**Unit-III**

**Transport Layer and Application Layer 05 Hours**

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

**Services:** Berkeley Sockets, Connection Establishment, Connection Release

**Application Layer:** Domain Name System (DNS) and File Transfer Protocol (FTP)

**WWW:** Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0) and HTTPS with SSL.

**Email:** SMTP, MIME, POP3 and Webmail.

**1. Transport Layer**

Transport Layer is the second layer of the TCP/IP model. It is an **end-to-end** layer used to deliver messages to a host. It is termed as an end-to-end layer because it provides a point-to-point connection **rather than** hop-to- hop, between the source host and destination host to deliver the services reliably. The unit of data encapsulation in Transport Layer is a segment.

The standard protocols used by Transport Layer to enhance its functionalities are TCP(Transmission Control Protocol), UDP( User Datagram Protocol), DCCP( Datagram Congestion Control Protocol) etc.

Various responsibilities of a Transport Layer –

* **Process to process delivery –**

While Data Link Layer requires the MAC address (48 bits address contained inside the Network Interface Card of every host machine) of source-destination hosts to correctly deliver a frame and Network layer requires the IP address for appropriate routing of packets , in a similar way Transport Layer requires a Port number to correctly deliver the segments of data to the correct process amongst the multiple processes running on a particular host. A **port number** is a 16 bit address used to identify any client-server program uniquely.

* **End-to-end Connection between hosts –**

The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP. TCP is a secure, connection- orientated protocol which uses a handshake protocol to establish a robust connection between two end- hosts. TCP ensures reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol which ensures best-effort delivery. It is suitable for the applications which have little concern with flow or error control and requires to send the bulk of data like video conferencing. It is often used in multicasting protocols.

* **Multiplexing and Demultiplexing –**

Multiplexing allows simultaneous use of different applications over a network which is running on a host. The transport layer provides this mechanism which enables us to send packet streams from various applications simultaneously over a network. The transport layer accepts these packets from different processes differentiated by their port numbers and passes them to the network layer after adding proper headers. Similarly, Demultiplexing is required at the receiver side to obtain the data coming from various processes. Transport receives the segments of data from the network layer and delivers it to the appropriate process running on the receiver’s machine.

* **Congestion Control –**

Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occur. As a result retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides Congestion Control in different ways. It uses **open loop** congestion control to prevent the congestion and **closed loop** congestion control to remove the congestion in a network once it occurred. TCP provides AIMD- additive increase multiplicative decrease, leaky bucket technique for congestion control.

* **Data integrity and Error correction –**

Transport layer checks for errors in the messages coming from application layer by using error detection codes, computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.

* **Flow control –**

The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

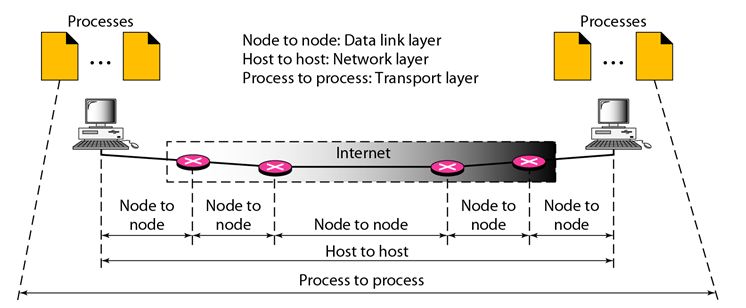
* 1. **Process-to-Process Delivery Concepts- UDP, TCP, SCTP**

The Internet model has three protocols at the transport layer: UDP, TCP, and SCTP.

The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called node-to-node delivery. The network layer is responsible for delivery of datagrams between two hosts. This is called host-to-host delivery. Communication on the Internet is not defined as the exchange of data between two nodes or between two hosts. Real communication takes place between two processes. So that we need process-to-process delivery.

However, at any moment, several processes may be running on the source host and several on the destination host. To complete the delivery, we need a mechanism to deliver data from one of these processes running on the source host to the corresponding process running on the destination host.

The transport layer is responsible for process-to-process delivery-the delivery of a packet, part of a message, from one process to another. The following figure shows these three types of deliveries and their domains.



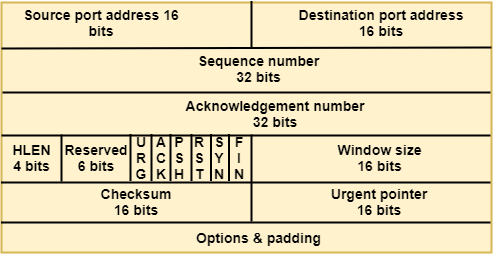
**2. TCP**

* 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.

**2.1 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.

**2.2 TCP Segment Format**



**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.

**2.3 TCP connection**

TCP is a connection oriented protocol and every connection oriented protocol needs to establish connection in order to reserve resources at both the communicating ends.

**2.3.1 Connection Establishment –**

1. Sender starts the process with following:
   * **Sequence number (Seq=521):** contains the random initial sequence number which generated at sender side.
   * **Syn flag (Syn=1):** request receiver to synchronize its sequence number with the above provided sequence number.
   * **Maximum segment size (MSS=1460 B):** sender tells its maximum segment size, so that receiver sends datagram which won’t require any fragmentation. MSS field is present inside **Option** field in TCP header.
   * **Window size (window=14600 B):** sender tells about his buffer capacity in which he has to store messages from receiver.

1. TCP is a full duplex protocol so both sender and receiver require a window for receiving messages from one another.
   * **Sequence number (Seq=2000):** contains the random initial sequence number which generated at receiver side.
   * **Syn flag (Syn=1):** request sender to synchronize its sequence number with the above provided sequence number.
   * **Maximum segment size (MSS=500 B):** sender tells its maximum segment size, so that receiver sends datagram which won’t require any fragmentation. MSS field is present inside **Option** field in TCP header.  
     Since MSSreceiver < MSSsender, both parties agree for minimum MSS i.e., 500 B to avoid fragmentation of packets at both ends.
   * Therefore, receiver can send maximum of 14600/500 = 29 packets.

This is the receiver's sending window size.

* + **Window size (window=10000 B):** receiver tells about his buffer capacity in which he has to store messages from sender.
  + Therefore, sender can send a maximum of 10000/500 = 20 packets.

This is the sender's sending window size.

* + **Acknoledgement Number (Ack no.=522):** Since sequence number 521 is received by receiver so, it makes a request of next sequence number with Ack no.=522 which is the next packet expected by receiver since Syn flag consumes 1 sequence no.
  + **ACK flag (ACk=1):** tells that acknowledgement number field contains the next sequence expected by receiver.

1. Sender makes the final reply for connection establishment in following way:
   * **Sequence number (Seq=522):** since sequence number = 521 in 1st step and SYN flag consumes one sequence number hence, next sequence number will be 522.
   * **Acknowledgement Number (Ack no.=2001):** since sender is acknowledging SYN=1 packet from the receiver with sequence number 2000 so, the next sequence number expected is 2001.
   * **ACK flag (ACK=1):** tells that acknowledgement number field contains the next sequence expected by sender.

**https://tutorialspoint.dev/image/net.png**

**2.4 TCP connection release**

In TCP 3-way Handshake Process we studied that how connection establish between client and server in Transmission Control Protocol (TCP) using **SYN** bit segments. In this article we will study about how TCP close connection between Client and Server. Here we will also need to send bit segments to server which **FIN** bit is set to 1.

TCP supports two types of connection releases like most connection-oriented transport protocols:

1. **Graceful connection release –**In Graceful connection release, the connection is open until both parties have closed their sides of the connection.
2. **Abrupt connection release -**In Abrupt connection release, either one TCP entity is forced to close the connection or one user closes both directions of data transfer.

**Abrupt connection release :**An abrupt connection release is carried out when a RST segment is sent. A RST segment can be sent for the below reasons:

1. When a non-SYN segment was received for a non-existing TCP connection.
2. In an open connection, some TCP implementations send a RST segment when a segment with an invalid header is received. This will prevent attacks by closing the corresponding connection.
3. When some implementations need to close an existing TCP connection, they send a RST segment. They will close an existing TCP connection for the following reasons:
   * Lack of resources to support the connection
   * The remote host is now unreachable and has stopped responding.

When a TCP entity sends a RST segment, it should contain 00 if it does not belong to any existing connection else it should contain the current value of the sequence number for the connection and the acknowledgment number should be set to the next expected in-sequence sequence number on this connection.

**Graceful Connection Release :** The common way of terminating a TCP connection is by using the TCP header’s FIN flag. This mechanism allows each host to release its own side of the connection individually.

11

How mechanism works In TCP :

1. **Step 1 (FIN From Client) –** Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client send a TCP segment with the **FIN** bit set to **1** to server and to enter the **FIN\_WAIT\_1** state. While in the **FIN\_WAIT\_1** state, the client waits for a TCP segment from the server with an acknowledgment (ACK).
2. **Step 2 (ACK From Server) –** When Server received FIN bit segment from Sender (Client), Server Immediately send acknowledgement (ACK) segment to the Sender (Client).
3. **Step 3 (Client waiting) –** While in the **FIN\_WAIT\_1** state, the client waits for a TCP segment from the server with an acknowledgment. When it receives this segment, the client enters the **FIN\_WAIT\_2** state. While in the **FIN\_WAIT\_2** state, the client waits for another segment from the server with the FIN bit set to 1.
4. **Step 4 (FIN from Server) –** Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment (because of some closing process in the Server).
5. **Step 5 (ACK from Client) –** When Client receive FIN bit segment from the Server, the client acknowledges the server’s segment and enters the **TIME\_WAIT** state. The **TIME\_WAIT** state lets the client resend the final acknowledgment in case the **ACK** is lost.The time spent by client in the **TIME\_WAIT** state is depend on their implementation, but their typical values are 30 seconds, 1 minute, and 2 minutes. After the wait, the connection formally closes and all resources on the client side (including port numbers and buffer data) are released.

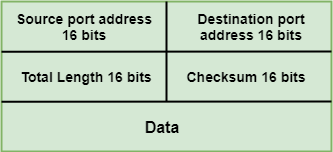
**3. UDP**

**User Datagram Protocol (UDP)**is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. Unlike TCP, it is unreliable and connectionless protocol. So, there is no need to establish connection prior to data transfer.  Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of Internet services; provides assured delivery, reliability and much more but all these services cost us with additional overhead and latency. Here, UDP comes into picture. For the realtime services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also save bandwidth.  User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

* 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:



**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.

**3.1 Comparision**

| **Transmission control protocol (TCP)** | **User datagram protocol (UDP)** |
| --- | --- |
| TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data. | UDP is the Datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission. |
| TCP is reliable as it guarantees the delivery of data to the destination router. | The delivery of data to the destination cannot be guaranteed in UDP. |
| TCP provides extensive error checking mechanisms. It is because it provides flow control and acknowledgement of data. | UDP has only the basic error checking mechanism using checksums. |
| Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in-order at the receiver. | There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer. |
| TCP is comparatively slower than UDP. | UDP is faster, simpler, and more efficient than TCP. |
| Retransmission of lost packets is possible in TCP, but not in UDP. | There is no retransmission of lost packets in the User Datagram Protocol (UDP). |
| TCP has a (20-60) bytes variable length header. | UDP has an 8 bytes fixed-length header. |
| TCP is heavy-weight. | UDP is lightweight. |
| TCP doesn’t support Broadcasting. | UDP supports Broadcasting. |
| TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet. | UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP. |

**4. Berkeley Sockets**

Berkeley Sockets (also called **BSD sockets**) are the foundational **API for network communication** used in UNIX-like operating systems. They provide a programming interface to use transport-layer protocols such as **TCP and UDP**.

**What is a Socket?**

A **socket** is an endpoint for communication. In the Berkeley API, a socket is treated much like a file descriptor. Processes can read() from and write() to a socket just like they would with files.

**Typical Socket Operations:**

| **Operation** | **Description** |
| --- | --- |
| socket() | Creates a socket and returns its descriptor. |
| bind() | Assigns an IP address and port to the socket. |
| listen() | Used by a server to wait for incoming connections (TCP). |
| accept() | Accepts an incoming TCP connection request. |
| connect() | Initiates a connection to a remote socket (client-side for TCP). |
| send()/recv() | Used to transmit and receive messages (TCP/UDP). |
| close() | Terminates the connection and releases resources. |

**Socket Types:**

* **SOCK\_STREAM** → for TCP (reliable, connection-oriented)
* **SOCK\_DGRAM** → for UDP (unreliable, connectionless)

**Socket Programming**

The term *network programming* refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

The java.net package of the J2SE APIs contains a collection of classes and interfaces that provide the low-level communication details, allowing you to write programs that focus on solving the problem at hand.

The java.net package provides support for the two common network protocols:

* **TCP:** TCP stands for Transmission Control Protocol, which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.
* **UDP:** UDP stands for User Datagram Protocol, a connection-less protocol that allows for packets of data to be transmitted between applications.

**4.1 Socket Programming:**

Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket.

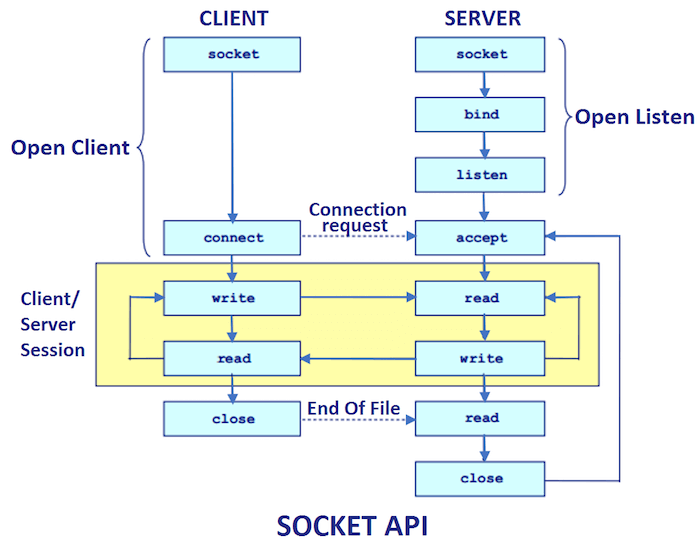
The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

***The following steps occur when establishing a TCP connection between two computers using sockets:***

* The server instantiates a ServerSocket object, denoting which port number communication is to occur on.
* The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.
* After the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.
* The constructor of the Socket class attempts to connect the client to the specified server and port number. If communication is established, the client now has a Socket object capable of communicating with the server.
* On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

After the connections are established, communication can occur using I/O streams. Each socket has both an OutputStream and an InputStream. The client's OutputStream is connected to the server's InputStream, and the client's InputStream is connected to the server's OutputStream.

TCP is a two way communication protocol, so data can be sent across both streams at the same time. There are following use full classes providing complete set of methods to implement sockets.



|  |
| --- |
| **import** java.io.BufferedReader;  **import** java.io.InputStream;  **import** java.io.InputStreamReader;  **import** java.io.OutputStream;  **import** java.io.PrintWriter;  **import** java.net.Socket;  **public** **class** Client  {  **public** **static** **void** main(String[] args) **throws** Exception  {  Socket sock = **new** Socket("127.0.0.1", 3000);  BufferedReader keyRead = **new** BufferedReader(**new** InputStreamReader(System.*in*));  BufferedReader socketRead = **new** BufferedReader(**new** InputStreamReader(sock.getInputStream()));  PrintWriter pwrite = **new** PrintWriter(sock.getOutputStream(), **true**);  String receiveMessage, sendMessage;  **while**(**true**)  {  sendMessage = keyRead.readLine(); // keyboard reading  pwrite.println(sendMessage); // sending to server  pwrite.flush();  receiveMessage =socketRead.readLine();// flush the data  **if**(receiveMessage.equals("exit")) //receive from server  {  **break**;  }  System.*out*.println(receiveMessage);  }  }  } |

|  |
| --- |
| **import** java.io.BufferedReader;  **import** java.io.InputStreamReader;  **import** java.io.PrintWriter;  **import** java.net.ServerSocket;  **import** java.net.Socket;  **public** **class** Server  {  **public** **static** **void** main(String[] args) **throws** Exception  {  ServerSocket sersock = **new** ServerSocket(3000);  Socket sock = sersock.accept( );    BufferedReader keyRead = **new** BufferedReader(**new** InputStreamReader(System.*in*));  BufferedReader socketRead = **new** BufferedReader(**new** InputStreamReader(sock.getInputStream()));    PrintWriter pwrite = **new** PrintWriter(sock.getOutputStream(), **true**);    String receiveMessage, sendMessage;  **while**(**true**)  {  receiveMessage =socketRead.readLine();// flush the data  **if**(receiveMessage.equals("exit")) //receive from server  {  **break**;  }  System.*out*.println(receiveMessage); // displaying at DOS prompt  sendMessage = keyRead.readLine();  pwrite.println(sendMessage);  pwrite.flush();  }  }  } |

**5. Domain Name System (DNS)**

DNS is a host name to IP address translation service. DNS is a distributed database implemented in a hierarchy of name servers. It is an application layer protocol for message exchange between clients and servers.

**Requirement**

Every host is identified by the IP address but remembering numbers is very difficult for the people and also the IP addresses are not static therefore a mapping is required to change the domain name to IP address. So DNS is used to convert the domain name of the websites to their numerical IP address.

**Domain :**

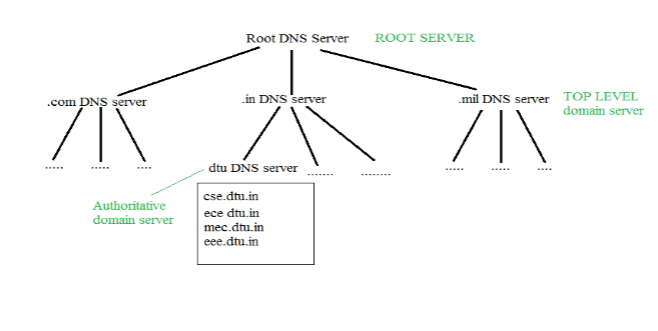
There are various kinds of DOMAIN :

1. **Generic domain :** .com(commercial) .edu(educational) .mil(military) .org(non profit organization) .net(similar to commercial) all these are generic domain.

|  |  |
| --- | --- |
| **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 |

1. **Country domain** .in (india) .us .uk. 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.
2. **Inverse domain** if we want to know what is the domain name of the website. Ip to domain name mapping.So DNS can provide both the mapping for example to find the ip addresses of geeksforgeeks.org then we have to type nslookup [www.geeksforgeeks.org](http://www.geeksforgeeks.org).

**5.1 Organization of Domain**



It is Very difficult to find out the ip address associated to a website because there are millions of websites and with all those websites we should be able to generate the ip address immediately,  
there should not be a lot of delay for that to happen organization of database is very important.  
**DNS record** – Domain name, ip address what is the validity?? what is the time to live ?? and all the information related to that domain name. These records are stored in tree like structure.

**5.2 Namespace** – Set of possible names, flat or hierarchical . Naming system maintains a collection of bindings of names to values – given a name, a resolution mechanism returns the corresponding value.

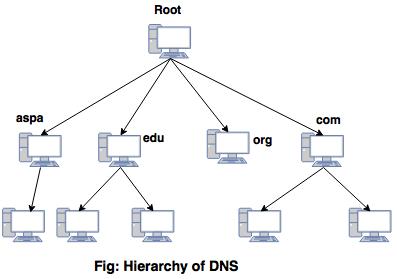
A namespace is a context within which the names of all objects must be unambiguously resolvable. For example, the internet is a single DNS name space, within which all network devices with a DNS name can be resolved to a particular address (for example, www.microsoft.com resolves to 207.46.131.13).

**Flat name spaces**

* In a flat name space, a name is a sequence of characters without structure.
* A name in this space is assigned to an address.
* The names were convenient and short.
* A flat name space cannot be used in a large system such as the internet because it must be centrally controlled to avoid ambiguity and duplication.

**Hierarchical Name Space**

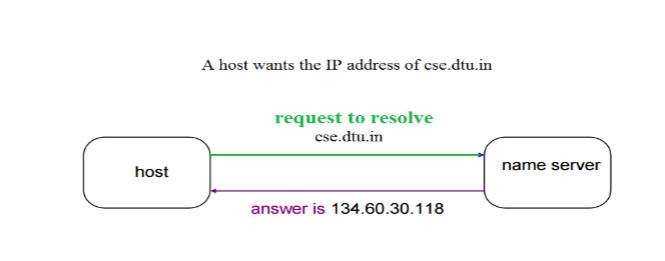
* In hierarchical name space, each name consists of several parts.
* First part defines the nature of the organization, second part defines the name of an organization, third part defines department of the organization, and so on.
* In hierarchical name space, the authority to assign and control the name spaces can be decentralized.
* Authority for names in each partition is passed to each designated agent.



**5.3 Name server** – It is an implementation of the resolution mechanism.. DNS (Domain Name System) = Name service in Internet – Zone is an administrative unit, domain is a subtree.

A name server refers to the server component of the [Domain Name System](https://en.wikipedia.org/wiki/Domain_Name_System) (DNS), one of the two principal [namespaces](https://en.wikipedia.org/wiki/Namespace) of the [Internet](https://en.wikipedia.org/wiki/Internet). The most important function of DNS servers is the translation (resolution) of human-memorable [domain names](https://en.wikipedia.org/wiki/Domain_name) (example.com) and [hostnames](https://en.wikipedia.org/wiki/Hostname) into the corresponding numeric [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP) addresses (93.184.216.34), the second principal name space of the Internet, which is used to identify and locate computer systems and resources on the Internet.

**5.4 Name to Address Resolution**

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/gq/2017/02/DNS_2.png)  
The host request the DNS name server to resolve the domain name. And the name server returns the IP address corresponding to that domain name to the host so that the host can future connect to that IP address.

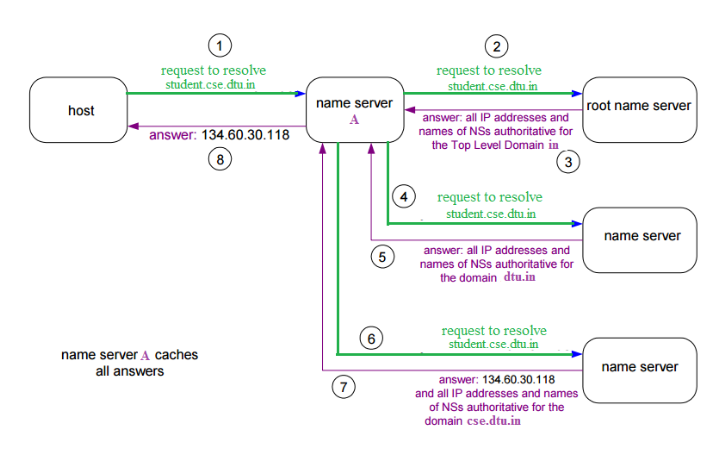
**5.5 Hierarchy of Name Servers**

**Root name servers**– It is contacted by name servers that can not resolve the name. It contacts authoritative name server if name mapping is not known. It then gets the mapping and return the IP address to the host.

**Top level server** – It is responsible for com, org, edu etc and all top level country domains like uk, fr, ca, in etc. They have info about authoritative domain servers and know names and IP addresses of each authoritative name server for the second level domains.

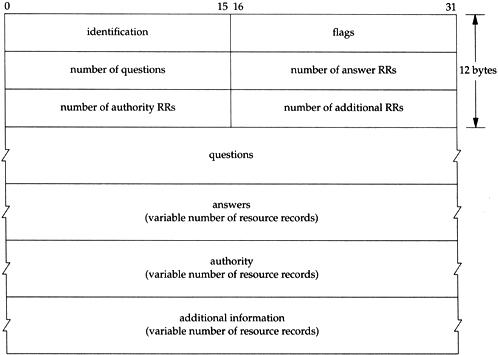
**Authoritative name servers** This is organization’s DNS server, providing authoritative hostName to IP mapping for organization servers. It can be maintained by organization or service provider. In order to reach cse.dtu.in we have to ask the root DNS server, then it will point out to the top level domain server and then to authoritative domain name server which actually contains the IP address. So the authoritative domain server will return the associative ip address.

**4.6 Domain Name Server**

The client machine sends a request to the local name server, which , if root does not find the address in its database, sends a request to the root name server , which in turn, will route the query to an intermediate or authoritative name server. The root name server can also contain some hostName to IP address mappings . The intermediate name server always knows who the authoritative name server is. So finally the IP address is returned to the local name server which in turn returns the IP address to the host.

**4.7 DNS Message Format**

There is one DNS message defined for both queries and responses. Figure 14.3 shows the overall format of the message.



The message has a fixed 12-byte header followed by four variable-length fields.

The *identification*is set by the client and returned by the server. It lets the client match responses to requests .

The 16-bit *flags*field is divided into numerous pieces, as shown in Figure

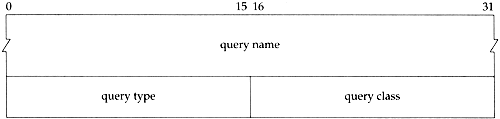
graphics/14fig04.gif

* *QR*is a 1-bit field: 0 means the message is a query, 1 means it's a response.
* *opcode*is a 4-bit field. The normal value is 0 (a standard query). Other values are 1 (an inverse query) and 2 (server status request).
* *AA*is a 1-bit flag that means "authoritative answer." The name server is authoritative for the domain in the question section.
* *TC*is a 1-bit field that means "truncated." With UDP this means the total size of the reply exceeded 512 bytes, and only the first 512 bytes of the reply was returned.
* *RD*is a 1-bit field that means "recursion desired." This bit can be set in a query and is then returned in the response. This flag tells the name server to handle the query itself, called a *recursive query.*If the bit is not set, and the requested name server doesn't have an authoritative answer, the requested name server returns a list of other name servers to contact for the answer. This is called an *iterative query.*We'll see examples of both types of queries in later examples.
* *RA*is a 1-bit field that means "recursion available." This bit is set to 1 in the response if the server supports recursion. We'll see in our examples that most name servers provide recursion, except for some root servers.
* There is a 3-bit field that must be 0.
* *rcode*is a 4-bit field with the return code. The common values are 0 (no error) and 3 (name error). A name error is returned only from an authoritative name server and means the domain name specified in the query does not exist.

The next four 16-bit fields specify the number of entries in the four variable-length fields that complete the record. For a query, the *number of questions*is normally 1 and the other three counts are 0. Similarly, for a reply the *number of answers*is at least 1, and the remaining two counts can be 0 or nonzero.

**Question Portion of DNS Query Message**

The format of each question in the *question*section is shown in Figure There is normally just one question.



The *query name*is the name being looked up. It is a sequence of one or more labels. Each *label*begins with a 1-byte count that specifies the number of bytes that follow. The name is terminated with a byte of 0, which is a label with a length of 0, which is the label of the root. Each count byte must be in the range of 0 to 63, since labels are limited to 63 bytes. (We'll see later in this section that a count byte with the two high-order bits turned on, values 192 to 255, is used with a compression scheme.) Unlike many other message formats that we've encountered , this field is allowed to end on a boundary other than a 32-bit boundary. No padding is used.

**6. File Transfer Protocol (FTP)**

File Transfer Protocol(FTP) is an application layer protocol which moves files between local and remote file systems. It runs on the top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.



**What is control connection?**

For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of control connection. The control connection is initiated on port number 21.

**What is data connection?**

For sending the actual file, FTP makes use of data connection. A data connection is initiated on port number 20. FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.

**FTP Session :**

When a FTP session is started between a client and a server, the client initiates a control TCP connection with the server side. The client sends control information over this. When the server receives this, it initiates a data connection to the client side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session. As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

**Data Structures :** FTP allows three types of data structures :

1. **File Structure –** In file-structure there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure –** In record-structure the file is made up of sequential records.
3. **Page Structure –** In page-structure the file is made up of independent indexed pages.

**FTP Commands –** Some of the FTP commands are :

*USER* – This command sends the user identification to the server.  
*PASS* – This command sends the user password to the server.  
*CWD* – This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information.  
*RMD* – This command causes the directory specified in the path-name to be removed as a directory.  
*MKD* – This command causes the directory specified in the pathname to be created as a directory.  
*PWD* – This command causes the name of the current working directory to be returned in the reply.  
*RETR* – This command causes the remote host to initiate a data connection and to send the requested file over the data connection.  
*STOR* – This command causes to store a file into the current directory of the remote host.  
*LIST* – Sends a request to display the list of all the files present in the directory.  
*ABOR* – This command tells the server to abort the previous FTP service command and any associated transfer of data.  
*QUIT* – This command terminates a USER and if file transfer is not in progress, the server closes the control connection.

**FTP Replies –** Some of the FTP replies are :

200 Command okay.  
530 Not logged in.  
331 User name okay, need a password.  
225 Data connection open; no transfer in progress.  
221 Service closing control connection.  
551 Requested action aborted: page type unknown.  
502 Command not implemented.  
503 Bad sequence of commands.  
504 Command not implemented for that parameter.

**7. HTTP (Hyper Text Transfer Protocol)**

The Hypertext Transfer Protocol (HTTP) is application-level protocol for collaborative, distributed, hypermedia information systems. It is the data communication protocol used to establish communication between client and server.

HTTP is TCP/IP based communication protocol, which is used to deliver the data like image files, query results, HTML files etc on the World Wide Web (WWW) with the default port is TCP 80. It provides the standardized way for computers to communicate with each other.

**The Basic Characteristics of HTTP (Hyper Text Transfer Protocol):**

* It is the protocol that allows web servers and browsers to exchange data over the web.
* It is a request response protocol.
* It uses the reliable TCP connections by default on TCP port 80.
* It is stateless means each request is considered as the new request. In other words, server doesn't recognize the user by default.

**The Basic Features of HTTP (Hyper Text Transfer Protocol):**

There are three fundamental features that make the HTTP a simple and powerful protocol used for communication:

* **HTTP is media independent:** It specifies that any type of media content can be sent by HTTP as long as both the server and the client can handle the data content.
* **HTTP is connectionless:** It is a connectionless approach in which HTTP client i.e., a browser initiates the HTTP request and after the request is sent the client disconnects from server and waits for the response.
* **HTTP is stateless:** The client and server are aware of each other during a current request only. Afterwards, both of them forget each other. Due to the stateless nature of protocol, neither the client nor the server can retain the information about different request across the web pages.

**7.1 HTML**

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

**7.2 HTTP/1.1**

**Introduced:** 1997 (RFC 2068, improved in RFC 2616)  
**Key Features:**

* **Persistent connections:** Keep-Alive by default
* **Pipelining:** Multiple requests sent without waiting for responses (limited by Head-of-Line blocking)
* **Chunked transfer encoding:** Allows streaming content
* **Improved caching & host headers**

**Limitations:**

* Head-of-Line blocking: response order must match request order
* No compression of headers
* Poor multiplexing (one TCP connection per request limits performance)

**7.3 HTTP/2.0**

**Introduced:** 2015 (RFC 7540)  
**Built on:** Google's SPDY protocol

**Key Features:**

* **Binary protocol:** Efficient parsing and compact message framing
* **Multiplexing:** Multiple requests/responses over a single TCP connection
* **Header compression:** Using HPACK (reduces overhead)
* **Stream prioritization:** Allows better resource allocation
* **Server Push:** Server can send resources proactively

**Advantages Over HTTP/1.1:**

* Reduced latency and bandwidth usage
* Eliminates Head-of-Line blocking at HTTP level
* Improved performance for mobile and secure web
  1. **Comparison Summary:**

| **Feature** | **HTTP/1.1** | **HTTP/2.0** |
| --- | --- | --- |
| Protocol Type | Text-based | Binary |
| Multiplexing | ❌ | ✅ |
| Header Compression | ❌ | ✅ (HPACK) |
| Server Push | ❌ | ✅ |
| Connection Efficiency | Low | High |

**8. HTTPS with SSL**

**Definition:**  
**HTTPS (Hypertext Transfer Protocol Secure)** is the secure version of HTTP. It combines **HTTP** with **SSL/TLS encryption** to ensure that communication between the client (usually a web browser) and the server is **encrypted, authenticated**, and **integrity-protected**.

**8.1 What is SSL/TLS?**

* **SSL (Secure Sockets Layer)**: A cryptographic protocol developed by Netscape for secure communication.
* **TLS (Transport Layer Security)**: The modern, more secure successor to SSL (SSL is deprecated).
* HTTPS today uses **TLS** (usually TLS 1.2 or 1.3), though it is still commonly referred to as “SSL”.

**8.2 How HTTPS Works (with SSL/TLS)**

1. **Client Hello**: Browser sends a request to the server, including supported TLS versions, cipher suites, etc.
2. **Server Hello**: Server responds with its digital certificate (includes public key).
3. **Certificate Validation**: Browser validates the certificate using trusted Certificate Authorities (CAs).
4. **Key Exchange**: A secure session key is exchanged using public-key cryptography (e.g., RSA or Diffie-Hellman).
5. **Secure Communication**: All further communication is encrypted using symmetric encryption (AES, ChaCha20, etc.) with the shared session key.

**8.3 Advantages of HTTPS with SSL**

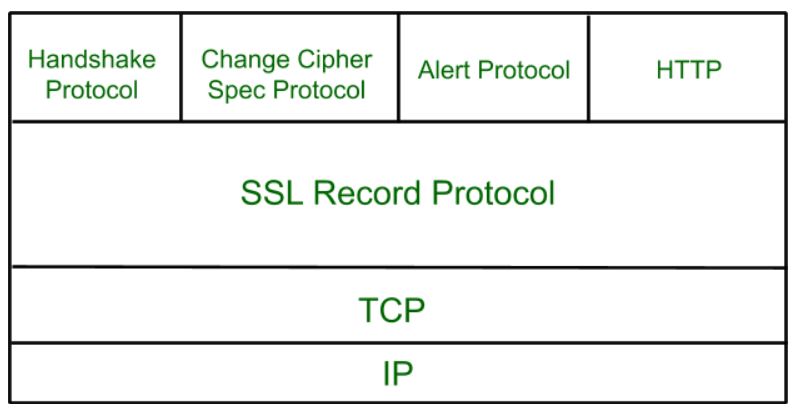
* **Encryption**: Protects data from eavesdropping.
* **Authentication**: Confirms server identity (prevents man-in-the-middle attacks).
* **Data Integrity**: Ensures that data is not altered in transit.
* **Trust**: Users trust websites with HTTPS (padlock icon in browsers).

**8.4 Limitations**

* Slight overhead in connection setup (reduced in TLS 1.3).
* Requires valid SSL/TLS certificates (can expire).
* Misconfigured certificates can lead to errors.

**8.5 Secure Socket Layer Protocols**

1. SSL Record Protocol
2. Handshake Protocol
3. Change-Cipher Spec Protocol
4. Alert Protocol

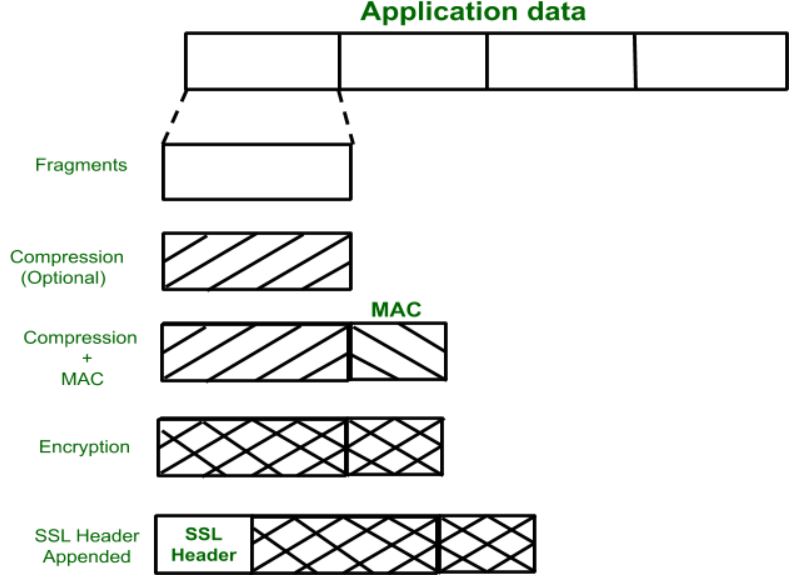


**8.5.1 SSL Record Protocol**

SSL Record provides two services to SSL connection.

* Confidentiality
* Message Integrity

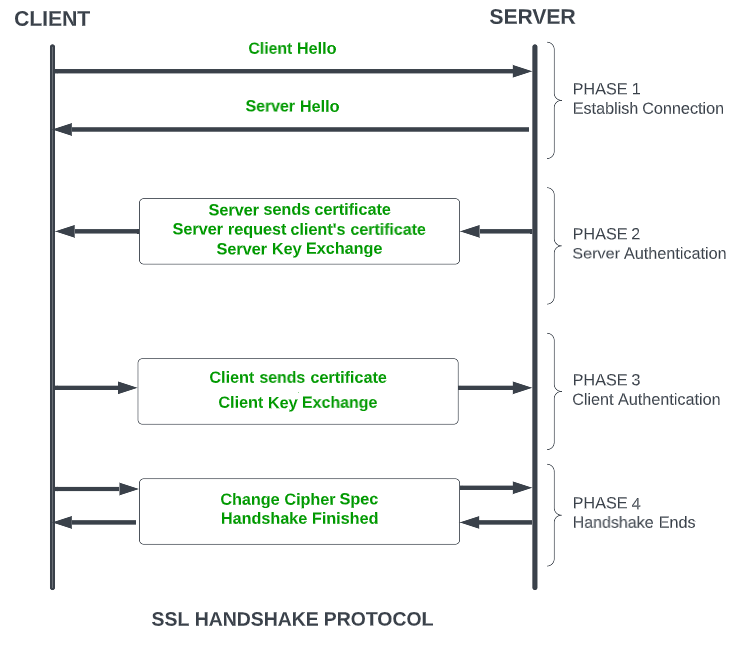
In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA ([Secure Hash Protocol](https://www.geeksforgeeks.org/sha-1-hash-in-java/)) and MD5 ([Message Digest](https://www.geeksforgeeks.org/what-is-the-md5-algorithm/)) is appended. After that encryption of the data is done and in last SSL header is appended to the data.

****

**8.5.2 Handshake Protocol**

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

* **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
* **Phase-2:** Server sends it certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
* **Phase-3:** In this phase, Client replies to the server by sending it certificate and Client-exchange-key.
* **Phase-4:** In Phase-4 Change Cipher Spec occurs and after this the Handshake Protocol ends.



**8.3 Change-Cipher Protocol**

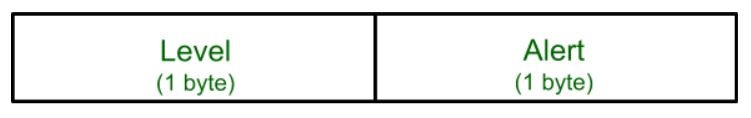
This protocol uses the SSL record protocol. Unless [Handshake](https://www.geeksforgeeks.org/tcp-3-way-handshake-process/)Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state.

Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.



**8.4 Alert Protocol**

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.



The level is further classified into two parts:

**Warning (level = 1)**

This Alert has no impact on the connection between sender and receiver. Some of them are:

* **Bad Certificate:**When the received certificate is corrupt.
* **No Certificate:** When an appropriate certificate is not available.
* **Certificate Expired:** When a certificate has expired.
* **Certificate Unknown:** When some other unspecified issue arose in processing the certificate, rendering it unacceptable.
* **Close Notify**: It notifies that the sender will no longer send any messages in the connection.
* **Unsupported Certificate:**The type of certificate received is not supported.
* **Certificate Revoked:**The certificate received is in revocation list.

**Fatal Error (level = 2):**

This Alert breaks the connection between sender and receiver. The connection will be stopped, cannot be resumed but can be restarted. Some of them are :

* **Handshake Failure:** When the sender is unable to negotiate an acceptable set of security parameters given the options available.
* **Decompression Failure**: When the decompression function receives improper input.
* **Illegal Parameters:** When a field is out of range or inconsistent with other fields.
* **Bad Record MAC:**When an incorrect MAC was received.
* **Unexpected Message:**When an inappropriate message is received.

**9. Simple Mail Transfer Protocol (SMTP)**

Email is emerging as one of the most valuable services on the internet today. Most internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver’s side.

**SMTP Fundamentals**

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly.

**SMTP Protocol**

The SMTP model is of two type :

1. End-to- end method
2. Store-and- forward method

The end to end model is used to communicate between different organizations whereas the store and forward method is used within an organization. An 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 that initiates the session so let us call it client- SMTP and the server SMTP is the one which responds to the session request so let us call it receiver-SMTP. The client- SMTP will start the session and the receiver-SMTP will respond to the request.

**Model of SMTP system**

In the SMTP model user deals with the user agent (UA), for example, Microsoft Outlook, Netscape, Mozilla, etc. In order to exchange the mail using TCP, MTA is used. The user sending the mail doesn’t have to deal with MTA as it is the responsibility of the system admin to set up a local MTA. The MTA maintains a small queue of mails so that it can schedule repeat delivery of mails in case the receiver is not available. The MTA delivers the mail to the mailboxes and the information can later be downloaded by the user agents.

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

1. User agent (UA)
2. Local MTA

**Communication between sender and the receiver :**

The sender’s user agent prepares the message and sends it to the MTA. The MTA’s responsibility is to transfer the mail across the network to the receiver’s MTA. To send mails, a system must have a client MTA, and to receive mails, a system must have a server MTA.

**SENDING EMAIL:**

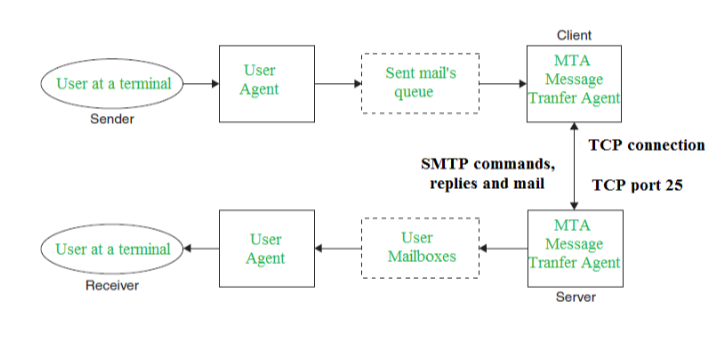
Mail is sent by a series of request and response messages between the client and the server. The message which is sent across consists of a header and a body. A null line is used to terminate the mail header and everything after the null line is considered as the body of the message, which is a sequence 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



**10. 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.

**Purpose and Functionality of MIME –**

Growing demand for Email Message as people also want to express in terms of Multimedia. So, MIME another email application is introduced as it is not restricted to textual data.

MIME *transforms non-ASCII data* at sender side to NVT 7-bit data and delivers it to the client SMTP. The message at receiver side is transferred back to the original data. As well as we can send video and audio data using MIME as it transfers them also in 7-bit ASCII 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.

**Working of MIME –**

Suppose a user wants to send an email through user agent and it is in a non-ASCII format so there is a MIME protocol which converts it into 7-bit NVT ASCII format. Message is transferred through e-mail system to the other side in 7-bit format now MIME protocol again converts it back into non-ASCII code and now the user agent of receiver side reads it and then information is finally read by the receiver. MIME header is basically inserted at the beginning of any e-mail transfer.

**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.

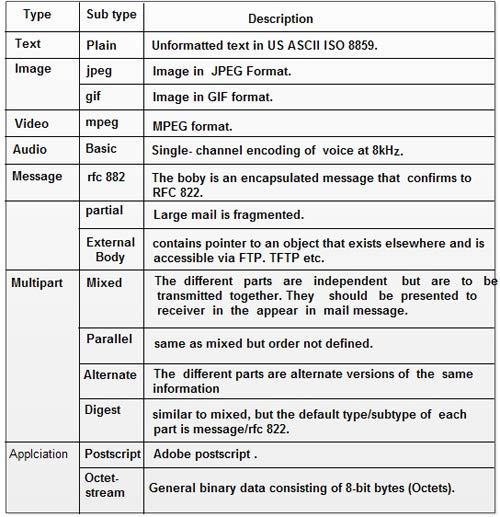
**MIME was invented to overcome the following limitations of SMTP:**

1. SMTP cannot transfer executable files and binary objects.  
2. SMTP cannot transmit text data of other language, e.g. French, Japanese, Chinese etc, as these are represented in 8-bit codes.  
3. SMTP services may reject mails having size greater than a certain size.  
4. SMTP cannot handle non-textual data such as pictures, images, and video/audio content.

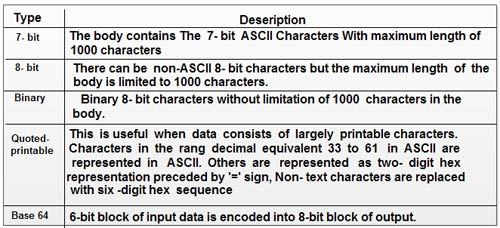
**MIME Header:**

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

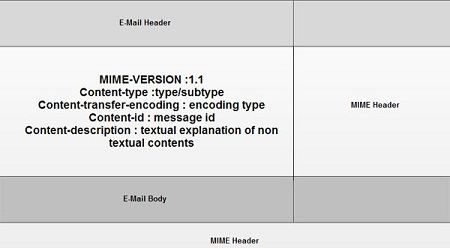
1. **MIME Version –** Defines version of MIME protocol. It must have the parameter *Value 1.0*, which indicates that message is formatted using MIME.
2. **Content Type –** Type of data used in the body of message. They are of different types like text data (plain, HTML), audio content or video content.



1. **Content Type Encoding –** It defines the method used for encoding the message. Like 7-bit encoding, 8-bit encoding, etc.



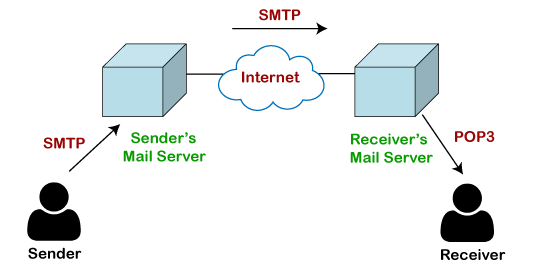
1. **Content Id –** It is used for uniquely identifying the message.
2. **Content description –** It defines whether the body is actually image, video or audio.



**11. POP Protocol**

The POP protocol stands for Post Office Protocol. As we know that SMTP is used as a message transfer agent. When the message is sent, then SMPT is used to deliver the message from the client to the server and then to the recipient server. But the message is sent from the recipient server to the actual server with the help of the Message Access Agent. The Message Access Agent contains two types of protocols, i.e., POP3 and IMAP.

**How is mail transmitted?**



Suppose sender wants to send the mail to receiver. First mail is transmitted to the sender's mail server. Then, the mail is transmitted from the sender's mail server to the receiver's mail server over the internet. On receiving the mail at the receiver's mail server, the mail is then sent to the user. The whole process is done with the help of Email protocols. The transmission of mail from the sender to the sender's mail server and then to the receiver's mail server is done with the help of the SMTP protocol. At the receiver's mail server, the POP or IMAP protocol takes the data and transmits to the actual user.

Since SMTP is a push protocol so it pushes the message from the client to the server. As we can observe in the above figure that SMTP pushes the message from the client to the recipient's mail server. The third stage of email communication requires a pull protocol, and POP is a pull protocol. When the mail is transmitted from the recipient mail server to the client which means that the client is pulling the mail from the server.

**What is POP3?**

The POP3 is a simple protocol and having very limited functionalities. In the case of the POP3 protocol, the POP3 client is installed on the recipient system while the POP3 server is installed on the recipient's mail server.

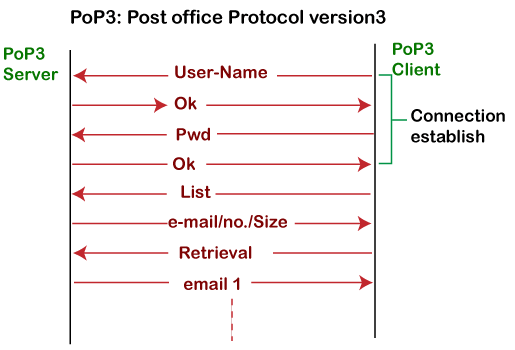
**History of POP3 protocol**

The first version of post office protocol was first introduced in 1984 as RFC 918 by the internet engineering task force. The developers developed a simple and effective email protocol known as the POP3 protocol, which is used for retrieving the emails from the server. This provides the facility for accessing the mails offline rather than accessing the mailbox offline.

In 1985, the post office protocol version 2 was introduced in RFC 937, but it was replaced with the post office protocol version 3 in 1988 with the publication of RFC 1081. Then, POP3 was revised for the next 10 years before it was published. Once it was refined completely, it got published on 1996.

Although the POP3 protocol has undergone various enhancements, the developers maintained a basic principle that it follows a three-stage process at the time of mail retrieval between the client and the server. They tried to make this protocol very simple, and this simplicity makes this protocol very popular today.

**Let's understand the working of the POP3 protocol.**



To establish the connection between the POP3 server and the POP3 client, the POP3 server asks for the user name to the POP3 client. If the username is found in the POP3 server, then it sends the ok message. It then asks for the password from the POP3 client; then the POP3 client sends the password to the POP3 server. If the password is matched, then the POP3 server sends the OK message, and the connection gets established. After the establishment of a connection, the client can see the list of mails on the POP3 mail server. In the list of mails, the user will get the email numbers and sizes from the server. Out of this list, the user can start the retrieval of mail.

Once the client retrieves all the emails from the server, all the emails from the server are deleted. Therefore, we can say that the emails are restricted to a particular machine, so it would not be possible to access the same mails on another machine. This situation can be overcome by configuring the email settings to leave a copy of mail on the mail server.

**Advantages of POP3 protocol**

**The following are the advantages of a POP3 protocol:**

* It allows the users to read the email offline. It requires an internet connection only at the time of downloading emails from the server. Once the mails are downloaded from the server, then all the downloaded mails reside on our PC or hard disk of our computer, which can be accessed without the internet. Therefore, we can say that the POP3 protocol does not require permanent internet connectivity.
* It provides easy and fast access to the emails as they are already stored on our PC.
* There is no limit on the size of the email which we receive or send.
* It requires less server storage space as all the mails are stored on the local machine.
* There is maximum size on the mailbox, but it is limited by the size of the hard disk.
* It is a simple protocol so it is one of the most popular protocols used today.
* It is easy to configure and use.

**Disadvantages of POP3 protocol**

**The following are the advantages of a POP3 protocol:**

* If the emails are downloaded from the server, then all the mails are deleted from the server by default. So, mails cannot be accessed from other machines unless they are configured to leave a copy of the mail on the server.
* Transferring the mail folder from the local machine to another machine can be difficult.
* Since all the attachments are stored on your local machine, there is a high risk of a virus attack if the virus scanner does not scan them. The virus attack can harm the computer.
* The email folder which is downloaded from the mail server can also become corrupted.
* The mails are stored on the local machine, so anyone who sits on your machine can access the email folder.

**12. What is Webmail?** 

Webmail is a cloud-based service or Web-based email system that allows you to access and use your email from almost anywhere through an internet connection. Unlike Thunderbird or Microsoft Outlook, it does not need software installation. It is a kind of service, which is provided by certain companies and ISPs (Internet service providers).

Especially, these kinds of server-based email systems are more popular for younger users. As with Microsoft Outlook, where emails are stored on-site in the hardware storage drive and logging into a connection with the server is needed to get email; so, in this situation, these services provide an appropriate option to email services.

For people who frequently away from their computers and use multiple devices, Webmail is a great solution for those people. Gmail, Hotmail, and other mainstream providers are the common examples of webmail from Yahoo!, which offer huge amounts of storage, and almost all are free.

They are very calm to set up and use. Although experts have pointed out, these models have advantages and limitations. With client-side email, users do not need an internet connection to review old emails as they can be archived directly on the computer. However, with webmail, you always need an internet connection to review mails as they are available via the dedicated servers over a network connection. Like some resident systems, webmail systems do not need communications protocols; that is one of another benefit of webmail. Some of the less tech-savvy users are frustrated by mail delivery failures while using continue resident or non-webmail systems, but a webmail product helps to prevent that issue.

**Why is webmail so popular?**

Webmail accounts allow users to send, reply, read, organize their email into saving attachments and folders without the need of installing or using an email application software like Microsoft Outlook. The webmail service provides you a web page for accessing your email account and holds all of your emails on their computer systems and storage systems.

Some popular webmail services

In modern times, many webmail services are available for users, which are not software-based. Below, a list contains some the free webmail services.

* **Gmail:** Gmail is a type of Webmail, a free Web-based e-mail service that allows users a gigabyte of storage for messages or other data. It is a very popular email service developed by Google. There are 1.5 billion active users of Gmail by October 2019.
* **Yahoo! Mail:** It is a web and cloud-based messaging solution that is launched by the American company Yahoo! on 8 October 1997. You can connect with your email with one-tap access to your inbox with the help of Yahoo! Mail, and it had 225 million users by January 2020. You can use to create Yahoo account by using this link -<https://overview.mail.yahoo.com/>
* **com:** It is a free web-based e-mail service that allows you to send and receive e-mail on your computer. Somewhat, it is like Google's Gmail service but something different in terms of linking desktop Outlook data. Outlook has two types of versions: Microsoft Outlook and Microsoft Outlook Express. To create an Outlook account, you can use this link <https://signup.live.com/?lic=1>
* **ProtonMail:** Unlike Gmail and Outlook.com., it uses client-side encryption to protect user data and email content, which is founded in 2013. To create a ProtonMail account, use this link - <https://protonmail.com/>
* **Zoho:** It holds a lot of potential for businesses, which is the first of the lesser-known free email accounts for making a list. It is an email service that very user-friendliness. It provides an easier way to accomplish all of your daily tasks by integrating with Google Drive, cloud-based file managers, Box.
* **GMX Mail:** GMX Mail is a free advertising-supported email service that may be accessed via POP3 and IMAP4 protocols as well as through webmail. It is provided by GMX (Global Mail eXchange) in Germany in 1997 that offers 65GB of storage.

**Are all webmail accounts the same?**

Commonly, they are also known as IMAP accounts, as most webmail accounts use IMAP. With a web browser, all webmail accounts cannot be used. IMAP short for Internet Message Access Protocol, which is a kind of email account service. On the email service's computers, this kind of email protocol keeps a master copy of all emails; therefore, it is used by numerous webmail accounts. Without the need for software running the email service, any email account can be accessed by webmail over the internet using a Web browser.

Microsoft Exchange is not IMAP, which is used in business and government as it is a common application. However, a webmail option is included in the Microsoft Exchange, known as OWA (Outlook Web Access).

**Advantages of Webmail**

As Webmail allows you to access and use your email through an internet connection; it offers various you benefit. Some are discussed below:

**Convenience**

One of the most important advantages of webmail is convenience, which is beneficial at the time when you travel frequently and work from remote locations. With the help of using an internet connection, you can access and use it through any computer. Public access terminals like libraries and hotels are included in it. No matter where you access your account because all your messages will be stored centrally; hence, you can review all your messages.

**Cost**

Usually, big providers provide basic webmail services free of cost. Also, some premium services are also offered by some providers.

**Retain Your Address**

Another benefit of webmail is that you do not need to change your old email address, even if you have changed Internet providers. That allows you to restore your old contacts, and when your contacts are trying to get in touch with you, they do not need to worry about all of your contacts remembering your latest address.

**Large Storage**

One of the biggest advantages of webmail is storage capacity. It means the storage in gigabytes or unlimited storage is offered by big webmail providers that allows you to save a large number of messages as well as large attachments. Therefore, freeing up space for other things, you do not require to keep all that data on your own computer.

**Disadvantages of Webmail**

**Personal Name**

You may be unable to create an address on webmail using the name you had in mind or your own name as it has a large customer base. Whenever you try to create an address on webmail with your own name, it may be this name has taken by another one. Therefore, you cannot address using your own name. You can create an address with a memorable and simple email address that is available. Also, using webmail for business purpose may be a disadvantage because it is considered by many people that the webmail addresses is unprofessional.

**Ads**

From advertisers, webmail providers have to get their revenue as you are using the service for free of cost. That means you are not allowed to carry personalized advertising, something traditional email services, and even you will have to put up with banners.

**More Spam**

Because of spammers' large customer base, they tend to target webmail services more as compared to traditional providers. Therefore, the spammer can find your address and bombard you even if you are cautious not to sign up for commercial services.

**No Offline Working**

Although broadband becomes more appropriate in terms of reliability, both are required to connect your device with the network to write, send messages, and as well as review messages. This means you will be unable to look at old emails if you are not connected to the internet, and even you cannot download your latest messages. Thus, a message cannot be composed offline and then connect and send it.

**History of Webmail**

In 1993, Web Mail was implemented for the first time as a test of the HTTP protocol stack and developed by Phillip Hallam-Baker at CERN, but its development was not continue. However, working webmail applications were produced by several people in the next two years.

Søren Vejrum's "WWW Mail", Remy Wetzels' "WebMail" and Luca Manunza's "WebMail," these were three implementations of webmail in Europe. Søren Vejrum's "WWW Mail" was released on February 28, 1995, which was whitened at the time he was studying and working at the Copenhagen Business School in Denmark. Remy Wetzels' "WebMail" was released in early January 1995 while he was studying for the DSE at the Eindhoven University of Technology in the Netherland. Later 'Webex' was written by Matt Mankins and Lotus cc: Mail written by Bill Fitler that he demonstrated publicly at Lotusphere on January 24, 1995.

On 8 August 1995, at the University of Miami, Matt Mankins released his "Webex" application source code in a post to comp.mail.misc under the supervision of Dr. Burt Rosenberg. However, at the School of Construction, it was used as the main email application.

The development of Bill Fitler's webmail implementation was continued as a commercial product, which was announced and released for the World Wide Web 1.0 by Lotus in the fall of 1995 as cc: Mail. Later the name of Webex has changed to EMUmail within DotShop. And, before the sale to AccuRev in 2001, it would be sold to companies like UPS and Rackspace. Hotmail and Four11's RocketMail became very popular after they launched in 1996 as free services.

In the 1990s, progress increased in the field of webmail, and it became very easy to have access to webmail by the 2000s because:

* Web hosting providers like Verio and internet providers like EarthLink started to include webmail into their service providing.
* Many large corporations and universities began providing webmail in terms of accessing email to their user communities.
* In order to as a free service to the general public, webmail service providers like Hotmail and RocketMail emerged in 1996 and rapidly became popular.

In some cases, webmail application software is obtained from software companies as part of an integrated mail server package. Netscape Messaging Server is an early example. In the 2010s, the sale of webmail application software has continued in the market.