
- ✓ Bootp / DHCP uses UDP.
- ✓ Other protocols that use UDP are- Kerberos, Network Time Protocol (NTP), Network News Protocol (NNP), Quote of the day protocol etc.

## Application Layer

### 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.

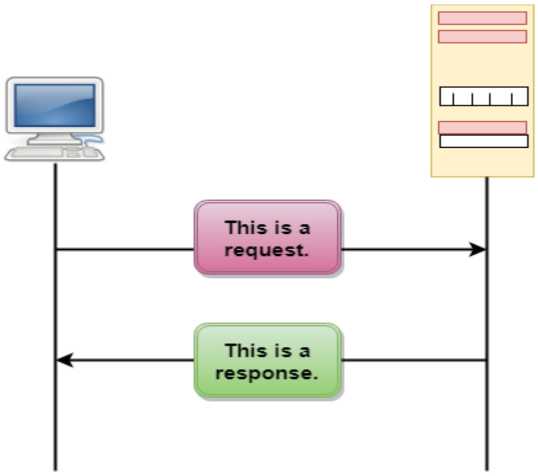
**Features of HTTP:**

**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.

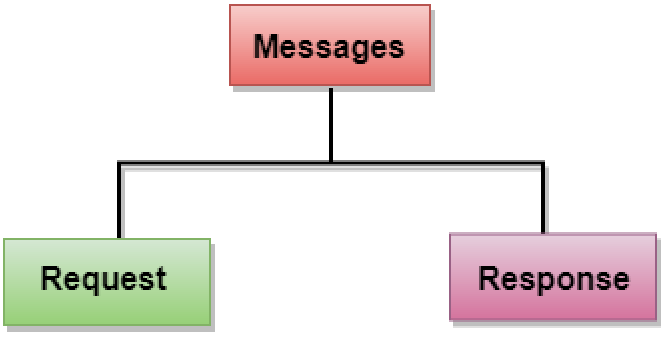
**HTTP Transactions**



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.

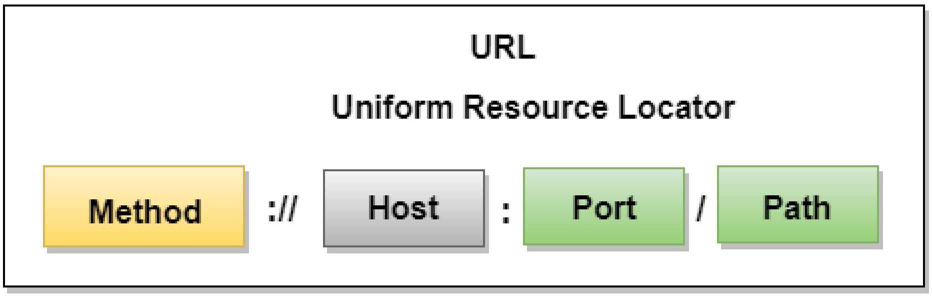
**Response Message**: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body

**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.

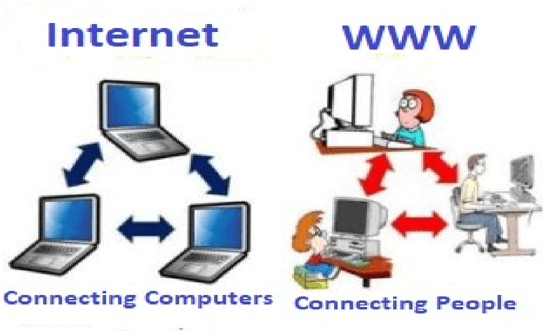
**WWW**

- ✓ The World Wide Web was invented by a British scientist, Tim Berners-Lee in 1989
- ✓ World Wide Web, which is also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc.
- ✓ Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc

- ✓ The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP.
- ✓ These links are electronic connections that link related pieces of information so that users can access the desired information quickly. Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

- ✓ A web page is given an online address called a Uniform Resource Locator (URL). A particular collection of web pages that belong to a specific URL is called a website, e.g., www.facebook.com, www.google.com, etc

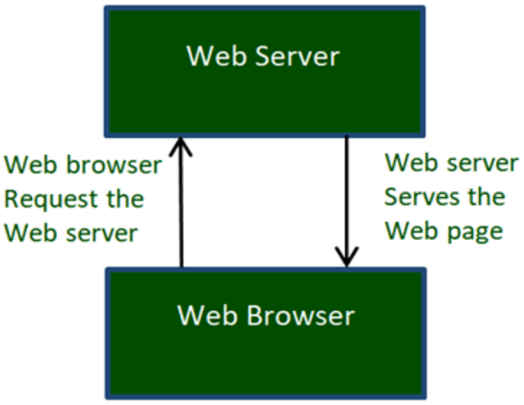
**Difference between World Wide Web and Internet:**



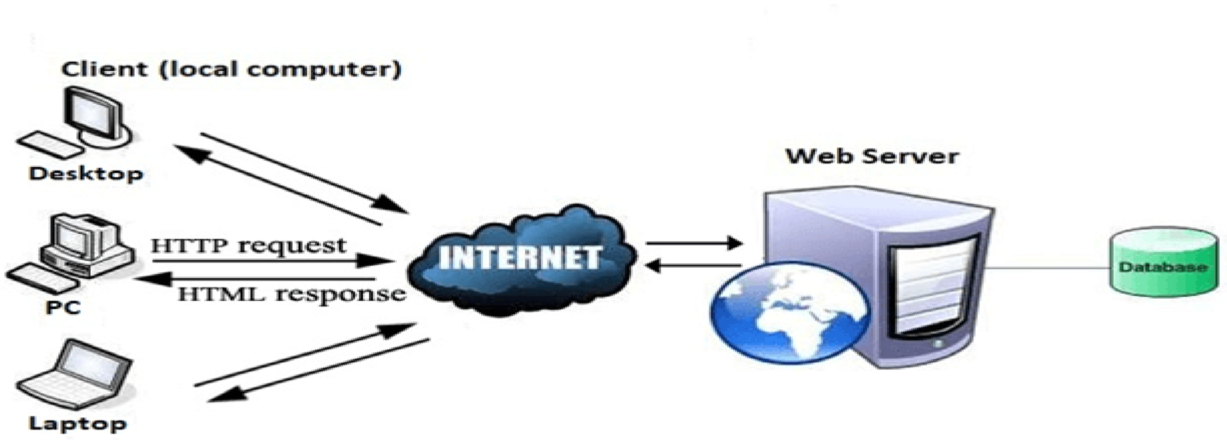
Some people use the terms 'internet' and 'World Wide Web' interchangeably. They think they are the same thing, but it is not so. Internet is entirely different from WWW. It is a worldwide network of devices like computers, laptops, tablets, etc. It enables users to send emails to other users and chat with them online. For example, when you send an email or chatting with someone online, you are using the internet.

But, when you have opened a website like google.com for information, you are using the World Wide Web; a network of servers over the internet. You request a webpage from your computer using a browser, and the server renders that page to your browser. Your computer is called a client who runs a program (web browser), and asks the other computer (server) for the information it needs.

**How the World Wide Web Works?**


Web Server
Web browser Request the Web server
Web server Serves the Web page
Web Browser

The Web works as per the internet's basic client-server format as shown in the following image. The servers store and transfer web pages or information to user's computers on the network when requested by the users. A web server is a software program which serves the web pages requested by web users using a browser. The computer of a user who requests documents from a server is known as a client. Browser, which is installed on the user' computer, allows users to view the retrieved documents.



**Advantages of WWW**

- It mainly provides all the information for Free.
- Provides rapid Interactive way of Communication.
- It is accessible from anywhere.
- It has become the Global source of media.
- It mainly facilitates the exchange of a huge volume of data.

**Disadvantages of WWW**

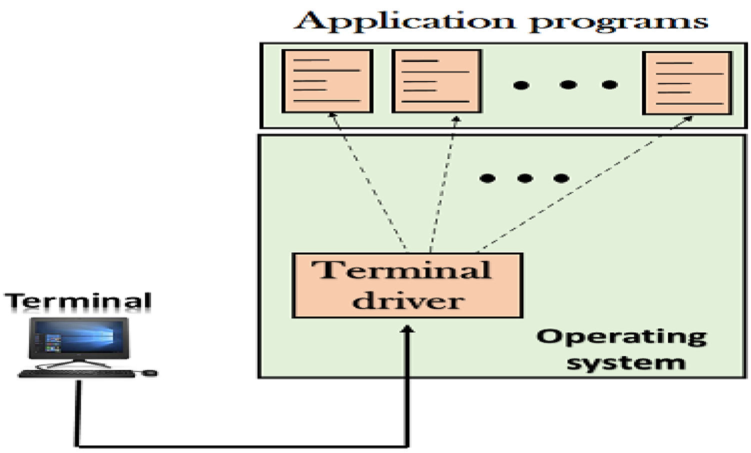
- It is difficult to prioritize and filter some information.
- There is no guarantee of finding what one person is looking for.
- There occurs some danger in case of overload of Information.
- There is no quality control over the available data.
- There is no regulation.

# Telnet

- ✓ The main task of the internet is to provide services to users.
  For example, users want to run different application programs at the remote site and transfers a result to the local site.
  This requires a client-server program such as FTP, SMTP. But this would not allow us to create a specific program for each demand.
- ✓ The better solution is to provide a general client-server program that lets the user access any application program on a remote computer. Therefore, a program that allows a user to log on to a remote computer. A popular client-server program Telnet is used to meet such demands. Telnet is an abbreviation for Terminal Network.
- ✓ Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

**There are two types of login:**

**Local Login**

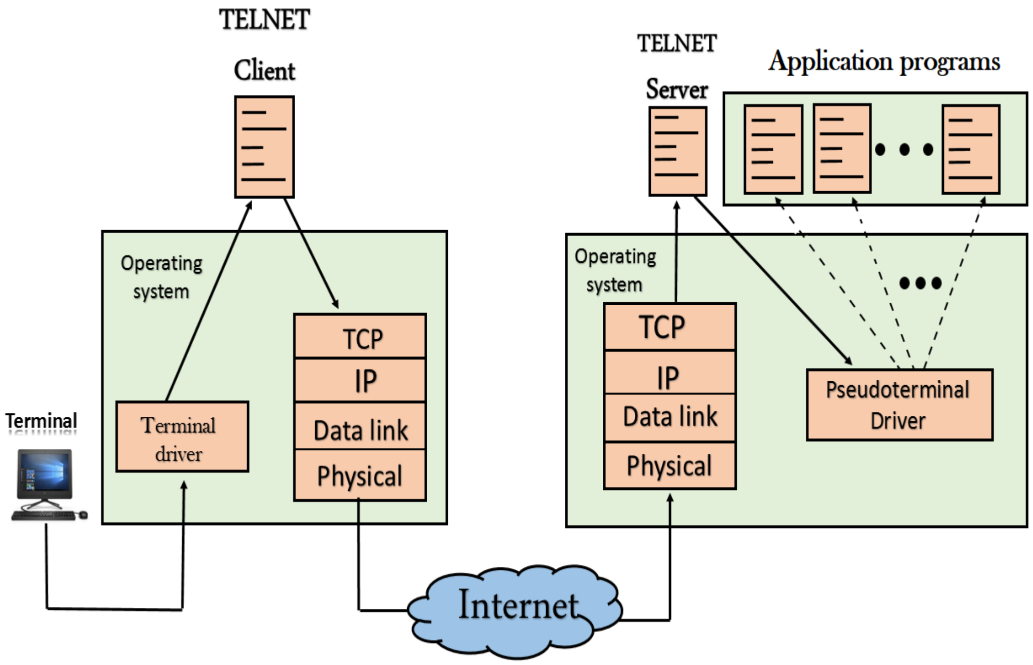


When a user logs into a local computer, then it is known as local login.

When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes these characters to the operating system which in turn, invokes the desired application program.

However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters has special meanings such as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.

**Remote login**



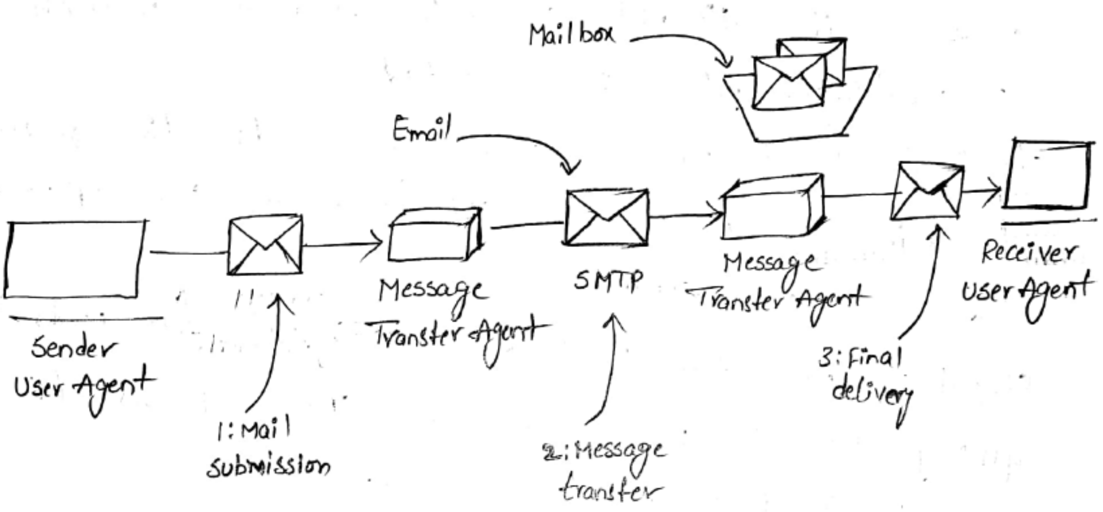
When the user wants to access an application program on a remote computer, then the user must perform remote login.

## Electronic Mail Architecture

- The architecture of email system consist of two kinds of subsystems : the User Agent
the Message Transfer Agent.

<u>User Agent</u> :- It allows people to read & send email

<u>Message transfer Agents</u> : It moves the messages from source to destination. They run in the background on mail server machines.



<u>Architecture</u> of <u>email system</u>

@ <u>Mail Submission</u> : The user agent is a program that provides an interface that allows the users to interact with the email system.

- Here the user can compose messages, replies to messages & organize messages.
- The act of sending new messages into the mail system for delivery is called mail submission.

(b) · Message Transfer : The message transfer agent at the sender side forwards the email to the message transfer agent at the receiver side by using SMTP (Simple Mail Transfer Protocol). This is the message Transfer step.

(c) Final delivery : At the receiver side, the user agent and the message transfer agent are linked using mailboxes. They store email that is received for a user. They are maintained by mail servers.

- The retrieval of mail from the mailboxes is the final delivery.

Email message format.

- It consists of two parts envelope and message.

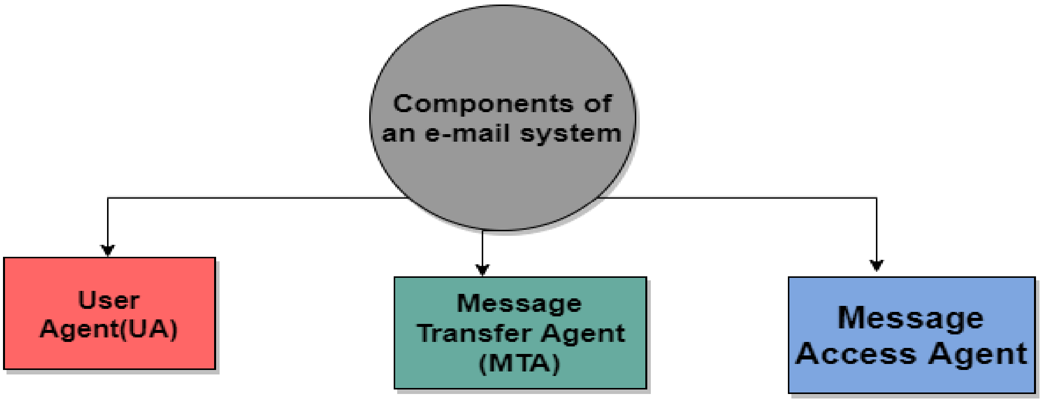
(a) Envelope : It contains all the information needed for ~~transfo~~ transporting the message such as destination address, priority and security level.

(b) Message : It consist of two separate parts : the header and the body.

- header : It contains the control information for user agent.
- ~~Message~~ body :- It contains the original message for the receiver.

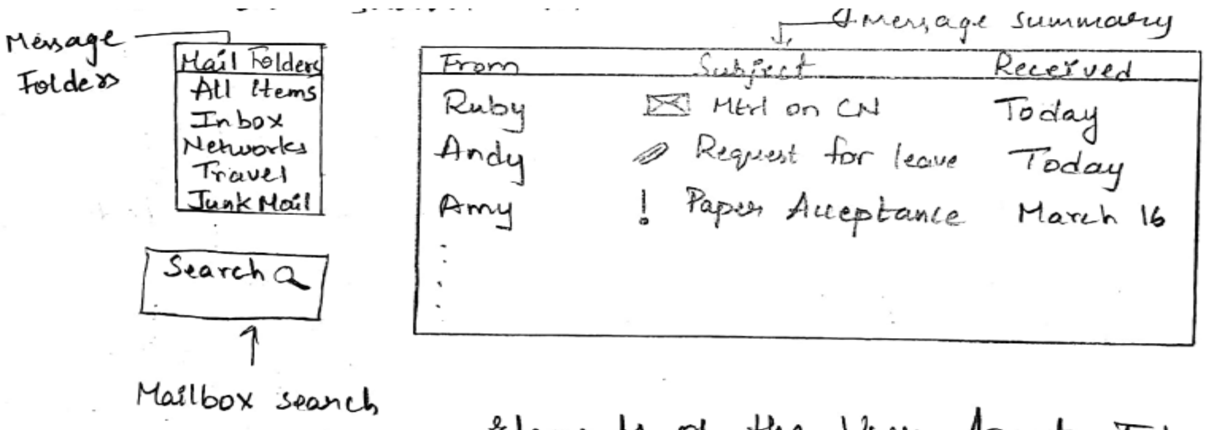
## Components of E-mail System

The basic Components of an Email system are as follows:



### 1. User Agent (UA)

- A user agent is a program that accepts a Variety of commands for composing, receiving & replying to messages.
- There are many popular user agents including Google gmail, Mozilla Thunderbird & Apple Mail.
- Most user agents have a menu or icon-driven graphical interface that requires a mouse or a touch interface on smaller mobile devices.
- The typical elements of a user agent interface are as shown in the diagram.



| From | Subject | Received |
|---|---|---|
| Ruby | ✉ Mtrl on CN | Today |
| Andy | Request for leave | Today |
| Amy | ! Paper Acceptance | March 16 |

Elements of the User Agent Interface

- When a user agent is started, it will usually present a summary of the messages in the User's mailbox.
- The user agent present the summary as follows:

. it uses From, Subject and Received fields to display 27 who sent the message, what it is about and when it was received.
- People who fail to include a subject field often discover that responses to their emails tend not to get the highest priority.
- The icons present near the subject might indicate unread mail (the envelope), attached mtrl (the paperclip) & important mail (the exclamation point)
- Many sorting orders are possible. The most common is to order messages based on the time. that they were received, most recent displayed first
- User agents provide a short preview of a message to help users decide when to read further.
- After a message has been read, the user can decide what to do with it. This is called message disposition.

- It includes deleting the message, sending a reply, forwarding the message to another users & keeping the message for later reference.

## Message Formats:

### RFC 5322 - The Internet Message Format:

| Header | Meaning |
|---|---|
| To: | Email address of primary recipient |
| Cc: | Email address of Secondary recipient |
| Bcc: | Email address for Blind carbon copies |
| From: | Person or people who created the message |
| Sender: | Email address of the actual sender. |
| Received: | Line added by each transfer agent |

- To: Email add of primary recipient.
- Cc: Email add of secondary recipient
  Cc stands for Carbon Copy
  - Email addresses listed here will receive a copy of email that we sent to the people listed in the To: field.
  - Everyone listed under the Cc field will see everyone's email addresses that are under the To and Cc field.
- Bcc: Bcc stands for Blind Carbon Copy.
  - Email addresses listed here will receive a copy of email that you sent to the people listed in the To: field.

- Everyone listed under the Cc field will see everyone's email address that are listed under the To & Cc field but will not see the addresses listed in Bcc field.
- Each person listed on the Bcc field will not see the email address of other recipients.

- From: It tells who wrote the message.
- Sender :- It tells who sent the message
- Received : It is added by each message transfer agent. It contains the agent's identity, the date & time message was received & other information that can be used for debugging the routing system.

- Return-path : It is added by the final message transfer agent and was intended to tell how to get back to the sender.

In addition to the fields mentioned above, RFC 5322 messages also contain a variety of header fields used by the user agents or human recipients. The most common ones are listed below.

| Header | Meaning |
|---|---|
| Date : | the date & time the message was sent |
| Reply-To : | Email add to which replies should be sent |
| Message-Id : | Unique number for referencing the msg later |
| In-Reply-To : | Message-Id of the message which this is a reply. |
| References : | Other relevant message-Ids. |
| keywords : | User chosen keywords. |
| Subject : | Short summary of the message for the one-line display |

## Message Transfer :-

### SMTP and Extensions :-

- SMTP is a simple ASCII protocol. Using ASCII text makes protocols easy to develop, test & debug.
- Email is delivered by establishing a TCP connection with port number : 25 b/w the sending machine and receiving machine.
- After establishing the TCP connection to port 25, the sending machine operates as client & the receiving machine operates as server.
- Before sending email, the client announces whom the email is coming from & whom it is going to
- If such a recipient exists at the destination, the

server gives the client the go-ahead to send the message.

- Then the client sends the message & the server acknowledges it.
- No checksums are needed bcz TCP provide a reliable connection.
- When all the email has been exchanged in both directions, the connection is released.

Disadv of SMTP:

- It doesn't include authentication.

- DNS contains multiple types of records including the MX or mail exchanger record.
- So, a DNS query is sent to get the MX records of the receiver domain.
- This query returns an ordered list of IP addresses of one or more mail server.

## Message Transfer :

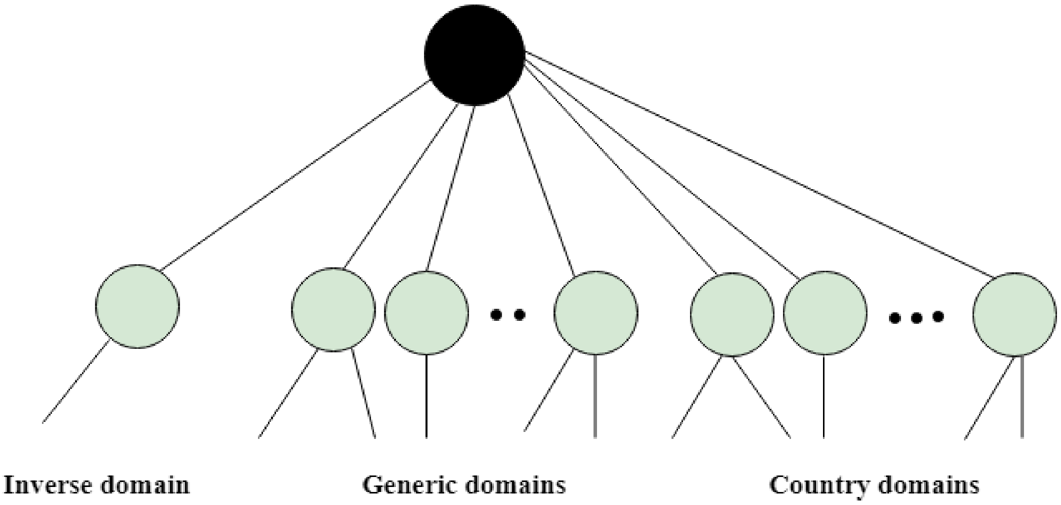
- Once the sending mail transfer agent receives a message from the user agent, it will deliver it to the receiving mail transfer agent using SMTP.
- To do this, the sender uses the destination address
- the Message transfer agents runs on the mail server machines.
- So, we should determine the correct mail server to contact, for this purpose DNS is used.
- DNS contains multiple types of records including 34 the MX or mail exchanger record.
- So, a DNS query is sent to get the MX records of the receiver domain.
- This query returns an ordered list of IP addresses of one or more mail server.

## DNS

- 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.

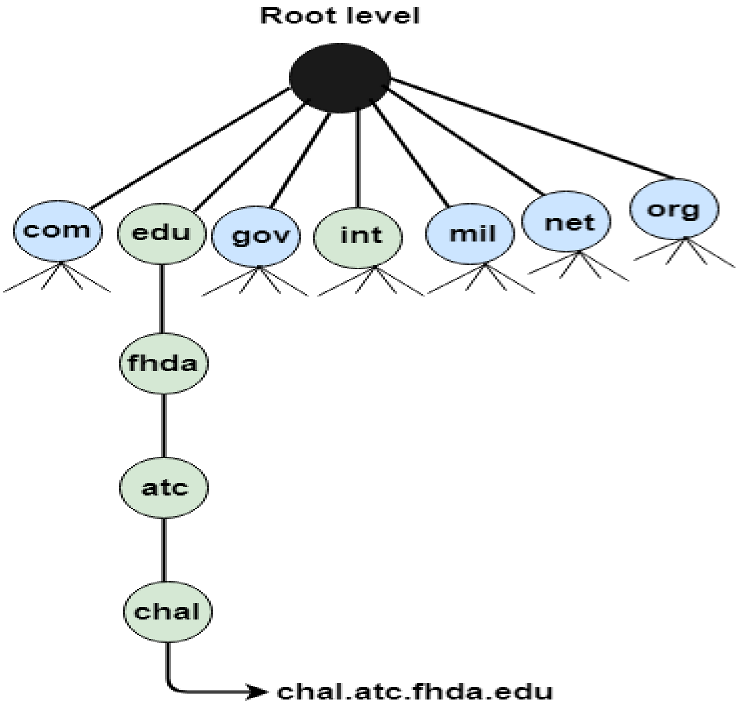
DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



## 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

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

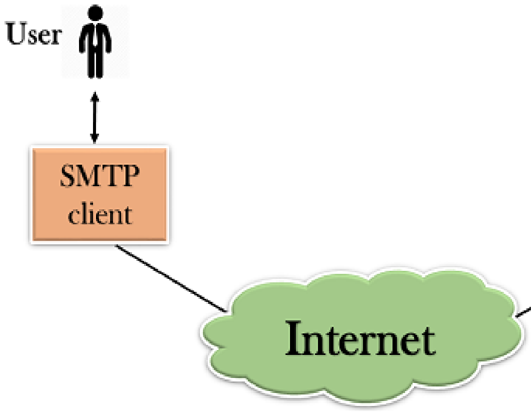
- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted 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

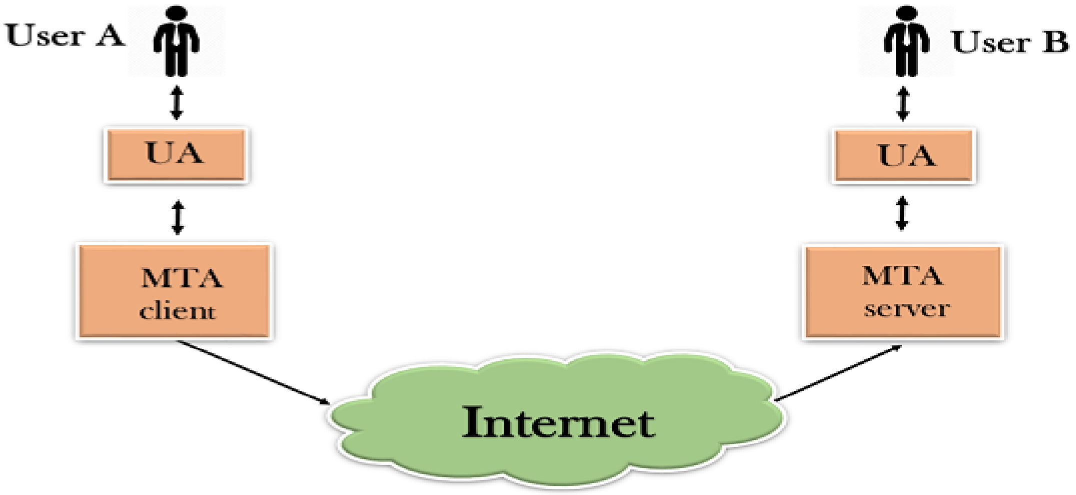
- SMTP stands for Simple Mail Transfer Protocol.
- SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called Simple Mail Transfer Protocol.
- It is a program used for sending messages to other computer users based on e-mail addresses.
- It 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.

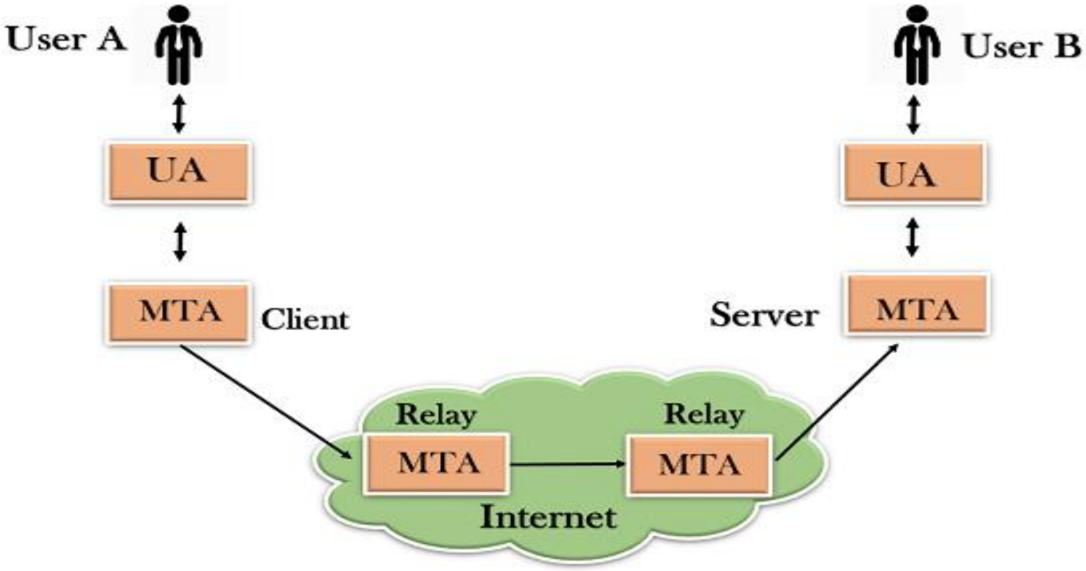
### Components of SMTP



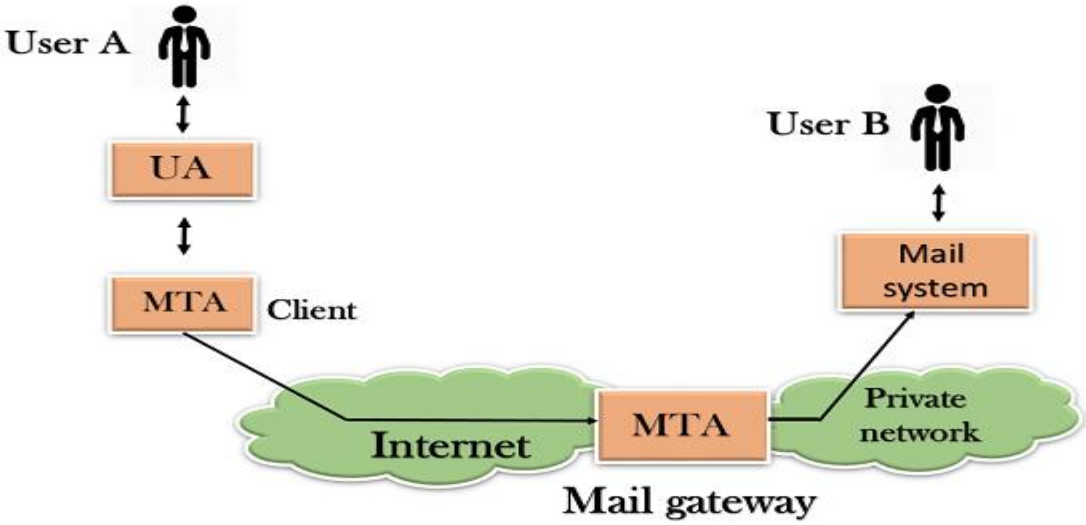
First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.



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**

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.

**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.

**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.

**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**.**

**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.

**MIME -**MIME stands for Multipurpose Internet Mail Extensions.

- It is used to extend the capabilities of Internet e-mail protocols such as SMTP. The MIME protocol allows the users to exchange various types of digital content such as pictures, audio, video, and various types of documents and files in the e-mail.
- MIME was created in 1991 by a computer scientist named Nathan Borenstein at a company called Bell Communications.

**MIME is an e-mail extension protocol**, i.e., it does not operate independently, but it helps to extend the capabilities of e-mail in collaboration with other protocols such as SMTP.

Since MIME was able to transfer only text written file in a limited size English language with the help of the internet. At present, it is used by almost all e-mail related service companies such as Gmail, Yahoo-mail, Hotmail.

**Need of MIME Protocol**

- MIME protocol is used to transfer e-mail in the computer network for the following reasons:
- The MIME protocol supports multiple languages in e-mail, such as Hindi, French, Japanese, Chinese, etc.
- Simple protocols can reject mail that exceeds a certain size, but there is no word limit in MIME.
- Images, audio, and video cannot be sent using simple e-mail protocols such as SMTP. These require MIME protocol.
- Many times, emails are designed using code such as HTML and CSS, they are mainly used by companies for marketing their product. This type of code uses MIME to send email created from HTML and CSS.

**MIME Header**

MIME adds five additional fields to the header portion of the actual e-mail to extend the properties of the simple email protocol. These fields are as follows:

**MIME Version**

- Content Type
- Content Type Encoding
- Content Id
- Content description

**1. MIME Version**

It defines the version of the MIME protocol. This header usually has a parameter value 1.0, indicating that the message is formatted using MIME.

**2. Content Type**

It describes the type and subtype of information to be sent in the message. These messages can be of many types such as Text, Image, Audio, Video, and they also have many subtypes such that the subtype of the image can be png or jpeg. Similarly, the subtype of Video can be WEBM, MP4 etc.

**3. Content Type Encoding**

In this field, it is told which method has been used to convert mail information into ASCII or Binary number, such as 7-bit encoding, 8-bit encoding, etc.

**4. Content Id**

In this field, a unique "Content Id" number is appended to all email messages so that they can be uniquely identified.

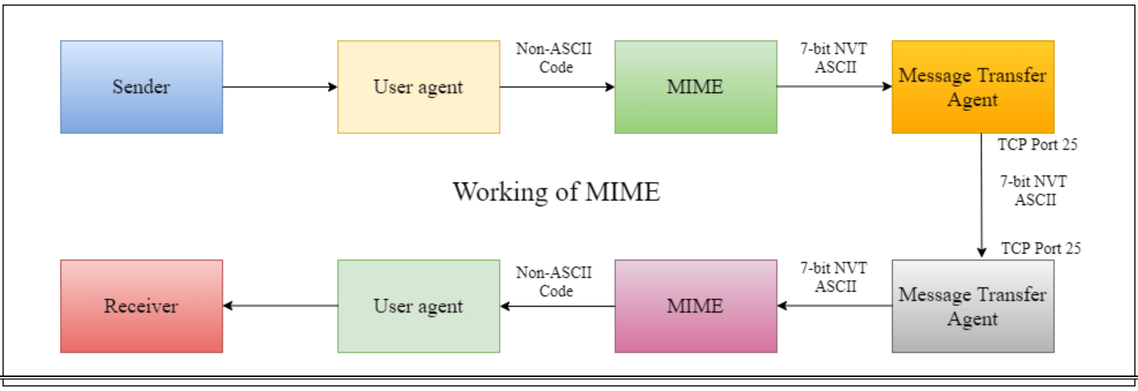
**5. Content description**

This field contains a brief description of the content within the email. This means that information about whatever is being sent in the mail is clearly in the "Content Description". This field also provides the information of name, creation date, and modification date of the file.

**Example of Content description**

Content-Description:attachment;filename=javatpoint.jpeg;
modification-date = "Wed, 12 Feb 1997 16:29:51 -0500";

Working diagram of MIME Protocol



**Features of MIME Protocol**

- It supports multiple attachments in a single e-mail.
- It supports the non-ASCII characters.
- It supports unlimited e-mail length.
- It supports multiple languages.
- Advantage of the MIME

**The MIME protocol has the following advantages:**

- It is capable of sending various types of files in a message, such as text, audio, video files.
- It also provides the facility to send and receive emails in different languages like Hindi, French, Japanese, Chinese etc.
- It also provides the facility of connecting HTML and CSS to email, due to which people can design email as per their requirement and make it attractive and beautiful.
- It is capable of sending the information contained in an email regardless of its length.
- It assigns a unique id to all e-mails.