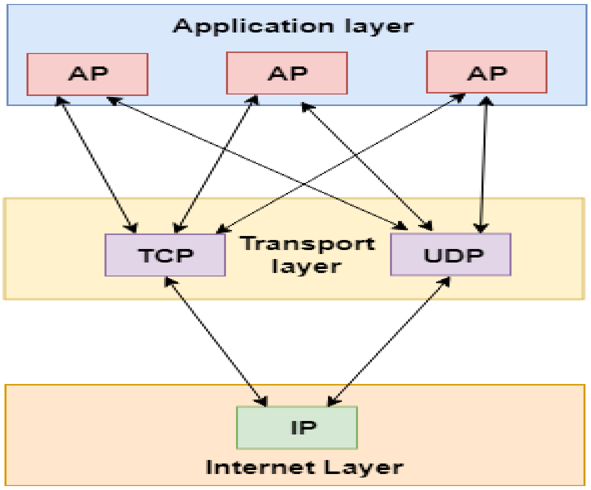
Transport Layer

- 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.

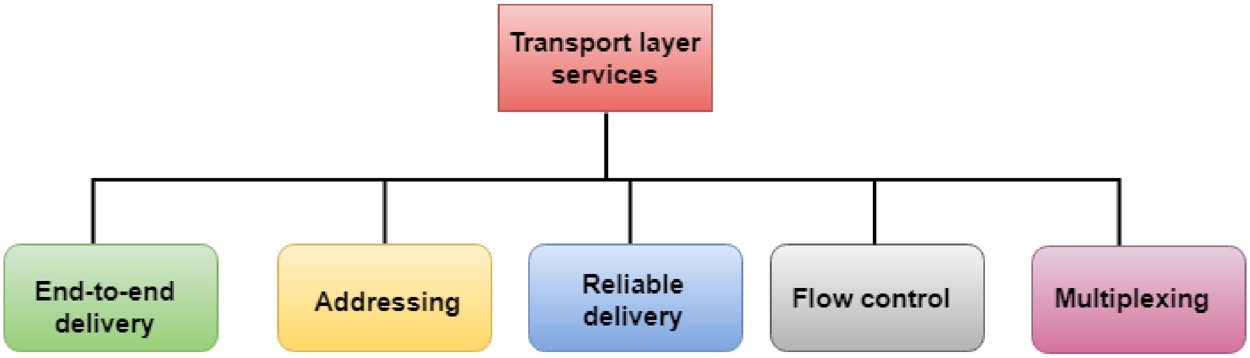


Services provided by the Transport Layer

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:

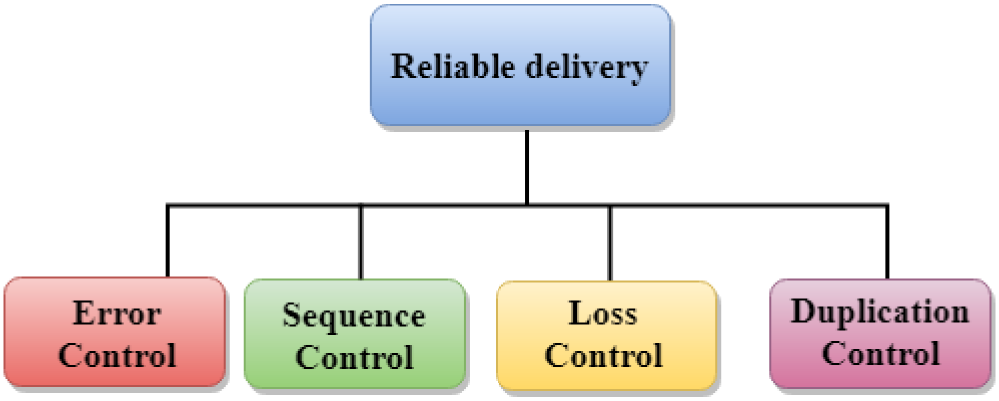
The transport layer transmits the entire message to the destination. Therefore, it ensures the end-to-end delivery of an entire message from a source to the destination.

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

The transport layer uses the multiplexing to improve transmission efficiency.

Multiplexing can occur in two ways:

- **Upward multiplexing:** Upward multiplexing means multiple transport layer connections use the same network connection. To make more cost-effective, the transport layer sends several transmissions bound for the same destination along the same path; this is achieved through upward multiplexing.
- **Downward multiplexing:** Downward multiplexing means one transport layer connection uses the multiple network connections. Downward multiplexing allows the transport layer to split a connection among several paths to improve the throughput. This type of multiplexing is used when networks have a low or slow capacity.

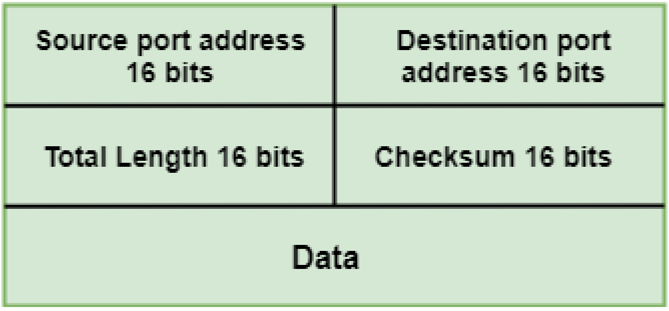
Addressing

- According to the layered model, the transport layer interacts with the functions of the session layer. Many protocols combine session, presentation, and application layer protocols into a single layer known as the application layer. In these cases, delivery to the session layer means the delivery to the application layer. Data generated by an application on one machine must be transmitted to the correct application on another machine. In this case, addressing is provided by the transport layer.

- The transport layer provides the user address which is specified as a station or port. The port variable represents a particular TS user of a specified station known as a Transport Service access point (TSAP). Each station has only one transport entity.
- The transport layer protocols need to know which upper-layer protocols are communicating.

UDP

- UDP stands for **User Datagram Protocol**.
- UDP is a simple protocol and it provides nonsequenced transport functionality.
- UDP is a connectionless protocol.
- This type of protocol is used when reliability and security are less important than speed and size.
- UDP is an end-to-end transport level protocol that adds transport-level addresses, checksum error control, and length information to the data from the upper layer.
- The packet produced by the UDP protocol is known as a user datagram.
- ## User Datagram Format
- The user datagram has a 16-byte header which is shown below:

- 

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

Where,

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

Features of UDP

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.
- It is a suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP(Routing Information Protocol).
- Normally used for real-time applications which can not tolerate uneven delays between sections of a received message.

Advantages of UDP

- It does not require any connection for sending or receiving data.
- Broadcast and Multicast are available in UDP.
- UDP can operate on a large range of networks.
- UDP has live and real-time data.
- UDP can deliver data if all the components of the data are not complete.

Disadvantages of UDP

- We can not have any way to acknowledge the successful transfer of data.
- UDP cannot have the mechanism to track the sequence of data.
- UDP is connectionless, and due to this, it is unreliable to transfer data.
- In case of a Collision, UDP packets are dropped by Routers in comparison to TCP.
- UDP can drop packets in case of detection of errors.

TCP

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

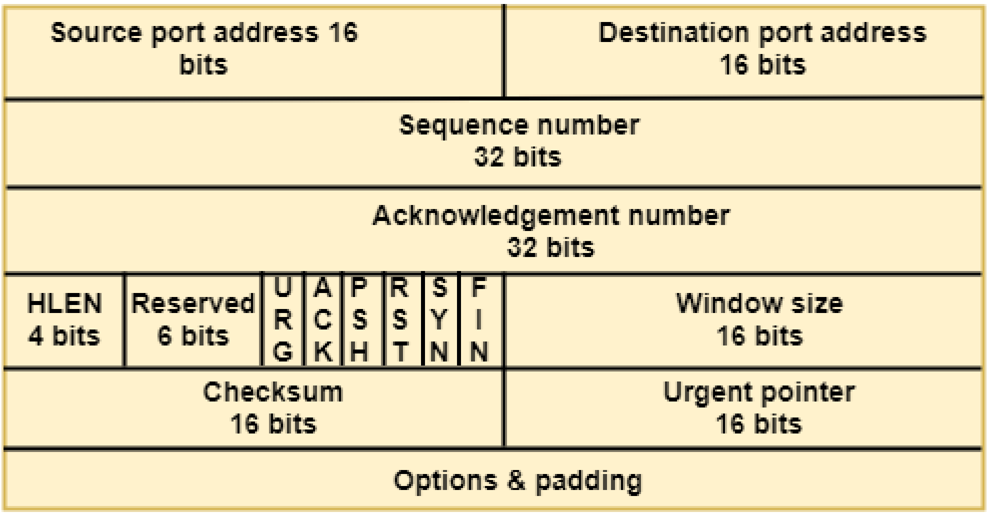
Features Of TCP protocol

- **Stream data transfer:** TCP protocol transfers the data in the form of contiguous stream of bytes. TCP group the bytes in the form of TCP segments and then passed it to the IP layer for transmission to the destination. TCP itself segments the data and forward to the IP.
- **Reliability:** TCP assigns a sequence number to each byte transmitted and expects a positive acknowledgement from the receiving TCP. If ACK is not received within a timeout interval, then the data is retransmitted to the destination.
  The receiving TCP uses the sequence number to reassemble the segments if they arrive out of order or to eliminate the duplicate segments.
- **Flow Control:** When receiving TCP sends an acknowledgement back to the sender indicating the number the bytes it can receive without overflowing its internal buffer. The number of bytes is sent in ACK in the form of the highest sequence number that it can receive without any problem. This mechanism is also referred to as a window mechanism.
- **Multiplexing:** Multiplexing is a process of accepting the data from different applications and forwarding to the different applications on different computers. At the receiving end, the data is forwarded to the correct application. This process is known as demultiplexing. TCP transmits the packet to the correct application by using the logical channels known as ports.
- **Logical Connections:** The combination of sockets, sequence numbers, and window sizes, is called a logical connection. Each connection is identified by the pair of sockets used by sending and receiving processes.
- **Full Duplex:** TCP provides Full Duplex service, i.e., the data flow in both the directions at the same time. To achieve Full Duplex service, each TCP should have sending and receiving buffers so that the segments can flow in both the directions. TCP is a connection-oriented protocol.

Suppose the process A wants to send and receive the data from process B. The following steps occur:

- Establish a connection between two TCPs.
- Data is exchanged in both the directions.
- The Connection is terminated.

TCP Segment Format



Source port address 16 bits

Destination port address 16 bits

Sequence number 32 bits

Acknowledgement number 32 bits

HLEN 4 bits

Reserved 6 bits

URG

ACK

PSH

RST

SYN

FIN

Window size 16 bits

Checksum 16 bits

Urgent pointer 16 bits

Options & padding

Where,

- **Source port address:** It is used to define the address of the application program in a source computer. It is a 16-bit field.
- **Destination port address:** It is used to define the address of the application program in a destination computer. It is a 16-bit field.
- **Sequence number:** A stream of data is divided into two or more TCP segments. The 32-bit sequence number field represents the position of the data in an original data stream.
- **Acknowledgement number:** A 32-field acknowledgement number acknowledge the data from other communicating devices. If ACK field is set to 1, then it specifies the sequence number that the receiver is expecting to receive.

- **Header Length (HLEN):** It specifies the size of the TCP header in 32-bit words. The minimum size of the header is 5 words, and the maximum size of the header is 15 words. Therefore, the maximum size of the TCP header is 60 bytes, and the minimum size of the TCP header is 20 bytes.
- **Reserved:** It is a six-bit field which is reserved for future use.
- **Control bits:** Each bit of a control field functions individually and independently. A control bit defines the use of a segment or serves as a validity check for other fields.

There are total six types of flags in control field:

- **URG:** The URG field indicates that the data in a segment is urgent.
- **ACK:** When ACK field is set, then it validates the acknowledgement number.
- **PSH:** The PSH field is used to inform the sender that higher throughput is needed so if possible, data must be pushed with higher throughput.
- **RST:** The reset bit is used to reset the TCP connection when there is any confusion occurs in the sequence numbers.
- **SYN:** The SYN field is used to synchronize the sequence numbers in three types of segments: connection request, connection confirmation (with the ACK bit set), and confirmation acknowledgement.
- **FIN:** The FIN field is used to inform the receiving TCP module that the sender has finished sending data. It is used in connection termination in three types of segments: termination request, termination confirmation, and acknowledgement of termination confirmation.
    - **Window Size:** The window is a 16-bit field that defines the size of the window.
    - **Checksum:** The checksum is a 16-bit field used in error detection.

- **Urgent pointer:** If URG flag is set to 1, then this 16-bit field is an offset from the sequence number indicating that it is a last urgent data byte.
- **Options and padding:** It defines the optional fields that convey the additional information to the receiver.

Features of TCP

- TCP keeps track of the segments being transmitted or received by assigning numbers to every single one of them.
- Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery.
- TCP implements an error control mechanism for reliable data transfer.
- TCP takes into account the level of congestion in the network.

Advantages of TCP

- It is reliable for maintaining a connection between Sender and Receiver.
- It is responsible for sending data in a particular sequence.
- Its operations are not dependent on OS.
- It allows and supports many routing protocols.
- It can reduce the speed of data based on the speed of the receiver.

Disadvantages of TCP

- It is slower than UDP and it takes more bandwidth.
- Slower upon starting of transfer of a file.
- Not suitable for LAN and PAN Networks.
- It does not have a multicast or broadcast category.
- It does not load the whole page if a single data of the page is missing.

Where TCP is Used?

- Sending Emails

- Transferring Files
- Web Browsing

Where UDP is Used?

- Gaming
- Video Streaming
- Online Video Chats

- ## Differences between TCP and UDP
- The main differences between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are:

Basis	**Transmission Control Protocol (TCP)**	**User Datagram Protocol (UDP)**
Type of Service	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, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	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.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.

Basis	**Transmission Control Protocol (TCP)**	**User Datagram Protocol (UDP)**
Sequence	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.
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	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).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It’s a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn’t support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.

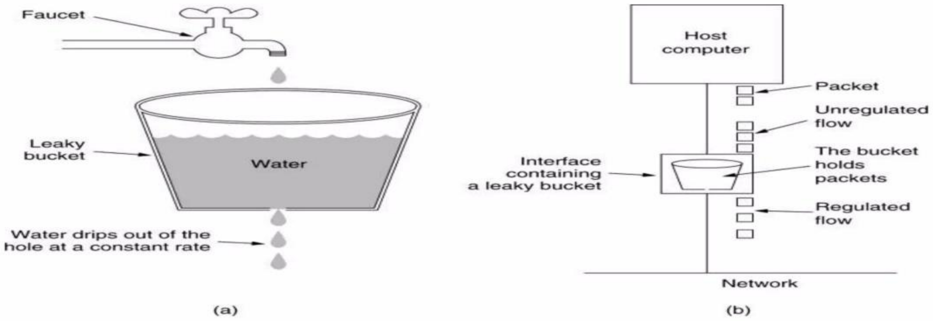
Traffic Shaping

- It is about regulating average rate of data flow.
- It is a method of congestion control by providing shape to data flow before entering the packet into the network.
- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape)
- There are two types of Traffic shaping algorithm :-
  - 1. **Leaky Bucket Algorithm.**
  - 2. **Token Bucket Algorithm**

Leaky Bucket Algorithm

- The **Leaky Bucket Algorithm** used to control rate in a network.
- It is implemented as a single-server queue with constant service time.
- If the bucket (buffer) overflows then packets are discarded.
- In this algorithm the input rate can vary but the output rate remains constant.
- This algorithm saves busty traffic into fixed rate traffic by averaging the data rate.

The Leaky Bucket Algorithm



(a) A leaky bucket with water.

(b) a leaky bucket with packets.

Leaky Bucket Algorithm

Algorithm

Step - 1 : Initialize the counter to **‘n’** at every tick of clock.

Step - 2 : If **n** is greater than the size of packet in the front of queue send the packet into the network and decrement the counter by size of packet. Repeat the step until **n** is less than the size of packet.

Step - 3 : Reset the counter and go to Step - 1.

Example

Let **n** = 1000

Packet =.

200	700	500	450	400	200

Since n > front of Queue i.e. n>200

Therefore, n= 1000-200 = 800

Packet size of 200 is sent to the network

200	700	500	450	400

Now Again n > front of queue i.e. n > 400

Therefore, n= 800-400 = 400

Packet size of 400 is sent to the network

200	700	500	450

Since n < front of queue .

There fore, the procedure is stop.

And we initialize **n = 1000** on another tick of clock.

This procedure is repeated until all the packets is sent to the network.

Leaky Bucket Algorithm

Example

Consider a frame relay network having a capacity of 1Mb and data is input at the rate of 25mbps.Calculate

1. What is the time needed to fill the bucket.
2. If the output rate is 2 mbps , the time needed to empty the bucket.

Ans.

Here ,

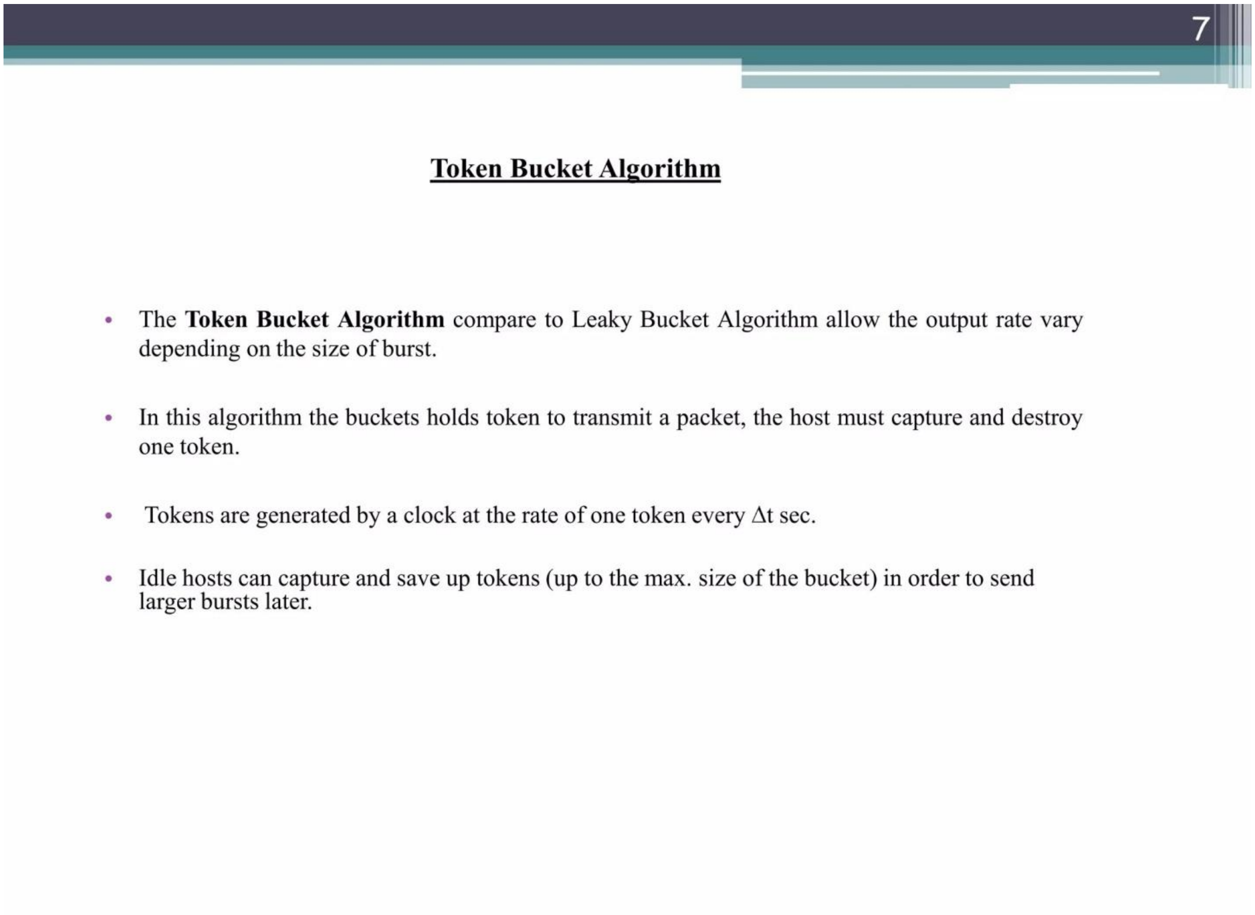
C is Capacity of bucket = 1mb

Data input rate = 25 mbps

output rate = 2mbps.

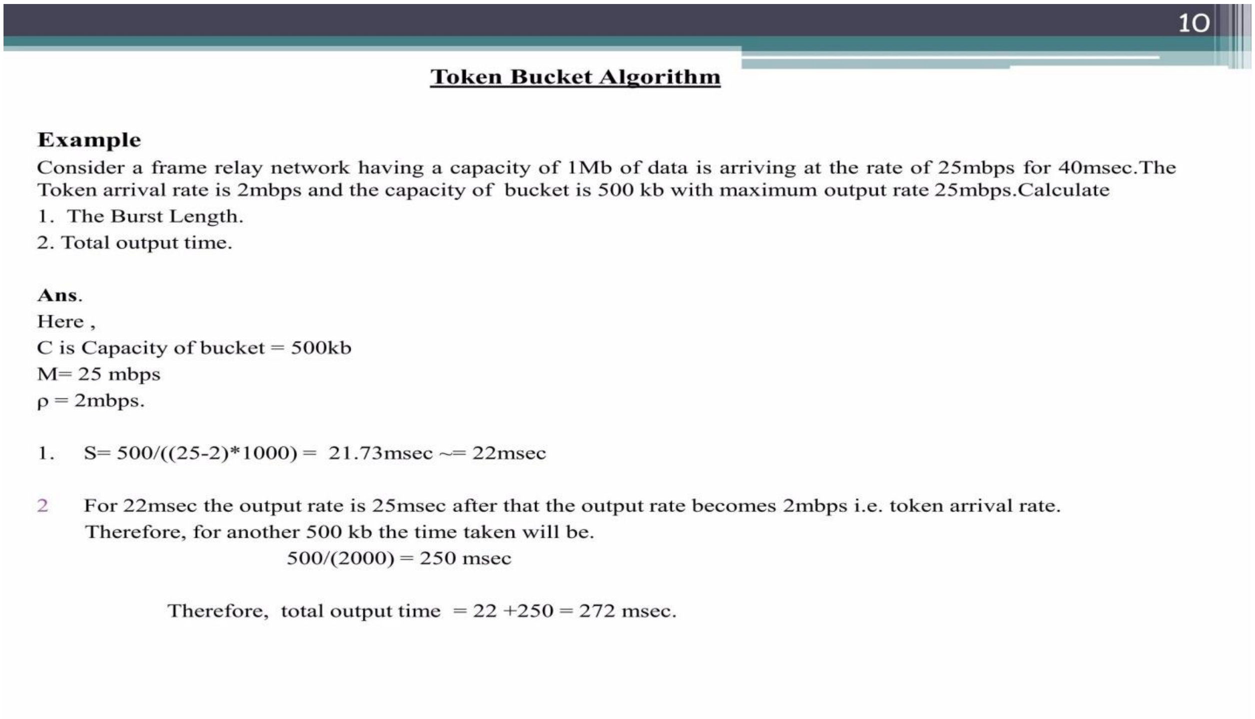
1. T = C/input rate = 1/25 = 40 msec

2 T = C/output rate = ½ = 500 msec



Token Bucket Algorithm

- The **Token Bucket Algorithm** compare to Leaky Bucket Algorithm allow the output rate vary depending on the size of burst.
- In this algorithm the buckets holds token to transmit a packet, the host must capture and destroy one token.
- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



Token Bucket Algorithm

Example

Consider a frame relay network having a capacity of 1Mb of data is arriving at the rate of 25mbps for 40msec.The Token arrival rate is 2mbps and the capacity of bucket is 500 kb with maximum output rate 25mbps.Calculate

1. The Burst Length.
2. Total output time.

Ans.

Here ,

C is Capacity of bucket = 500kb

M= 25 mbps

ρ = 2mbps.

1. S= 500/((25-2)*1000) = 21.73msec ~= 22msec

2 For 22msec the output rate is 25msec after that the output rate becomes 2mbps i.e. token arrival rate. Therefore, for another 500 kb the time taken will be.

500/(2000) = 250 msec

Therefore, total output time = 22 +250 = 272 msec.

Token Bucket Algorithm

Algorithm

Step - 1 : A token is added at every Δt time.

Step - 2 : The bucket can hold at most **b-**tokens. If a token arrive when bucket is full it is discarded.

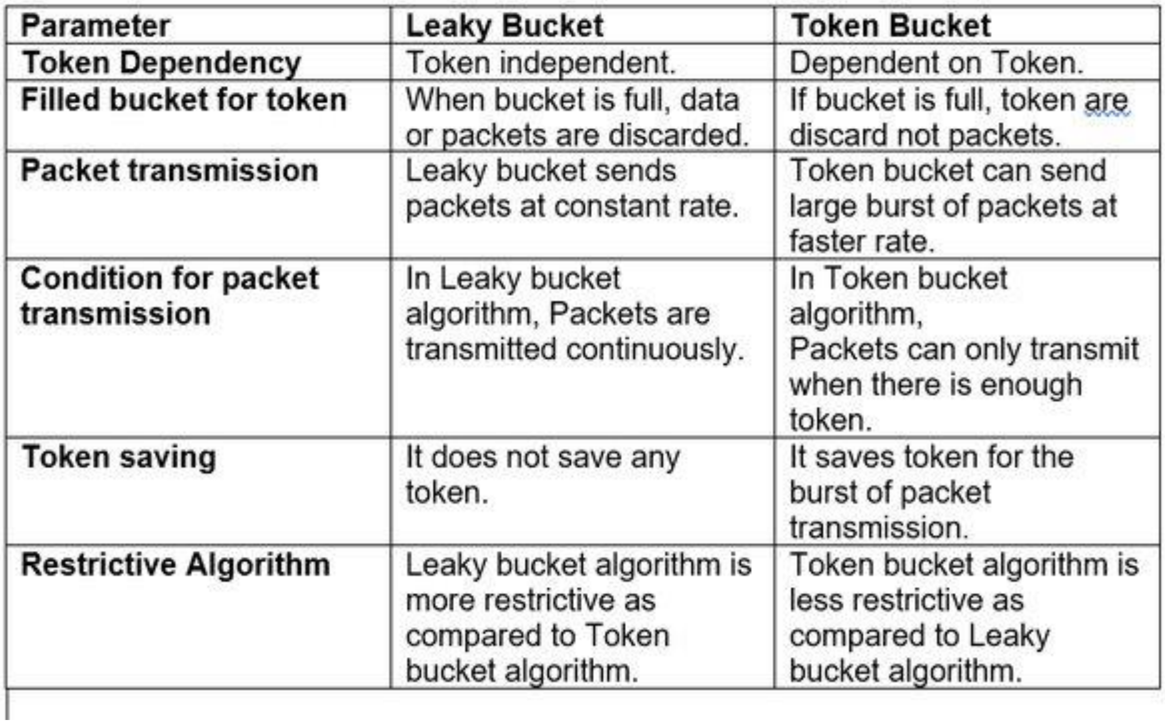
Step - 3 : When a packet of **m** bytes arrived **m** tokens are removed from the bucket and the packet is sent to the network.

Step – 4 : If less than **n** tokens are available no tokens are removed from the buckets and the packet is considered to be **non conformant**.

The **non conformant** packet may be enqueued for subsequent transmission when sufficient token have been accumulated in the bucket.

If **C** is the maximum capacity of bucket and **ρ** is the arrival rate and **M** is the maximum output rate then Burst Length **S** can be calculated as

$$\boxed{C + \rho S = MS}$$



Parameter	Leaky Bucket	Token Bucket
Token Dependency	Token independent.	Dependent on Token.
Filled bucket for token	When bucket is full, data or packets are discarded.	If bucket is full, token are discard not packets.
Packet transmission	Leaky bucket sends packets at constant rate.	Token bucket can send large burst of packets at faster rate.
Condition for packet transmission	In Leaky bucket algorithm, Packets are transmitted continuously.	In Token bucket algorithm, Packets can only transmit when there is enough token.
Token saving	It does not save any token.	It saves token for the burst of packet transmission.
Restrictive Algorithm	Leaky bucket algorithm is more restrictive as compared to Token bucket algorithm.	Token bucket algorithm is less restrictive as compared to Leaky bucket algorithm.

Quality-of-Service (QoS) Quality of service (QoS) is the description or measurement of the overall performance of a service, such as a telephony or computer network, or a cloud computing service, particularly the performance seen by the users of the network.

OR

It refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates. Basic phenomenon for QoS means in terms of packet delay and losses of various kinds.

What is quality of service?

Quality of service (QoS) refers to any technology that manages data traffic to reduce packet loss, latency and jitter on a network. QoS controls and manages network resources by setting priorities for specific types of data on the network.

Enterprise networks need to provide predictable and measurable services as applications -- such as voice, video and delay-sensitive data -- to traverse a network. Organizations use QoS to meet the traffic requirements of sensitive applications, such as real-time voice and video, and to prevent the degradation of quality caused by packet loss, delay and jitter.

Organizations can reach a QoS by using certain tools and techniques, such as jitter buffer and traffic shaping. For many organizations, QoS is included in the service-level agreement (SLA) with their network service provider to guarantee a certain level of network performance.

QoS parameters

Organizations can measure QoS quantitatively by using several parameters, including the following:

- **Packet loss.** This happens when network links become congested, and routers and switches start dropping packets. When packets are dropped during real-time communication, such as in voice or video calls, these sessions can experience jitter and gaps in speech. Packets can be dropped when a queue, or line of packets waiting to be sent, overflows.
- **Jitter.** This is the result of network congestion, timing drift and route changes. Too much jitter can degrade the quality of voice and video communication.
- **Latency.** This the time it takes a packet to travel from its source to its destination. Latency should be as close to zero as possible. If a voice over IP call has a high amount of latency, users can experience echo and overlapping audio.
- **Bandwidth.** This is the capacity of a network communications link to transmit the maximum amount of data from one point to another

in a given amount of time. QoS optimizes the network performance by managing bandwidth and giving high priority applications with stricter performance requirements more resources than others.

- **Mean opinion score (MOS).** This is a metric to rate voice quality that uses a five-point scale, with a five indicating the highest quality.

Need for QoS –

- Video and audio conferencing require bounded delay and loss rate.
- Video and audio streaming requires bounded packet loss rate, it may not be so sensitive to delay.
- Time-critical applications (real-time control) in which bounded delay is considered to be an important factor.
- Valuable applications should be provided better services than less valuable applications.

QoS Specification –

QoS requirements can be specified as:

1. Delay
2. Delay Variation(Jitter)
3. Throughput
4. Error Rate

There are two types of QoS Solutions:

1. **Stateless Solutions –**
   Routers maintain no fine-grained state about traffic, one positive factor of it is that it is scalable and robust. But it has weak services as there is no guarantee about the kind of delay or performance in a particular application which we have to encounter.
2. **Stateful Solutions –**
   Routers maintain a per-flow state as flow is very important in providing the Quality-of-Service i.e. providing powerful services such as guaranteed services and high resource utilization, providing protection, and is much less scalable and robust.