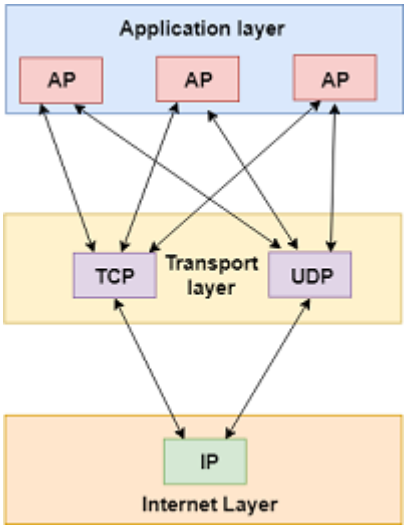
UNIT – 5

Transport Layer

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

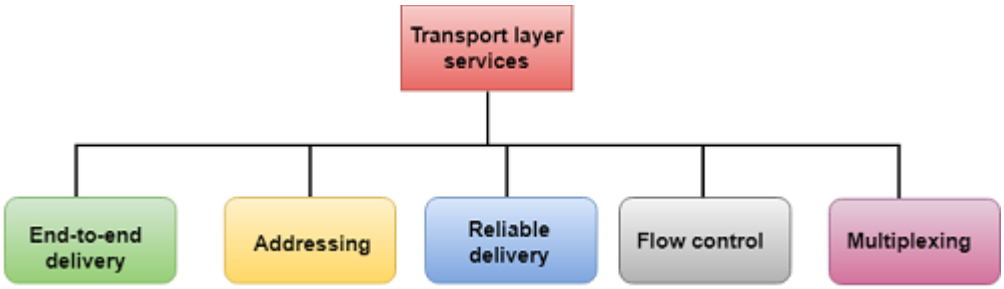


Services provided by the Transport Layer

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

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

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



End-to-end delivery:

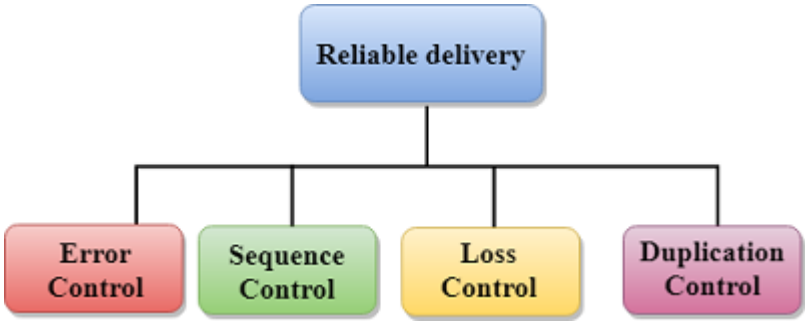
The transport layer transmits the entire message to the destination. Therefore, it ensures the end-to-end delivery of an entire message from a source to the destination.

Reliable delivery:

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

The reliable delivery has four aspects:

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

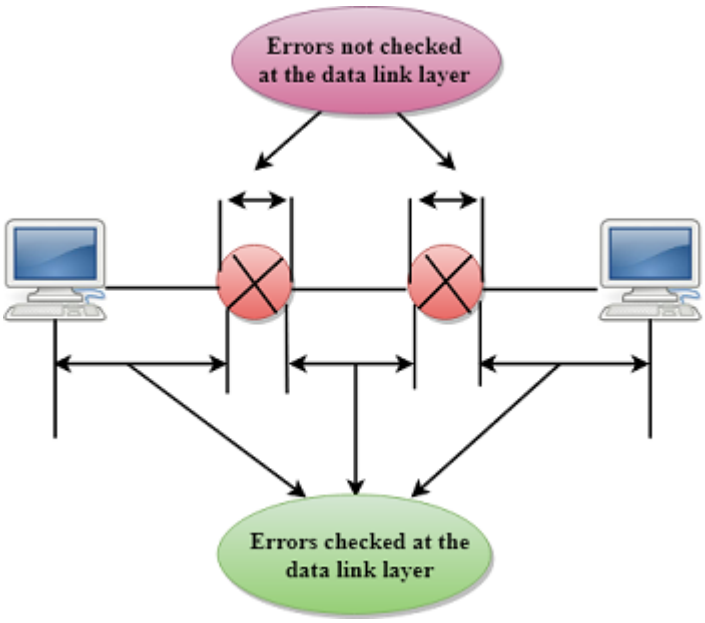


Error Control

The primary role of reliability is Error Control. In reality, no transmission will be 100 percent error-free delivery. Therefore, transport layer protocols are designed to provide error-free transmission.

The data link layer also provides the error handling mechanism, but it ensures only node-to-node error-free delivery. However, node-to-node reliability does not ensure the end-to-end reliability.

The data link layer checks for the error between each network. If an error is introduced inside one of the routers, then this error will not be caught by the data link layer. It only detects those errors that have been introduced between the beginning and end of the link. Therefore, the transport layer performs the checking for the errors end-to-end to ensure that the packet has arrived correctly.



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.

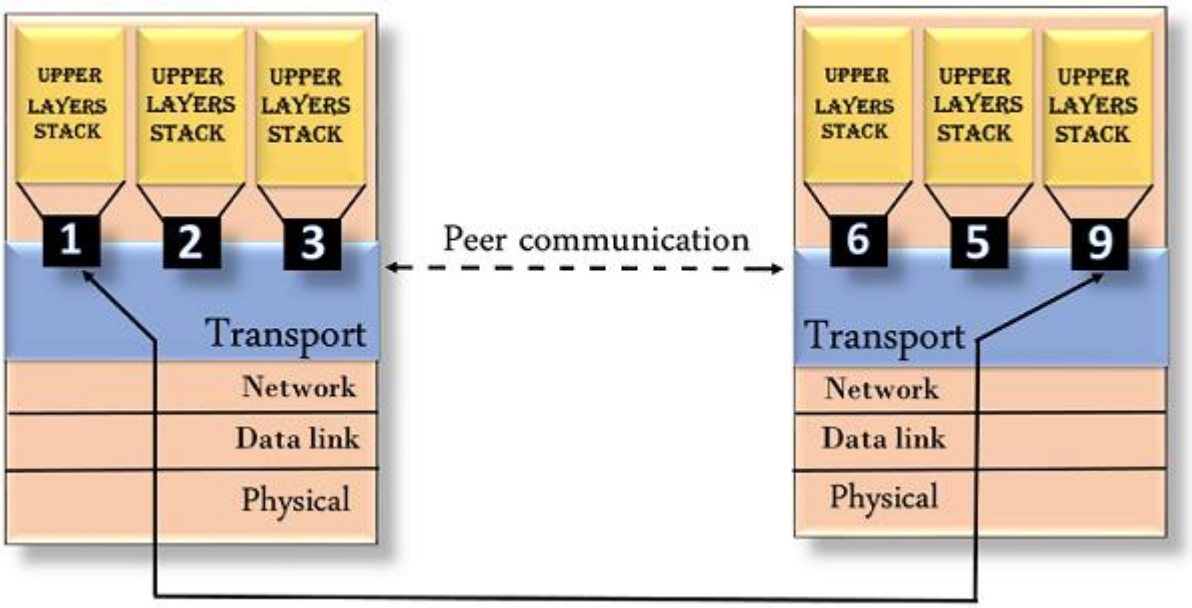
Multiplexing

Multiplexing allows simultaneous use of different applications over a network that 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.

Addressing

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



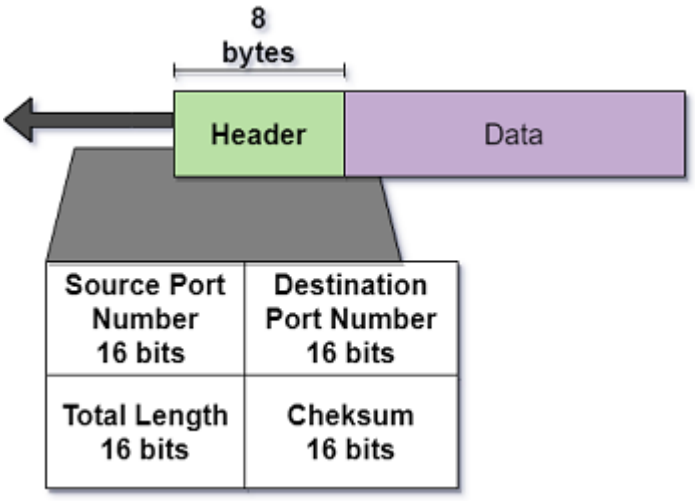
UDP Protocol

UDP is a short form for User Datagram protocol. It is one of the simplest transport layer protocol. It is a connectionless and unreliable transport protocol.

- This protocol is mainly designed in order to data send data packets over the Internet.
- This protocol does not add anything to the services of IP(Internet protocol) except that UDP provides process-to-process communication rather than host-to-host communication.
- UDP performs a very limited amount of error checking.
- This protocol uses the minimum number of overhead. Suppose if a process wants to send a small message with no concern of reliability then it can make the use of UDP.
- This protocol simply takes the datagram from the network layer, attaches its own header, and then sends it back to the user.

User Datagram

The UDP packets are commonly known as User Datagram and the size of the header is fixed that is 8 bytes.



1. Source Port Number

This port number is mainly used by the process that is running on the source host. Its length is **16 bits** which means that the port number can range from **0 to 65,535.** It is mainly used to identify the port of the sending or **source application.**

In case if the source host is a **client** (that mainly sends a request), then in most cases the ephemeral port number is requested by the process and also chosen by UDP software that runs on the source host.

But in the case of the **server as a source host**(mainly a server sending a response) then the port number is a well-known port number in such cases.

2. Destination Port Number

This port number is mainly used by the process that is running on the **destination host**. Its length is **16 bits.**

If the **destination host is the server** (a client sending the request), then the **port number is a well-known port number**.

In case if the **destination host is a client**(server sending the response), then in most cases the **port number** is an **ephemeral port number**.

3. Length

This field of the datagram header is mainly used to identify the combined length of UDP Header and Encapsulated data. It is a 16-bit field.

```
UDP Length= Length of UDP header + Length of Encapsulated data
```

4. Checksum

This field is mainly used to detect errors. It is a 16-bit field.

- The checksum calculation is although not mandatory in the User datagram protocol.

Characteristics of UDP

Given below are some of the characteristics of the User Datagram protocol:

- UDP is an unreliable and connectionless protocol.

- It is almost a Null Protocol.
- If the flow of data is in one direction, then it is a good protocol.
- This protocol does not guarantee the delivery of the data.
- No congestion control mechanism is provided by this protocol.
- UDP offers Minimal transport services.
- It is a stateless protocol.

Services of UDP

Given below are different services of UDP:

1. Connectionless Services

The User datagram protocol offers Connectionless Services which simply means that each user datagram that is sent by the UDP is an independent datagram. In different datagrams, there is no relationship, even if they are coming from the same source process and also going to the same destination program.

User datagrams are not numbered, there is no connection establishment and no connection termination.

Each datagram mainly travels through different paths.

2. Flow Control and Error Control

User datagram is a very simple and unreliable transport protocol. It does not provide any flow control mechanism and hence there is no window mechanism. Due to which the receiver may overflow with the incoming messages.

No error control mechanism is provided by UDP except checksum. Due to which the sender does not know if any message is has been lost or duplicated.

As there is a lack of flow control and error control it means that the process that uses the UDP should provide these mechanisms.

3. Encapsulation and decapsulation

In order to send the message from one process to another, the user datagram protocol encapsulates and decapsulates the message in the form of an IP datagram.

Applications of UDP

Given below are some applications of the User datagram protocol:

- **UDP** is used by those applications that require one response for one request.
- It is used by **broadcasting and multicasting applications.**
- Management processes such as **SNMP** make use of **UDP**.
- Route updating protocols like **Routing Information Protocol(RIP)** make use of User Datagram Protocol.
- The process that has an error and flows control mechanism makes use of UDP. One Application for the same is **Trivial File Transfer Protocol(TFTP).**

Advantages of UDP

Given below are some advantages of UDP:

- With UDP, broadcast and multicast transmission is possible.
- UDP uses the bandwidth efficiently, as there is a small packet overhead.
- As there is no need for connection establishment, hence UDP is very fast.
- There is no buffering and numbering of packets.
- There is no need for handshaking.
- There is no congestion control so it is used for real-time applications.

Disadvantages of UDP

Now its time to take a look at UDP:

- 1.There is a lack of guaranteed delivery.
- 2.There is no flow control.
- 3.There is no congestion control mechanism.

TCP - Transmission Control Protocol

TCP is an abbreviation of **Transmission Control Protocol**. This is a Transport Layer Protocol.TCP is a connection-oriented protocol. It is a reliable protocol used for transport. This protocol adds connection-oriented and reliability features to the services of the Internet Protocol.

This protocol seeks to deliver a stream of bytes from end-to-end in a particular order.

As it is the responsibility of the transport layer to provide end-to-end communication, thus TCP plays a very important role in the transport layer.

Features of Transmission Control Protocol

Given below are the features of TCP let us take a look at them:

1. Numbering System

There are two fields in TCP mainly sequence number and acknowledgment number. These two fields in the TCP mainly refers to Byte Number.

- **Byte Number:** The bytes of data that are being transferred in each connection are numbered by TCP. The numbering mainly starts with a randomly generated number.
  - TCP mainly numbers all the data bytes that are transmitted in a connection.
  - It generates a random number between 0 and 2 raised to the power of 32 -1 for the number of the first byte.
  - Example: If the random no. is 1056 and there are a total of 6000 bytes to be sent then the bytes are numbered from 1056 to 7055.

- **Sequence Number:** After the numbering of bytes, the TCP makes the grouping of bytes in the form of "segments".
  - A sequence is assigned to each segment that is being sent.
  - The Sequence number of each segment is the number of the first byte that is carried in that segment.
  - Thus the value in the sequence number field of the segment mainly defines the number of the first data byte that is contained in that segment.

- **Acknowledgment Number:** The value of the acknowledgment field in the segment mainly defines the number of the next byte that a party mainly expects to receive.
  - It is cumulative in nature.

2. Flow Control

The TCP provides the facility of Flow control. With the help of TCP, the receiver of the data control the amount of the data that are to be sent by the sender.

- The flow control is mainly done in order to prevent the receiver from being overwhelmed with the data
- The numbering system also allows the TCP to use byte-oriented flow control.

3. Error Control

As TCP provides reliable services, thus it implements an error control mechanism for this purpose. The Error control though considers the segment as the unit of data for error detection. Error control is byte-oriented in nature.

4. Congestion Control

Another main feature of TCP is that it facilitates Congestion Control in the network. The Amount of the data that the sender sends is not only controlled by the receiver, but congestion in the network also determines it.

5. Full Duplex

TCP provides another feature and that is Full Duplex which means by using TCP the data can be transmitted in both directions.

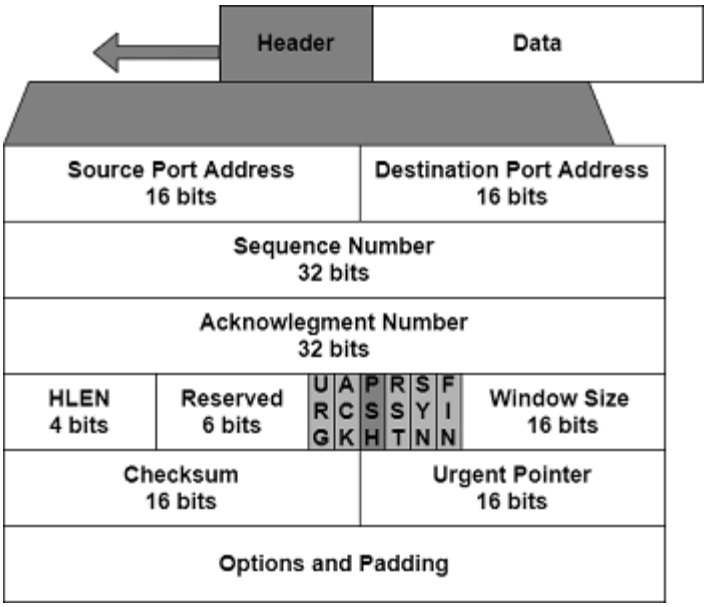
6. TCP is a transport layer protocol because it is mainly used to transmit the data from the sender to the receiver.

Segment

The packet in the TCP is mainly known as a segment.

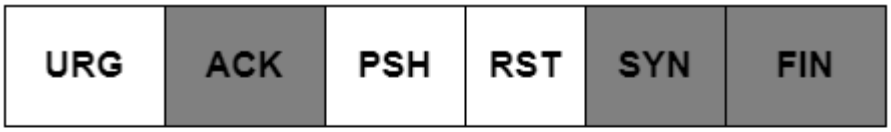
Format

Given below is the format of the TCP packet let us take a look at it:



The Segment mainly consists of a 20- to 60-byte header that is followed by the application program. Usually, the header is 20 bytes but sometimes in case if there are no options then it goes up to 60 bytes if it contains many options.

- **Source Port Address:** It is a 16-bit field and is mainly defines the port number of the application program in the host that is mainly used for sending the segment. The purpose of the Source port address is the same as the source port address in the header of the UDP.
- **Destination Port Address:** This is also a 16-bit address and is mainly defines the port number of the application program in the host that is mainly used for receiving the segment. The purpose of the Destina**ti**on port address is the same as the destination port address in the header of the UDP.
- **Sequence Number:** It is a 32-bit field that mainly defines the number assigned to the first byte of data that is contained in the segment.
- **Acknowledgment Number:** It is also a 32-bit field and is mainly used to define the byte number that the receiver of the segment is expecting to receive from the other party.
- **Header Length:** It is a 4-bit field and is mainly used to indicate the number of 4-byte words in the TCP header. The length of the header lies between 20 and 60 bytes.
- **Reserved:** It is a 6-bit field and is mainly reserved for future use.
- **Control:** This field mainly defines 6 different control bits or flags and among all only one can be set at that time.



URG: Urgent Pointer is Valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the Connection
SYN: Synchronize sequence Numbers
FIN:Terminate the Connection

These bits mainly enables the flow control, connection establishment, termination, and modes of transferring the data in TCP.

- **Window Size:** This field is mainly used to define the size of the window. The size of this field is 16-bit. It mainly contains the size of the data the receiver can accept. The value of this field is mainly determined by the receiver.
- **Checksum:** It is a 16-bit field and is mainly contains the checksum. This field is mandatory in the case of TCP/IP.
- **Urgent Pointer:** The size of this field is 16-bit and it is only valid in the case if the urgent flag is set. This field is used only when the segment contains urgent data.
- **Options:** This field is represented in 32 bits.

TCP Connection

As we know that TCP is a connection-oriented protocol and it means this protocol establishes a virtual path between the source and the destination. All the segments that belong to the message are then sent over this virtual path.

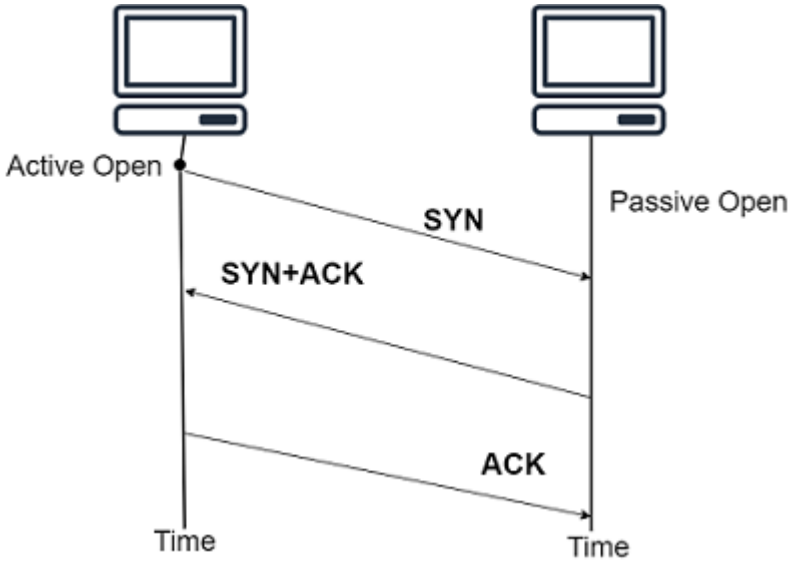
In TCP, the connection-oriented transmission mainly requires three phases and these phases are:

- Connection Establishment Phase
- Data Transfer Phase
- Connection Termination.

Connection Establishment Phase

Transmission of data is done in full-duplex mode. The connection establishment in TCP is mainly termed as three-way handshaking. Let us understand this with the help of an example: An application program called client wants to make a connection with another application program called server by using the TCP as the transport layer protocol.

Mainly the process starts with the server, the server program mainly tells the TCP that it is ready to accept a connection. This is mainly known as a request for a passive open. Though the server TCP is ready to accept any connection from any machine of the world and it cannot make the connection itself.



SYN

- This flag is used for the synchronization of sequence numbers.
- It does not carry any real data.
- It mainly consumes 1 sequence number.

SYN+ACK

- This is mainly used for synchronization in other directions and ACK for the signal received.
- It does not carry any real data.
- It also consumes 1 sequence number.

ACK

- It is just an ACK segment.
- It does not consume any sequence number if it does not carry any data.

Data Transfer Phase

After the establishment of the connection, the bidirectional data transfer can take place. Both the client and server can send data and acknowledgments.

Connection Termination Phase

The two parties that are involved in the data exchange can close the connection, although it is initiated usually by the client. There ate two ways for the connection termination:

- Three-way handshaking
- four-way handshaking with a half close option.

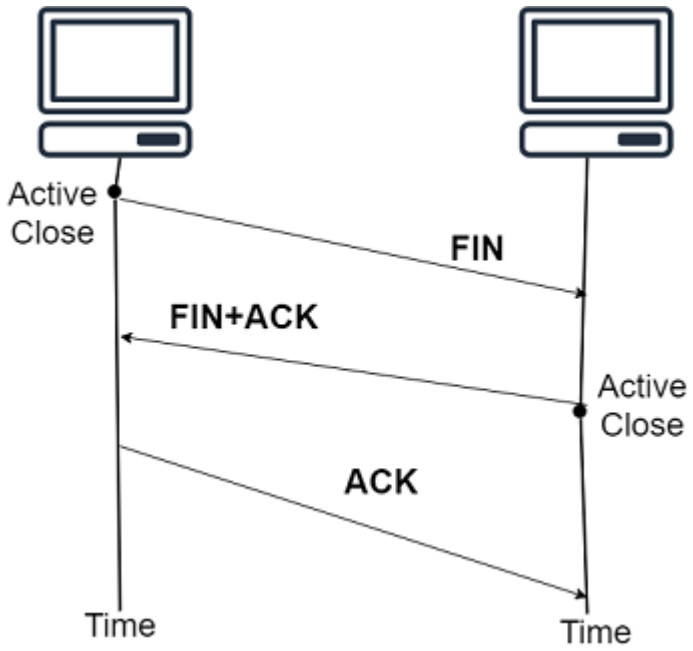


Figure shows Connection Termination using three-way handshaking

FIN

- This segment consumes one sequence number if it does not carries any data.
- It may or may not carries any real data.

FIN+ACK

- This segment consumes consequence numbers if it does not carries any data.
- The FIN segment announces the closing of the connection in another direction.
- ACK is for received FIN.
- It consumes only 1 sequence number.

ACK

- It is just an ACK segment.
- It does not consume any sequence number.

Advantages of TCP

Given below are some of the advantages of TCP:

- TCP performs data control and flow control mechanisms.
- TCP provides excellent support for cross-platform.
- The TCP protocol ensures the guaranteed delivery of the data.
- It transmits the data from the sender to the receiver in a particular order.
- It is a connection-oriented and reliable protocol.
- It has a good relative throughput on the modem or on the LAN.
- Provides error detection mechanism by using the checksum and error correction mechanism is provided by using ARP or go-back protocol.

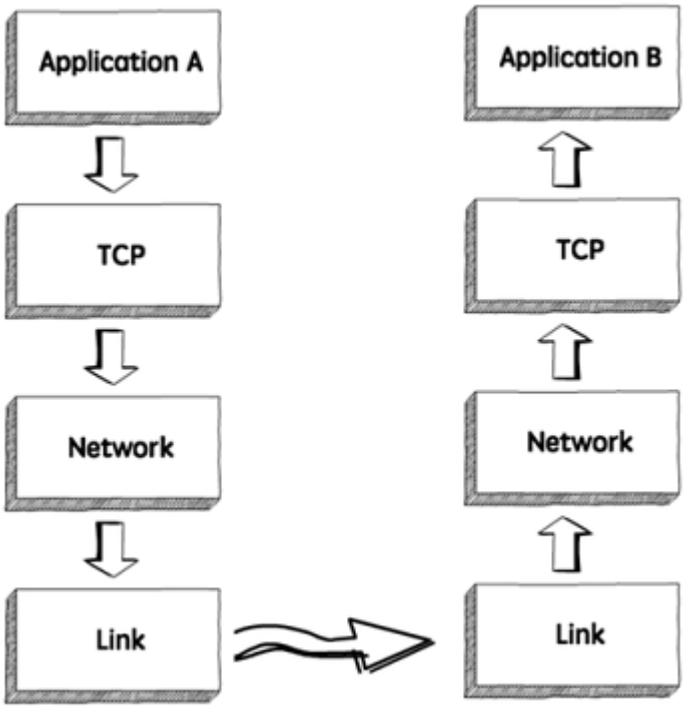
Disadvantages of TCP

Given below are the disadvantages of TCP:

- It cannot be used for broadcast or multicast transmission.
- There is an increase in the amount of overhead.

TCP Flow Control

TCP Flow Control is a protocol designed to manage the data flow between the user and the server. It ensures that there is a specific bandwidth for sending and receiving data so the data can be processed without facing any major issues. In order to achieve this, the TCP protocol uses a mechanism called the sliding window protocol.



The sliding window protocol

In the **sliding window protocol** method, when we are establishing a connection between sender and receiver, there are two buffers created. Each of these two buffers are assigned to the sender, called the **sending window**, and to the receiver, called the **receiving window**.

When the sender sends data to the receiver, the receiving window sends back the remaining receiving buffer space. As a result, the sender cannot send more data than the available receiving buffer space. We'll understand the concept better once we take a look at the illustration below:

Explanation

In this example, the sending window sends data to the receiving window. The receiving window sends the acknowledgment after receiving the data and then the sending window sends another data frame.

However, this time, along with the received acknowledgment, the receiving window also sends another message saying that the available memory is full.

The sending window pauses the transmission of data until it gets the acknowledgment of the receiving window that space has been released and it can continue the transmission process.

Error Control in TCP

TCP protocol has methods for finding out corrupted segments, missing segments, out-of-order segments and duplicated segments.

Error control in TCP is mainly done through the use of **three simple techniques** :

- **Checksum –** Every segment contains a checksum field which is used to find corrupted segments. If the segment is corrupted, then that segment is discarded by the destination TCP and is considered lost.
- **Acknowledgement –** TCP has another mechanism called acknowledgement to affirm that the data segments have been delivered. Control segments that contain no data but have sequence numbers will be acknowledged as well but ACK segments are not acknowledged.
- **Retransmission –** When a segment is missing, delayed to deliver to a receiver, corrupted when it is checked by the receiver then that segment is retransmitted again. Segments are retransmitted only during two events: when the sender receives three duplicate acknowledgements (ACK) or when a retransmission timer expires.
    - **Retransmission after RTO:** TCP always preserves one retransmission time-out (RTO) timer for all sent but not acknowledged segments. When the timer runs out of time, the earliest segment is retransmitted. Here no timer is set for acknowledgement. In TCP, the RTO value is dynamic in nature and it is updated using the round trip time (RTT) of segments. RTT is the time duration needed for a segment to reach the receiver and an acknowledgement to be received by the sender.
    - **Retransmission after Three duplicate ACK segments:** RTO method works well when the value of RTO is small. If it is large, more time is needed to get confirmation about whether a segment has been delivered or not. Sometimes one segment is lost and the receiver receives so many out-of-order segments that they cannot be saved. In order to solve this situation, three duplicate acknowledgement method is used and missing segment is retransmitted immediately instead of retransmitting already delivered segment. This is a fast retransmission

because it makes it possible to quickly retransmit lost segments instead of waiting for timer to end.

TCP 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

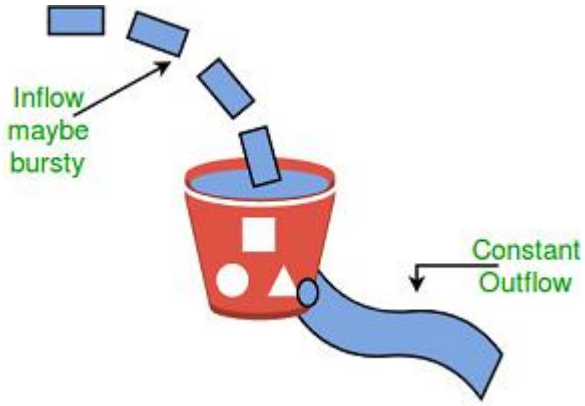
- Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.
- Congestive-Avoidance Algorithms (CAA) are implemented at the TCP layer as the mechanism to avoid congestive collapse in a network.
- There are two congestion control algorithm which are as follows:

Leaky Bucket Algorithm

- The leaky bucket algorithm discovers its use in the context of network traffic shaping or rate-limiting.
- A leaky bucket execution and a token bucket execution are predominantly used for traffic shaping algorithms.
- This algorithm is used to control the rate at which traffic is sent to the network and shape the burst traffic to a steady traffic stream.
- The disadvantages compared with the leaky-bucket algorithm are the inefficient use of available network resources.
- The large area of network resources such as bandwidth is not being used effectively.

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

- The leaky bucket algorithm has a rigid output design at an average rate independent of the bursty traffic.
- In some applications, when large bursts arrive, the output is allowed to speed up. This calls for a more flexible algorithm, preferably one that never loses information. Therefore, a token bucket algorithm finds its uses in network traffic shaping or rate-limiting.
- It is a control algorithm that indicates when traffic should be sent. This order comes based on the display of tokens in the bucket.
- The bucket contains tokens. Each of the tokens defines a packet of predetermined size. Tokens in the bucket are deleted for the ability to share a packet.
- When tokens are shown, a flow to transmit traffic appears in the display of tokens.

- No token means no flow sends its packets. Hence, a flow transfers traffic up to its peak burst rate in good tokens in the bucket.

TCP uses a congestion window and a congestion policy that avoid congestion. Previously, we assumed that only the receiver can dictate the sender's window size. We ignored another entity here, the network. If the network cannot deliver the data as fast as it is created by the sender, it must tell the sender to slow down. In other words, in addition to the receiver, the network is a second entity that determines the size of the sender's window.

Congestion policy in TCP –

1. Slow Start Phase: starts slowly increment is exponential to threshold
2. Congestion Avoidance Phase: After reaching the threshold increment is by 1
3. Congestion Detection Phase: Sender goes back to Slow start phase or Congestion avoidance phase.

Slow Start Phase: exponential increment – In this phase after every RTT the congestion window size increments exponentially.

Initially cwnd = 1

After 1 RTT, cwnd = 2^(1) = 2

2 RTT, cwnd = 2^(2) = 4

3 RTT, cwnd = 2^(3) = 8

Congestion Avoidance Phase:

Additive increment: This phase starts after the threshold value also denoted as *ssthresh*. The size of *cwnd*(congestion window) increases additive. After each RTT cwnd = cwnd + 1.

Initially cwnd = i

After 1 RTT, cwnd = i+1

2 RTT, cwnd = i+2

3 RTT, cwnd = i+3

Multiplicative decrement: If congestion occurs, the congestion window size is decreased. The only way a sender can guess that congestion has occurred is the need to retransmit a segment. Retransmission is needed to recover a missing packet that is assumed to have been dropped by a router due to congestion. Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.

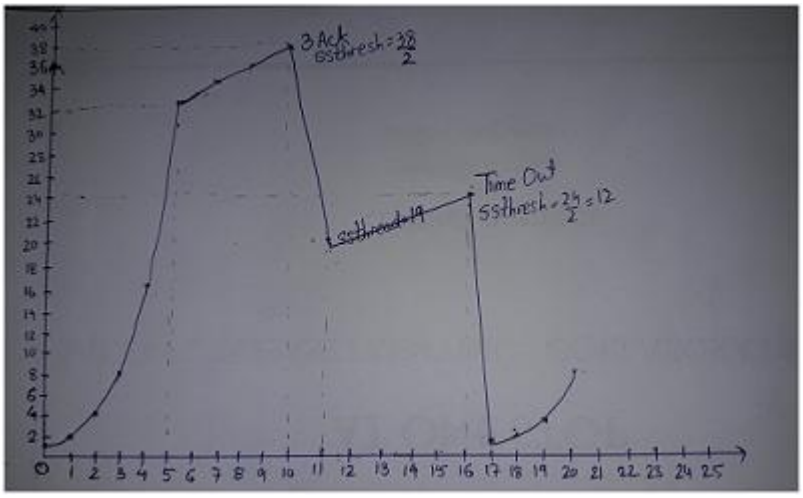
- **Case 1: Retransmission due to Timeout:** In this case congestion possibility is high.

a) ssthresh is reduced to half of the current window size.
b) set cwnd = 1
c) start with slow start phase again.

- **Case 2: Retransmission due to 3 Acknowledgement Duplicates:** In this case congestion possibility is less.

a) ssthresh value reduces to half of the current window size.
b) set cwnd= ssthresh
c) start with congestion avoidance phase

Example: Assume a TCP protocol experiencing the behavior of slow start. At 5th transmission round with a threshold (ssthresh) value of 32 goes into congestion avoidance phase and continues till 10th transmission. At 10th transmission round, 3 duplicate ACKs are received by the receiver and enter into additive increase mode. Timeout occurs at 16th transmission round. Plot the transmission round (time) vs congestion window size of TCP segments.



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.

HTTP

- HTTP stands for **HyperText Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.
- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

Messages

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

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

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

Electronic Mail

Electronic mail is often referred to as E-mail and it is a method used for **exchanging digital messages**.

- Electronic mail is mainly designed for **human use**.
- It allows a message to includes **text, image, audio** as well as **video**.
- This service allows one message to be **sent to one or more than one recipient**.
- The E-mail systems are mainly based on the **store-and-forward model** where the E-mail server system accepts, forwards, deliver and store the messages on behalf of users who only need to connect to the infrastructure of the Email.

- The Person who **sends the email** is referred to as **the Sender** while the person who receives an email is referred to as **the Recipient**.

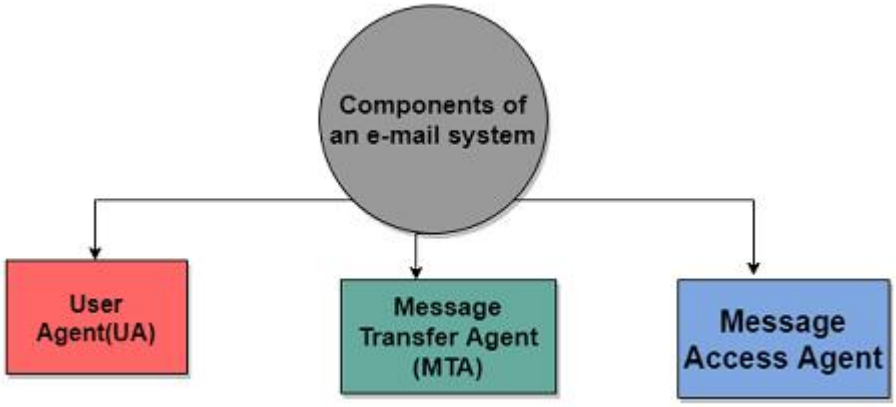
Need of an Email

By making use of Email, we can send any message at any time to anyone.

- We can send the same message to several peoples at the same time.
- It is a very fast and efficient way of transferring information.
- The email system is very fast as compared to the Postal system.
- Information can be easily forwarded to coworkers without retyping it.

Components of E-mail System

The basic Components of an Email system are as follows:



1. User Agent(UA)

It is a program that is mainly used to send and receive an email. It is also known as an email reader. User-Agent is used to compose, send and receive emails.

- It is the first component of an Email.
- User-agent also handles the mailboxes.
- The User-agent mainly provides the services to the user in order to make the sending and receiving process of message easier.

Given below are some services provided by the User-Agent:

- Reading the Message
- Replying the Message
- Composing the Message
- Forwarding the Message.

- Handling the Message.

2. Message Transfer Agent

The actual process of transferring the email is done through the Message Transfer Agent(MTA).

- In order to send an Email, a system must have an MTA client.
- In order to receive an email, a system must have an MTA server.
- The protocol that is mainly used to define the MTA client and MTA server on the internet is called SMTP(Simple Mail Transfer Protocol).
- The SMTP mainly defines how the commands and responses must be sent back and forth

3. Message Access Agent

In the first and second stages of email delivery, we make use of SMTP.

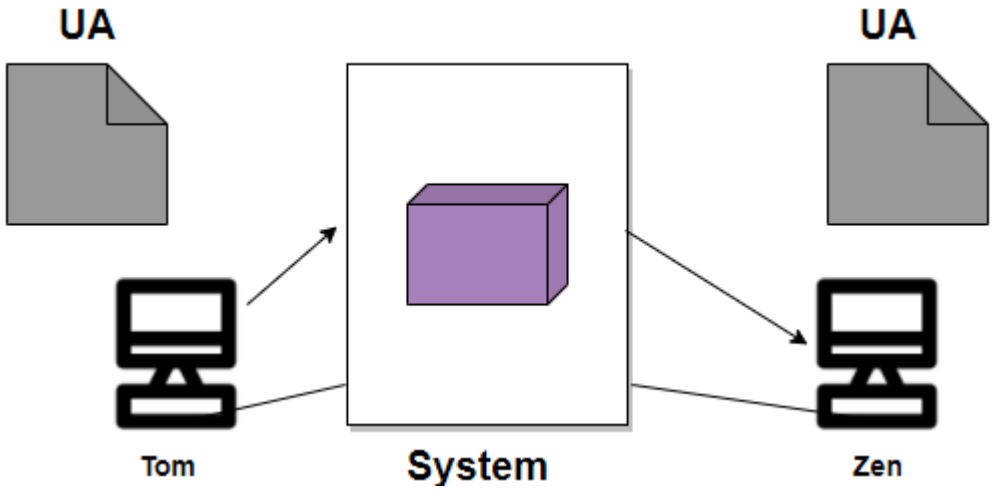
- SMTP is basically a Push protocol.
- The third stage of the email delivery mainly needs the pull protocol, and at this stage, the message access agent is used.
- The two protocols used to access messages are POP and IMAP4.

Architecture of Email

Now its time to take a look at the architecture of e-mail with the help of four scenarios:

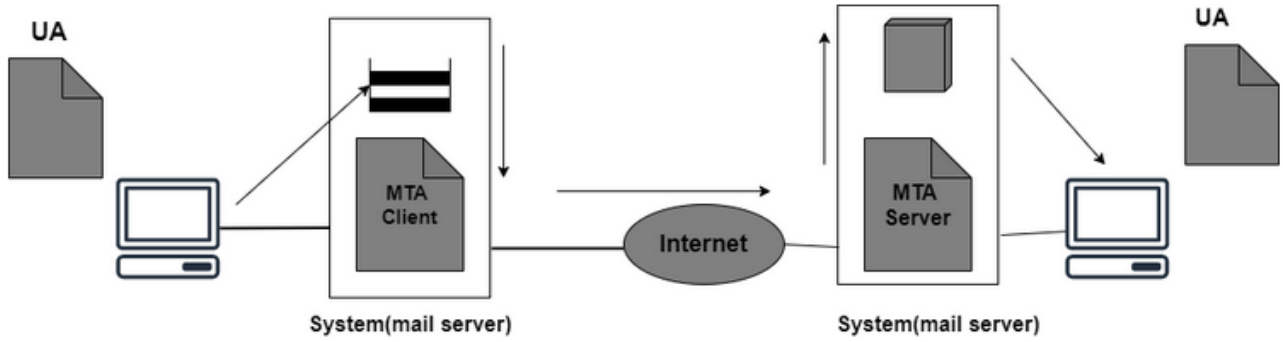
First Scenario

When the sender and the receiver of an E-mail are on the same system, then there is the need for only two user agents.



Second Scenario

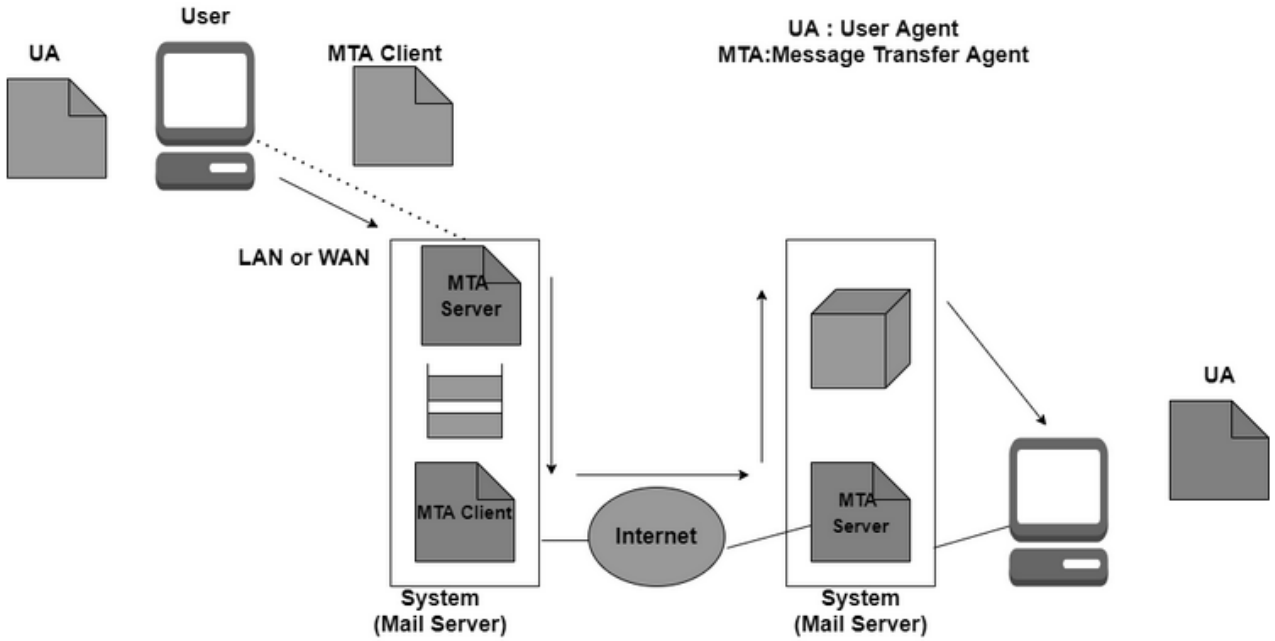
In this scenario, the sender and receiver of an e-mail are basically users on the two different systems. Also, the message needs to send over the Internet. In this case, we need to make use of User Agents and Message transfer agents(MTA).



Third Scenario

In this scenario, the sender is connected to the system via a point-to-point WAN it can be either a dial-up modem or a cable modem. While the receiver is directly connected to the system like it was connected in the second scenario.

Also in this case sender needs a User agent(UA) in order to prepare the message. After preparing the message the sender sends the message via a pair of MTA through LAN or WAN.



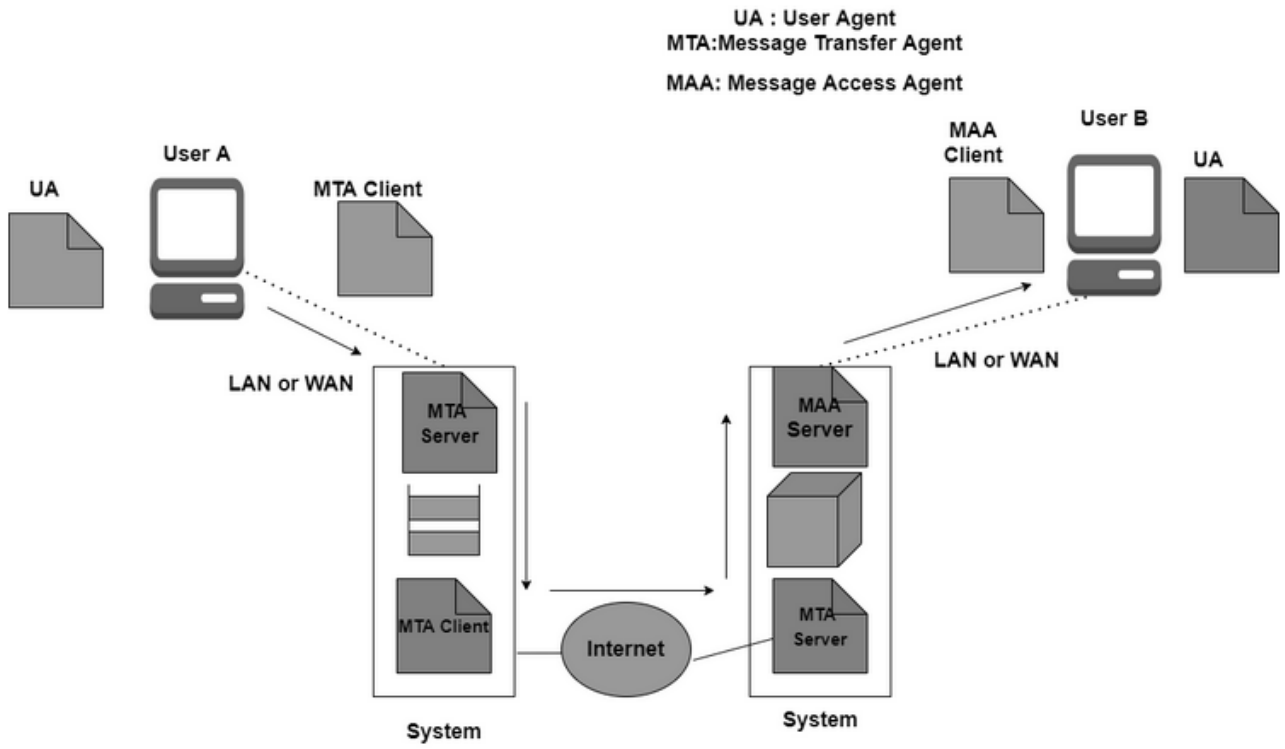
Fourth Scenario

In this scenario, the receiver is also connected to his mail server with the help of WAN or LAN.

When the message arrives the receiver needs to retrieve the message; thus there is a need for another set of client/server agents. The recipient makes use of MAA(Message access agent) client in order to retrieve the message.

In this, the client sends the request to the Mail Access agent(MAA) server and then makes a request for the transfer of messages.

This scenario is most commonly used today.



Web based e-mail

The term Webmail (or Web-based email) is used to describe two things. One use of the word is to describe a Webmail client: an email client implemented as a web application accessed via a web browser.

The other use of the word is to describe a Web-based email service: an email service offered through a web site (a webmail provider) such as Gmail, Yahoo! Mail, Hotmail and AOL Mail. Practically every webmail provider offers email access using a webmail client, and many of them also offer email access by a desktop email client using standard email protocols, while many internet service

providers provide a webmail client as part of the email service included in their internet service package.

As with any web application, webmail's main advantage over the use of a desktop email client is the ability to send and receive email anywhere from a web browser. Its main disadvantage is the need to be connected to the internet while using it (Gmail offers offline use of its webmail client through the installation of Gears.[2]). There exist also other software tools to integrate parts of the webmail functionality into the OS (e.g. creating messages directly from third party applications via MAPI).

Reading and managing mails

When you first login, your inbox is automatically displayed. To view an email, simply click on the subject. The selected email will be highlighted and the entire message will get loaded on a new screen. Bold messages are new or unread. Messages you have already looked at will be in un-bolded text. To view a different folder, just click on the name of the folder you want to view. The options for each folder work the same as your inbox. Let's go through what each of the buttons will do:

- **Check Mail:** This button will download any new messages received. This happens automatically each time you log in, click on the inbox link or explicitly click on this button.
- **Reply:** This is the reply button. It will automatically set up the composition page with the information necessary to reply to the sender of the selected message.
- **Reply All:** This is just like the reply button, but it's used when the selected email is addressed to more the one person and you want to reply to everyone, not just the person who sent the message.
- **Forward:** This will forward the selected message, and direct you to the composition page to enter your recipient.
- **Delete:** No points for guessing. Clicking on this button will delete the selected/current message and move it to the trash folder. If you wish to permanently delete a message you will have to either Empty the trash folder or select the message explicitly and delete it.
- **Compose:** This is the button you would click on to compose, write, or send a new email. When you click on this button, you will be sent to a new page to type out your email, subject, senders etc.

- **Actions:** This button will gives you the option to mark an email or multiple emails as Read, Unread, Flagged or Unflagged.

E-mail Security

E-mail Hacking

Email hacking can be done in any of the following ways:

- Spam
- Virus
- Phishing

Spam

E-mail spamming is an act of sending **Unsolicited Bulk E-mails (UBI)** which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

Virus

Some emails may incorporate with files containing malicious script which when run on your computer may lead to destroy your important data.

Phishing

Email phishing is an activity of sending emails to a user claiming to be a legitimate enterprise. Its main purpose is to steal sensitive information such as usernames, passwords, and credit card details. Such emails contains link to websites that are infected with malware and direct the user to enter details at a fake website whose look and feels are same to legitimate one.

E-mail Spamming and Junk Mails

Email spamming is an act of sending Unsolicited Bulk E-mails (UBI) which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

Spams may cause the following problems:

- It floods your e-mail account with unwanted e-mails, which may result in loss of important e-mails if inbox is full.
- Time and energy is wasted in reviewing and deleting junk emails or spams.
- It consumes the bandwidth that slows the speed with which mails are delivered.
- Some unsolicited email may contain virus that can cause harm to your computer.

Blocking Spams

Following ways will help you to reduce spams:

- While posting letters to newsgroups or mailing list, use a separate e-mail address than the one you used for your personal e-mails.
- Don't give your email address on the websites as it can easily be spammed.
- Avoid replying to emails which you have received from unknown persons.
- Never buy anything in response to a spam that advertises a product.

E-mail Cleanup and Archiving

In order to have light weighted Inbox, it's good to archive your inbox from time to time. Here I will discuss the steps to clean up and archive your Outlook inbox.

- Select **File** tab on the mail pane.
- Select **Cleanup Tools** button on account information screen.
- Select **Archive** from cleanup tools drop down menu.
- Select **Archive this folder and all subfolders** option and then click on the folder that you want to archive. Select the date from the **Archive items older than:** list. Click **Browse** to create new **.pst** file name and location. Click **OK.**

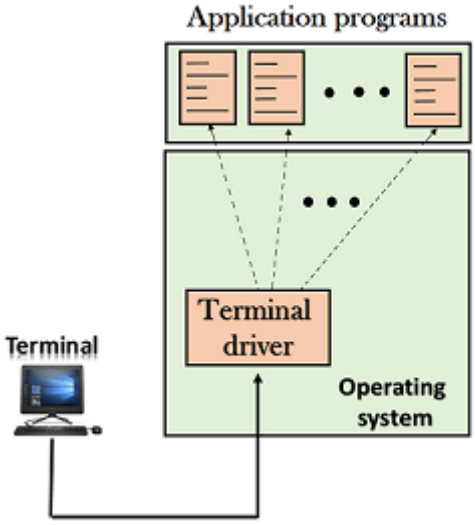
Telnet

- The main task of the internet is to provide services to users. For example, users want to run different application programs at the remote site and transfers a result to the local site. This requires a client-server program such as FTP, SMTP. But this would not allow us to create a specific program for each demand.

- The better solution is to provide a general client-server program that lets the user access any application program on a remote computer. Therefore, a program that allows a user to log on to a remote computer. A popular client-server program Telnet is used to meet such demands. Telnet is an abbreviation for **Terminal Network**.
- Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

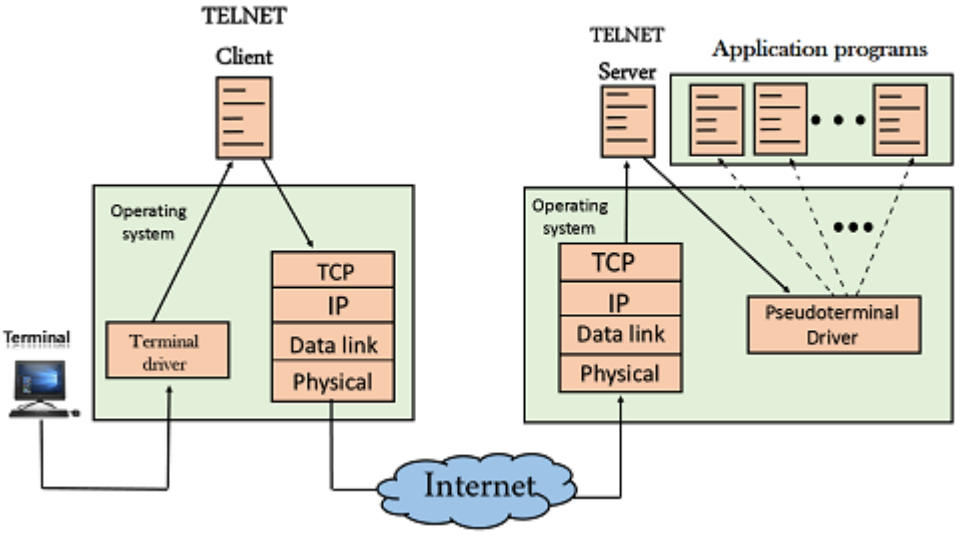
There are two types of login:

Local Login



- When a user logs into a local computer, then it is known as local login.
- When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes these characters to the operating system which in turn, invokes the desired application program.
- However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters have special meanings such as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.

Remote login



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

How remote login occurs

- **At the local site**

  The user sends the keystrokes to the terminal driver, the characters are then sent to the TELNET client. The TELNET client which in turn, transforms the characters to a universal character set known as network virtual terminal characters and delivers them to the local TCP/IP stack

- **At the remote site**

  The commands in NVT forms are transmitted to the TCP/IP at the remote machine. Here, the characters are delivered to the operating system and then pass to the TELNET server. The TELNET server transforms the characters which can be understandable by a remote computer. However, the characters cannot be directly passed to the operating system as a remote operating system does not receive the characters from the TELNET server. Therefore it requires some piece of software that can accept the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

Domain Name System

- Domain Name System is an Internet service that translates domain names into IP addresses.
- The DNS has a distributed database that resides on multiple machines on the Internet.

- DNS has some protocols that allow the client and servers to communicate with each other.
- When the Internet was small, mapping was done by using hosts.txt file.
- The host file was located at host's disk and updated periodically from a master host file.
- When any program or any user wanted to map domain name to an address, the host consulted the host file and found the mapping.
- Now Internet is not small, it is impossible to have only one host file to relate every address with a name and vice versa.
- The solution used today is to divide the host file into smaller parts and store each part on a different computer.
- In this method, the host that needs mapping can call the closest computer holding the needed information.
- This method is used in Domain Name System (DNS).

Name space

- The names assigned to the machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.
- There are two types of name spaces: Flat name spaces and Hierarchical names.

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.

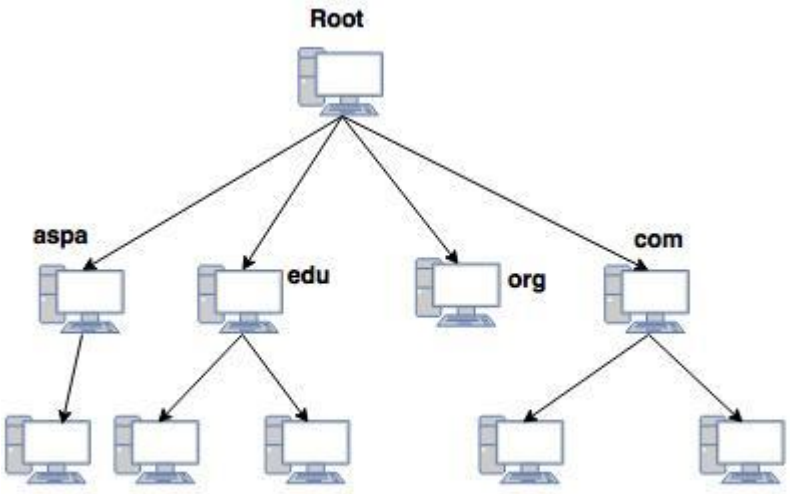


Fig: Hierarchy of DNS

DNS in the Internet

- DNS is a protocol that can be used in different platform.
- Domain Name Space is divided into different sections in the Internet: Generic domain, country domain and inverse domain.

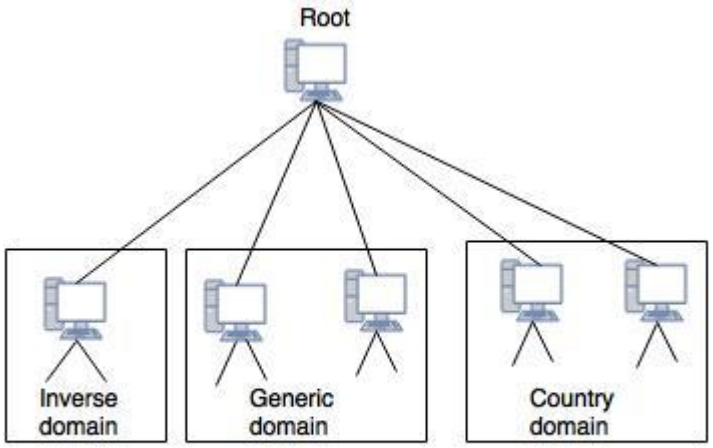


Fig. : DNS in the Internet

Generic Domains

The generic domains define registered hosts according to their generic behavior.

Generic domain labels are as stated below:

1. Country Domains

- Country domain uses two character country abbreviations.
- Second labels can be more specific, national designation.

- **For example,** for Australia the country domain is "au", Inida is .in, UK is .uk etc.

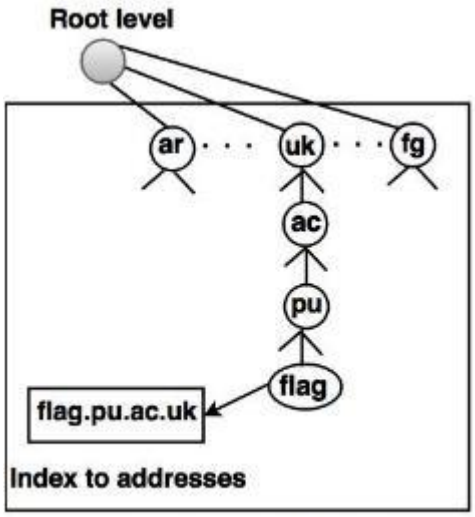


Fig: Country domains

Inverse Domains

- Inverse domain is used to map an address to a name.
- **For example,** a client send a request to the server for performing a particular task, server finds a list of authorized client. The list contains only IP addresses of the client.
- The server sends a query to the DNS server to map an address to a name to determine if the client is on the authorized list.
- This query is called an inverse query.
- This query is handled by first level node called arpa.

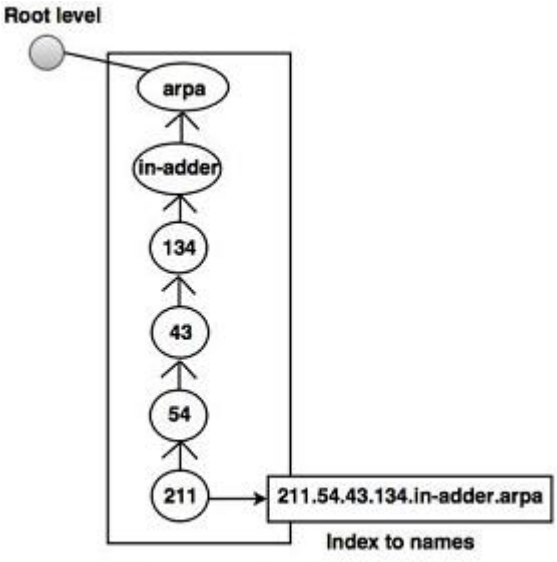


Fig. Inverse domain

DNS cache

The DNS is a great technology that allows us to use the internet the way we know currently. It resolves domain names to their IP addresses, and we get our answers almost instantly. But the DNS resolution is a complicated process that could involve many DNS servers placed far away from each other, and it takes time. There is a way to reduce the DNS queries and save time – DNS cache.

What is DNS cache?

The DNS cache (also known as DNS resolver cache) is a temporary DNS storage on a device (your computer, smartphone, server, etc.) that contains DNS records of already visited domain names (A records for IPv4 addresses, AAAA records for IPv6, etc.). It keeps those records, depending on their time-to-live (TTL).

Each time you visit a website, its addresses will be saved inside this temporary database of records to facilitate a later revisit.

Basically, the DNS cache is how your device is trying to save effort and time and skip a long DNS lookup by answering a DNS query with a DNS record that is already inside the temporary DNS cache.

Why do we need a DNS cache?

We need DNS cache to get a faster response for DNS query for domain names that we have already visited recently in the past.

Both the device, that the user is using (his or her computer) and the multiple DNS resolvers, that the request reaches, have DNS cache and they can resolve the domain if it is still in their cache memory. If not, the DNS query will need to follow the long way to the root server who will direct to the TLD servers and then they will direct to the authoritative name server for the domain name to finally get the answer.

How does it work?

Each time a user performs a DNS lookup, its device will first check inside the internal DNS cache that is part of the OS. There is a table of DNS records inside the DNS cache, their values, and the time they could be kept (TTL). The TTL value is set by the DNS administrator of each domain name, and it is the time limit that each DNS record has. After the time runs out, a new query is required.

If the DNS query can be resolved from the DNS cache, the user will get their answer, and they can visit the site they desired.

If no, the query will travel to a recursive DNS server. There are many DNS recursive servers out there. Like for example, there are inside your Internet Service Provider. They also have a cache that works in the same way. If the answer can be found there, the user will get it and resolve the domain.
If no, the query will travel to an authoritative nameserver to get the answer.

When it gets the answer, the DNS record or records will be saved in each of the DNS caches of the recursive DNS servers on the way and inside the user's device, too, for the period that the TTL value indicates.

Next time a new query starts for the same domain name, your device will repeat the process. If not so much time has passed, there is a high chance that the DNS record your device needs is still inside this temporary memory, and the query gets answered instantly.

DNS Resource Record

We can define DNS Resource Records simply as DNS Server database entries. Resource Records are usually a name to IP Address (IPv4 or IPv6) mapping (or vice versa). DNS Resource Records are used to answer DNS client queries. Resource Records are added to the DNS server for the portion of the DNS namespace which the DNS Server is hosting.

There are different types of Resource Records. Most important types of Resource Records are 1) IPv4 host address (A), 2) IPv6 host address (AAAA, pronounced "quad-A") 3) CNAME (Alias), 4) Pointer (PTR), 5) Mail Exchanger (MX) 6) Service (SRV)

DNS Resource Record Type	**Explanation**
A Record	IPv4 Host Record, used for mapping a Domain Name to an IPv4 address
AAAA Record (pronounced "quad-A")	IPv6 Host Record, used for mapping a Domain Name to an IPv6 address
CNAME Record (Canonical Names)	Alias Record, used for mapping an alias of a DNS domain name. CNAME Record are useful to use more than one name to a single host. CNAME Records allow using different names for same host.
MX Record	Mail Exchanger, used for mapping a DNS domain name to the mail server. MX (Mail Exchanger) Records are used by e-mail applications to locate mail server for a DNS domain, based on the destination e-mail address. MX (Mail Exchanger) Record stores the mail server information for a particular domain.
PTR Record	Pointer, used for reverse lookup (IP Address to Domain Name resolution)
SRV Record	SRV record, used to map available services. Mainly used by Active Directory in Microsoft Windows Servers

DNS Messages Format

DNS has two types of messages: query and response. Both types have the same format. The query message consists of a header and the question records; the response message consisits of a header, question records, answer records, authoritative records, and additional records.

Header

Both query and response messages have the same header format with some fields set to zero for the query messages. the header is 12 byte and its format is as follows:

Identification	**flags**
Number of question records	Number of answer records(All os in query message
Number of authoritative records(All os in query message)	Number of additional records(All os in query message)

The header fields are as follows:

- **Identification.** 16 bit field used by the client to match the response with the query. The client uses ientification number each time it sends a query. the server duplicates this number in response.
- **Flags.** 16 bit field consisting of other subfields as shown below.
  1. QR (query/response). If set (1) means message is a response , if 0 it means message type is query.
  2. OpCode. 4-bit defines type of query or response (0-standard, 1-inverse, 2-server status required).
  3. AA (authoritative answer). (1-bit and used inly in response message. Set (1)-means Authoritative server).
  4. TC (truncated) . if set means value of 1, means messgae was more than 512 bytes and is truncated.
  5. RD (recursion desired). A 1-bit field, when set means client desires reursive answer. It is repeated in both request and response.
  6. RA (recursion available). 1-bit, and it is set only in response message to indiacate that recursion is available.
  7. Reserved. A 3-bit subfield set to 000.
  8. rCode. A 4-bit field which shows the status of error in the response. Of course, only an authoritative server can make such judgement.

     Values of rCode:

     1. 0 - No error
     2. 1 - Format error.
     3. 2 - Problem at name server.

4. 3 - Domain reference problem.
5. 4 - Quert type not supported.
6. 5 - Administratively prohibited.
7. 6-15 - Reserved

- Number of Question Records. This is a16-bit field consisting of number of queries in question section of message.
- Number of Answer Records. This is a 16-bit field containing the number of answer records in the answer section of response message. Its value is 0in the query message.
- Number of authoritative records. A sixteen bit field which tells the number of authoritative records in the authoritative section of the response message. Its value is zero in the query message.
- Number of additional records. This is a 16 bit fieldcontaining the number of additional recordsin the additional section of the response message.

Question Section

This is a section consisting of one or more question records. It is present on both query and response messsages

Answer Section

This is section consisting of one or more resource records. It is present only in response messages. This section includes answer from the server to the client (resolver).

Authoritative Section

This section is also contained only in response messages of DNS, and gives information about domain names regarding authoritative servers for the query.

Additional Information Section

This section provides additional information to help the resolver and present only in response part of DNS message format.

Security of DNS Name Servers

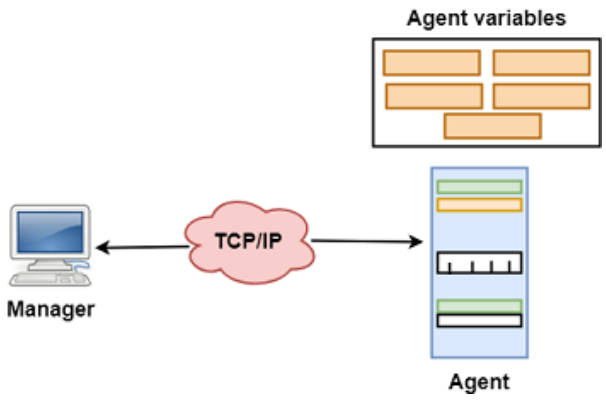
DNS is an old protocol, and it was built without any integrated security. Several solutions have been developed to help secure DNS, including:

- **Reputation Filtering:** Like any other Internet user, most malware needs to make DNS requests to find the IP addresses of the sites that it is visiting. Organizations can block or redirect DNS requests to known malicious domains – based on threat intelligence – to stop users from visiting dangerous sites or malware from communicating with its operator.
- **DNS Inspection:** The use of DNS for data exfiltration (via DNS tunneling) and other malicious activities can be detected and blocked by an intrusion prevention system (IPS) integrated into a next-generation firewall (NGFW). This helps to block the abuse of DNS for malware command and control and other attacks.
- **Secure the Protocol:** DNSSEC is a protocol that includes authentication for DNS responses. Since the authenticated response cannot be spoofed or modified, attackers cannot use DNS to send users to malicious sites.
- **Secure the Channel:** DNS over TLS (DoT) and DoH (DNS over HTTPS) adds a secure layer to an insecure protocol. This ensures that the requests are encrypted and authenticated, unlike traditional DNS. By using DoH and DoT, a user can ensure the privacy of DNS responses and block eavesdropping on their DNS requests (which reveals the sites that they are visiting).

SNMP

- SNMP stands for **Simple Network Management Protocol**.
- SNMP is a framework used for managing devices on the internet.
- It provides a set of operations for monitoring and managing the internet.

SNMP Concept



- SNMP has two components Manager and agent.
- The manager is a host that controls and monitors a set of agents such as routers.
- It is an application layer protocol in which a few manager stations can handle a set of agents.
- The protocol designed at the application level can monitor the devices made by different manufacturers and installed on different physical networks.
- It is used in a heterogeneous network made of different LANs and WANs connected by routers or gateways.

Managers & Agents

- A manager is a host that runs the SNMP client program while the agent is a router that runs the SNMP server program.
- Management of the internet is achieved through simple interaction between a manager and agent.
- The agent is used to keep the information in a database while the manager is used to access the values in the database. For example, a router can store the appropriate variables such as a number of packets received and forwarded while the manager can compare these variables to determine whether the router is congested or not.
- Agents can also contribute to the management process. A server program on the agent checks the environment, if something goes wrong, the agent sends a warning message to the manager.

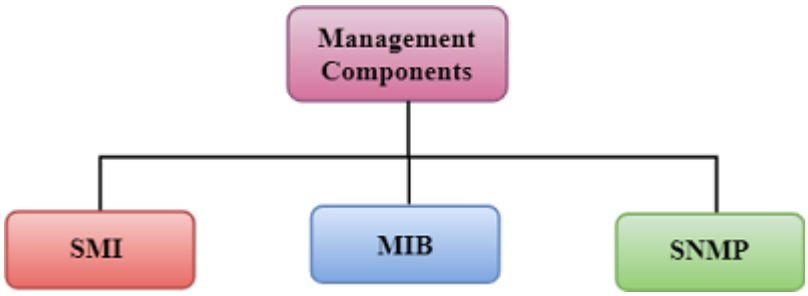
Management with SNMP has three basic ideas:

- A manager checks the agent by requesting the information that reflects the behavior of the agent.
- A manager also forces the agent to perform a certain function by resetting values in the agent database.
- An agent also contributes to the management process by warning the manager regarding an unusual condition.

Management Components

- Management is not achieved only through the SNMP protocol but also the use of other protocols that can cooperate with the SNMP protocol. Management is achieved through the use of the other two protocols: SMI (Structure of management information) and MIB(management information base).

- Management is a combination of SMI, MIB, and SNMP. All these three protocols such as abstract syntax notation 1 (ASN.1) and basic encoding rules (BER).

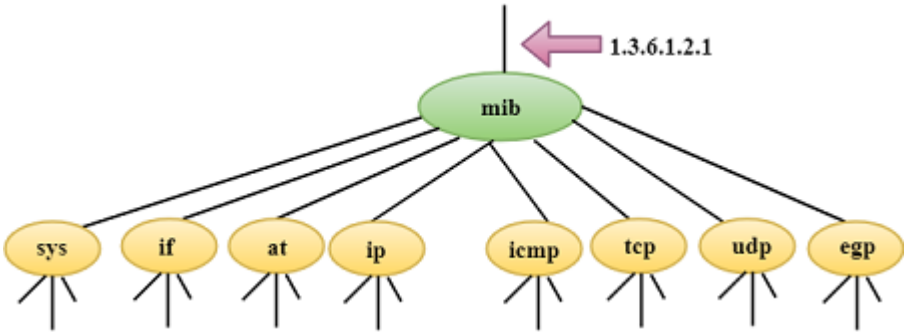


SMI

The SMI (Structure of management information) is a component used in network management. Its main function is to define the type of data that can be stored in an object and to show how to encode the data for the transmission over a network.

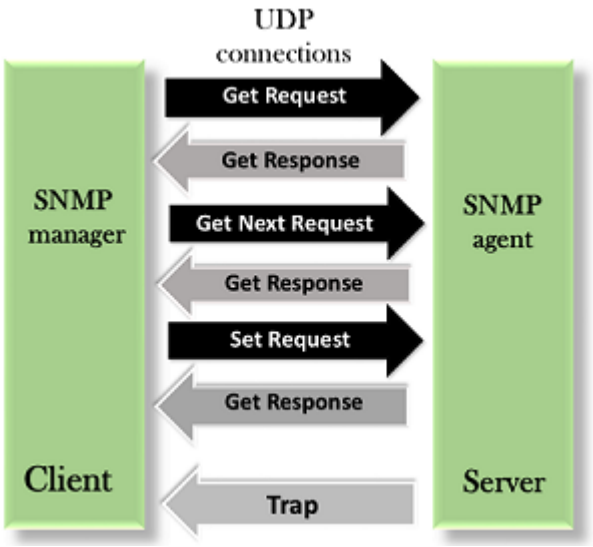
MIB

- The MIB (Management information base) is a second component for the network management.
- Each agent has its own MIB, which is a collection of all the objects that the manager can manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.



SNMP

SNMP defines five types of messages: GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap.



- **GetRequest:** The GetRequest message is sent from a manager (client) to the agent (server) to retrieve the value of a variable.

- **GetNextRequest:** The GetNextRequest message is sent from the manager to agent to retrieve the value of a variable. This type of message is used to retrieve the values of the entries in a table. If the manager does not know the indexes of the entries, then it will not be able to retrieve the values. In such situations, GetNextRequest message is used to define an object.

- **GetResponse:** The GetResponse message is sent from an agent to the manager in response to the GetRequest and GetNextRequest message. This message contains the value of a variable requested by the manager.

- **SetRequest:** The SetRequest message is sent from a manager to the agent to set a value in a variable.

- **Trap:** The Trap message is sent from an agent to the manager to report an event. For example, if the agent is rebooted, then it informs the manager as well as sends the time of rebooting.