18BSC52C – COMPUTER NETWORKS
UNIT – V

Congestion Control Algorithms – Approaches – Traffic-aware Routing – Admission Control – Traffic Throttling – Load shedding - The Transport Layer – Services provided to the upper layers - Transport service primitives – Elements of Transport protocols – addressing – Connection Establishment – Connection Release – Error Control and Flow control – Multiplexing - Crash recovery - Application Layer – DNS – The Domain Name System – Electronic mail – Architecture and services - the user agent – Message formats – Message Transfer – Final Delivery.

Congestion Control

- What is **congestion**?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Effects of Congestion

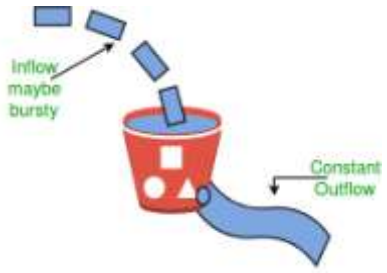
- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Congestion control algorithms

- **Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



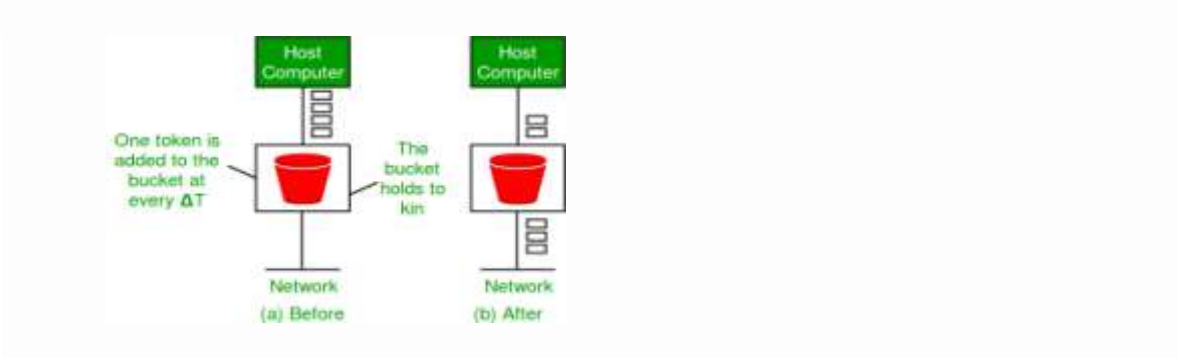
Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

- **Token bucket Algorithm**
  - Need of token bucket Algorithm:-
  - The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is.
  - So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost.
  - One such algorithm is token bucket algorithm.

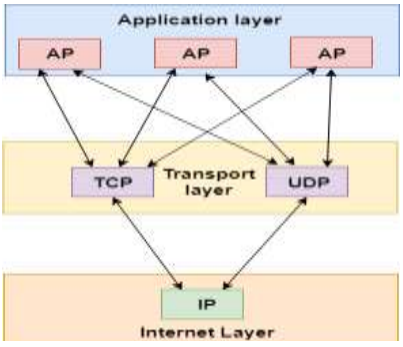
Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. *f*
2. The bucket has a maximum capacity. *f*
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.



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.

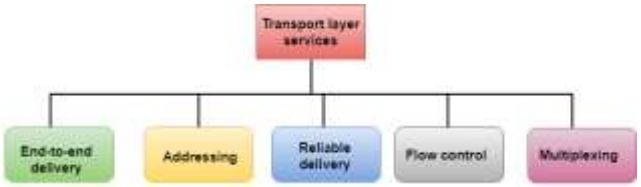


Services provided by the Transport Layer

- 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.
- 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.
- 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.
- The transport layer performs the checking for the errors end-to-end to ensure that the packet has arrived correctly.

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.

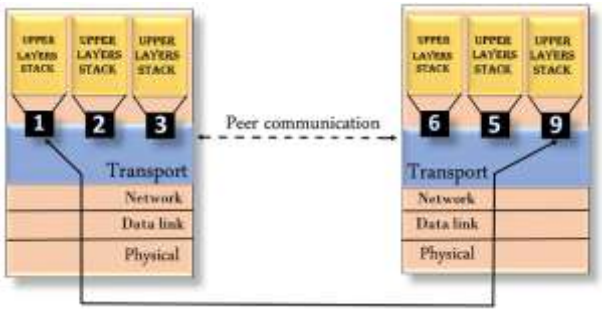
Multiplexing

- 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.
  - 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

- Data generated by an application on one machine must be transmitted to the correct application on another machine.
- 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.



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.

Domain Name System

- When **DNS** was not into existence, one had to download a **Host file** containing host names and their corresponding IP address.
- But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file.
- To solve this problem the DNS system was introduced.
- **Domain Name System** helps to resolve the host name to an address.
- It uses a hierarchical naming scheme and distributed database of IP addresses and associated names

IP Address

- IP address is a unique logical address assigned to a machine over the network.
- An IP address exhibits the following properties:
    - IP address is the unique address assigned to each host present on Internet.
    - IP address is 32 bits (4 bytes) long.
    - IP address consists of two components: **network component** and **host component**.
    - Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

Uniform Resource Locator (URL)

- **Uniform Resource Locator (URL)** refers to a web address which uniquely identifies a document over the internet.
- This document can be a web page, image, audio, video or anything else present on the web.

URL Types

There are two forms of URL as listed below:

- Absolute URL
- Relative URL

Absolute URL

- Absolute URL is a complete address of a resource on the web.
- This completed address comprises of protocol used, server name, path name and file name.
- For example http:// www.tutorialspoint.com / internet_technology /index.htm. where:
    - **http** is the protocol.
    - **tutorialspoint.com** is the server name.
    - **index.htm** is the file name.

Relative URL

- Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL.
- Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the links.

Difference between Absolute and Relative URL

Absolute URL	**Relative URL**
Used to link web pages on different websites	Used to link web pages within the same website.
Difficult to manage.	Easy to Manage
Changes when the server name or directory name changes	Remains same even of we change the server name or directory name.
Take time to access	Comparatively faster to access.

Domain Name System Architecture

- The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

Domain Names

- Domain Name is a symbolic string associated with an IP address.
- There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while some country level domain names such as **au, in, za, us** etc.

The following table shows the **Generic** Top-Level Domain names:

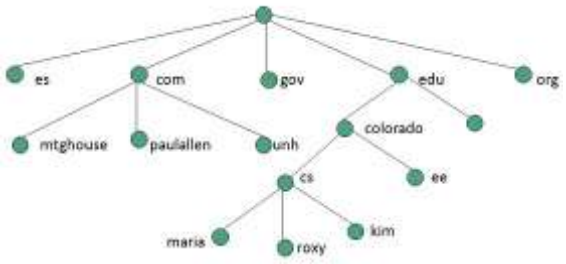
Domain Name	**Meaning**
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

The following table shows the **Country top-level** domain names:

Domain Name	**Meaning**
Au	Australia
In	India
Cl	Chile
Fr	France
Us	United States
Za	South Africa
Uk	United Kingdom
Jp	Japan
Es	Spain
De	Germany
Ca	Canada
Ee	Estonia
Hk	Hong Kong

Domain Name Space

- The domain name space refers a hierarchy in the internet naming structure.
- This hierarchy has multiple levels (from 0 to 127), with a root at the top.
- The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

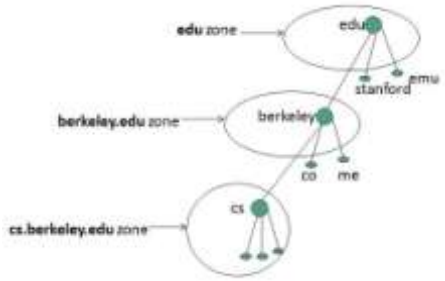
Name Server

- Name server contains the DNS database.

- This database comprises of various names and their corresponding IP addresses.
- Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.
- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

- Zone is collection of nodes (sub domains) under the main domain.
- The server maintains a database called zone file for every zone.



- If the domain is not further divided into sub domains then domain and zone refers to the same thing.
- The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

Types of Name Servers

Following are the three categories of Name Servers that manages the entire Domain Name System:

- Root Server
- Primary Server
- Secondary Server

Root Server

- Root Server is the top level server which consists of the entire DNS tree.
- It does not contain the information about domains but delegates the authority to the other server

Primary Servers

- Primary Server stores a file about its zone.
- It has authority to create, maintain, and update the zone file.

Secondary Server

- Secondary Server transfers complete information about a zone from another server which may be primary or secondary server.
- The secondary server does not have authority to create or update a zone file.

DNS Working

- DNS translates the domain name into IP address automatically.
- Following steps will take you through the steps included in domain resolution process:
- When we type **www.tutorialspoint.com** into the browser, it asks the local DNS Server for its IP address.
  Here the local DNS is at ISP end.
- When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.

- The root DNS server replies with delegation that **I do not know the IP address of www.tutorialspoint.com but know the IP address of DNS Server.**
- The local DNS server then asks the com DNS Server the same question.
- The **com** DNS Server replies the same that it does not know the IP address of www.tutorialspont.com but knows the address of tutorialspoint.com.
- Then the local DNS asks the tutorialspoint.com DNS server the same question.
- Then tutorialspoint.com DNS server replies with IP address of www.tutorialspoint.com.
- Now, the local DNS sends the IP address of www.tutorialspoint.com to the computer that sends the request.

Electronic - Mail

- Electronic mail or e-mail, as it is known by its fans became known to the public at large and its use grew exponentially.
- The first e-mail systems consisted of file transfer protocols, with the convention that the first line of the message contained the recipient's address.
- It is a store and forward method of composing, sending, storing, and receiving messages over electronic communication systems.
- The term "e-mail" applies both to the Internet e-mail system based on the Simple Mail Transfer Protocol (SMTP) and to intranet systems allowing users within one organization to e-mail each other.
- Often workgroup collaboration organizations may use the Internet protocols for internal e-mail service. E-mail is often used to deliver bulk unwanted messages, or "spam", but filter programs exist which can automatically delete most of these. E-mail systems based on RFC 822 are widely used.

Architecture :

- E-mail system normally consists of two sub systems
    - 1. User agents
    - 2. Message transfer agents

User agents

- The user agents allow people to read and send e-mails.
- The message transfer agents move the messages from source to destination.
- The user agents are local programs that provide a command based, menu-based, or graphical method for interacting with e-mail system.
- The message transfer agents are daemons, which are processes that run in background.
- Their job is to move datagram e-mail through system.
- A key idea in e-mail system is the distinction between the envelope and its contents.
- The envelope encapsulates the message.
- It contains all the information needed for transporting the message like destinations address, priority, and security level, all of which are distinct from the message itself.

The Message transfer agents

- The message transport agents use the envelope for routing. The message inside the envelope consists of two major sections:
    - The Header:
        - The header contains control information for the user agents. It is structured into fields such as summary, sender, receiver, and other information about the e-mail.
    - · Body:
        - The body is entirely for human recipient. The message itself as unstructured text; sometimes containing a signature block at the end

- Header format
  - The header is separated from the body by a blank line.
  - It consists of following fields
    - From: The e-mail address, and optionally name, of the sender of the message.
    - To: one or more e-mail addresses, and optionally name, of the receiver's of the message.
    - Subject: A brief summary of the contents of the message.
    - Date: The local time and date when the message was originally sent.

User agents:

- It is normally a program and sometimes called a mail reader.
- It accepts a variety of commands for composing, receiving, replying messages as well as manipulating the mail boxes.
- Some user agents have a fancy menu or icon driven interfaced that require a mouse where as others are one character commands from keyboard.
- Functionally these are same.
- Some systems are menu or icon driven but also have keyboard shortcuts.
- To send an e-mail, user provides the message, the destination address and possibly some other parameters.
- Most e-mail system supports mailing lists.
- Example: Reading e-mail
  - When a user is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen. Then it announces the number of messages in the mailbox or displays a one-line summary of each e-mail and wait for a command.
- Each line of the display contains several fields extracted from the envelope or header of the corresponding message.
- In a simple e-mail system, the choice of fields is built into the program.
- In more sophisticated system, user can specify which fields are to be displayed by providing a user profile.
- Referring to the display it contains following fields
  - 1. Message number: it is serial number of the message. It can be displayed from the most currently received messages or vice versa.
  - 2. Flags: contains K means the message is not new, A means the message is already read and F means the message has been forwarded to someone else.
  - 3. size of the message: indicates the length of the message
  - 4. source of the message: originator's address
  - 5. subject: gives a brief summary of what the message is about.

E-mail Services

Basic services:

- E-mail systems support five basic functions. These basic functions are:
  - 1. Composition:
    - It refers to the process of creating messages and answers. Any text editor can be used for the body of the message, the system itself can provide assistance with addressing and the numerous header fields attached to each message.
  - 2. Transfer:
    - It refers to moving messages from the originator to the recipient. This requires establishing a connection to the destination or some intermediate

machine, outputting the message, and finally releasing the connection. E-mail does it automatically without bothering the user.

- 3. Reporting:
  - It refers to acknowledging or telling the originator what happened to the message. Was the message delivered? Was it rejected? Numerous applications exist in which confirmation of delivery is important and may even have a legal significance. E-mail system is not very reliable.
- 4. Displaying
  - The incoming message has to be displayed so that people can read their e-mail. Sometimes conversation is required or a special viewer must be invoked. For example: if message is a postscript file or digitized voice. Simple conversations and formatting are sometimes attempted.
- 5. Disposition
  - It is the final step and concerns what the recipient does with the message after receiving it. Possibilities include throwing it away before reading, throwing it away after reading, saving it, and so on. It should be possible to retrieve and reread saved messages, forward them or process them in other ways.

Message Formats

- Messages sent by the user agent must be placed in a standard format to be handled by the message transfer agents.

RFC 5322—The Internet Message Format

- Messages consist of a primitive envelope (described as part of SMTP in RFC 5321), some number of header fields, a blank line, and then the message body.
- Each header field (logically) consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value.
- The original RFC 822 was designed decades ago and did not clearly distinguish the envelope fields from the header fields.
- The user agent builds a message and passes it to the message transfer agent, which then uses some of the header fields to construct the actual envelope, a somewhat old fashioned mixing of message and envelope.
- The principal header fields related to message transport.
  - The *To:* field gives the DNS address of the primary recipient. Having multiple recipients is also allowed.
  - The *Cc:* field gives the addresses of any secondary recipients.
  - The term *Cc:* (Carbon copy) is a bit dated, since computers do not use carbon paper, but it is well established.
  - The *Bcc:* (Blind carbon copy) field is like the *Cc:* field, except that this line is deleted from all the copies sent to the primary and secondary recipients. This feature allows people to send copies to third parties without the primary and secondary recipients knowing this.
  - *From:* and *Sender:*, tell who wrote and sent the message, respectively. These need not be the same.
  - The *Reply-To:* field is sometimes used when neither the person composing the message nor the person sending the message wants to see the reply.
  - The *Message-Id:* is an automatically generated number that is used to link messages together (e.g., when used in the *In-Reply-To:* field) and to prevent duplicate delivery.

Multipurpose Internet Mail Extension (MIME) Protocol

- **Multipurpose Internet Mail Extension (MIME)** is a standard which was proposed by Bell Communications in 1991 in order to expand limited capabilities of email.
- MIME is a kind of *add on or a supplementary protocol* which allows non-ASCII data to be sent through SMTP.
- It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.

Why do we need MIME?

Limitations of Simple Mail Transfer Protocol (SMTP):

1. SMTP has a very simple structure
2. It's simplicity however comes with a price as it only send messages in NVT 7-bit ASCII format.
3. It cannot be used for languages that do not support 7-bit ASCII format such as- French, German, Russian, Chinese and Japanese, etc. so it cannot be transmitted using SMTP. So, in order *to make SMTP more broad we use MIME*.
4. It cannot be used to send binary files or video or audio data.

Features of MIME –

1. It is able to send multiple attachments with a single message.
2. Unlimited message length.
3. Binary attachments (executables, images, audio, or video files) which may be divided if needed.
4. MIME provided support for varying content types and multi-part messages.

MIME with SMTP and POP –

- SMTP transfers the mail being a message transfer agent from senders side to the mailbox of receiver side and stores it and MIME header is added to the original header and provides additional information. while POP being the message access agent organizes the mails from the mail server to the receivers computer. POP allows user agent to connect with the message transfer agent.

MIME Header:

It is added to the original e-mail header section to define transformation. There are *five headers* which we add to the original header:

1. **MIME Version –** Defines version of MIME protocol. It must have the parameter *Value 1.0*, which indicates that message is formatted using MIME.
2. **Content Type –** Type of data used in the body of message. They are of different types like text data (plain, HTML), audio content or video content.
3. **Content Type Encoding –** It defines the method used for encoding the message. Like 7-bit encoding, 8-bit encoding, etc.
4. **Content Id –** It is used for uniquely identifying the message.
5. **Content description –** It defines whether the body is actually image, video or audio.

<u>Message Transfer</u>

- The mail transfer is done with the SMTP protocol.
- The simplest way to move messages is to establish a transport connection

from the source machine to the destination machine and then just transfer the message.

SMTP (Simple Mail Transfer Protocol) and Extensions

- Within the Internet, email is delivered by having the sending computer establish

a TCP connection to port 25 of the receiving computer.

- Listening to this port is a mail server that speaks **SMTP** (**Simple Mail Transfer Protocol**).
- This server accepts incoming connections, subject to some security checks, and accepts messages for delivery.
- If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.
- SMTP is a simple ASCII protocol.
- This is not a weakness but a feature.
- Most application-level Internet protocols now work this way (e.g., HTTP).

  STARTTLS Switch to secure transport
  UTF8SMTP Internationalized addresses

SMTP Protocol

The SMTP model is of two type :

- End-to- end method
- Store-and- forward method

- The end to end model is used to communicate between different organizations whereas the store and forward method are used within an organization.
- A SMTP client who wants to send the mail will contact the destination's host SMTP directly in order to send the mail to the destination.
- The SMTP server will keep the mail to itself until it is successfully copied to the receiver's SMTP.
- The client SMTP is the one which initiates the session let us call it as the client- SMTP and the server SMTP is the one which responds to the session request and let us call it as receiver-SMTP.
- The client- SMTP will start the session and the receiver-SMTP will respond to the request.

Both the SMTP-client and SMTP-server should have 2 components:

- User agent (UA)
- Local MTA

Communication between sender and the receiver

- The senders, user agent prepare the message and send it to the MTA.
- The MTA functioning is to transfer the mail across the network to the receivers MTA.
- To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.

Sending email:

- Mail is sent by a series of request and response messages between the client and a server.
- The message which is sent across consists of a header and the body.
- A null line is used to terminate the mail header.
- Everything which is after the null line is considered as the body of the message which is a sequence of ASCII characters.
- The message body contains the actual information read by the receipt.

Receiving email:

- The user agent at the server-side checks the mailboxes at a particular time of intervals.
- If any information is received it informs the user about the mail. \When the user tries to read the mail it displays a list of mails with a short description of each mail in the mailbox.

- By selecting any of the mail user can view its contents on the terminal.

Some SMTP Commands:

- HELO – Identifies the client to the server, fully qualified domain name, only sent once per session
- MAIL – Initiate a message transfer, fully qualified domain of originator
- RCPT – Follows MAIL, identifies an addressee, typically the fully qualified name of the addressee and for multiple addressees use one RCPT for each addressee
- DATA – send data line by line
- AUTH - Client authentication
- BINARYMIME - Server accepts binary messages
- CHUNKING - Server accepts large messages in chunks
- SIZE - Check message size before trying to send

Final Delivery

- Our mail message is almost delivered.
- This task was straightforward in the early Internet, when the user agent and mail transfer agent ran on the same machine as different processes.
- The mail transfer agent simply wrote new messages to the end of the mailbox file, and the user agent simply checked the mailbox file for new mail.
- The job of the user agent is to present a view of the contents of the mailbox, and to allow the mailbox to be remotely manipulated.
- Several different protocols can be used for this purpose, but SMTP is not one of them.
- SMTP is a push-based protocol. It takes a message and connects to a remote server to transfer the message.
- Final delivery cannot be achieved in this manner both because the mailbox must continue to be stored on the mail transfer agent and because the user agent may not be connected to the Internet at the moment that SMTP attempts to relay messages.

REFERENCES

1. Andrew S. Tanenbaum, David J. Wetherall, " Computer Networks", 5th Edition, Pearson Education Publ. – 2011.
2. https://www.tutorialspoint.com/internet_technologies/internet_domain_name_system.htm

----------xxxxxxxxxxxx----------