CN: UNIT-4 Transport Layer

Transport Layer

- 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/de-multiplexing** 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.

Transport Layer

- To allow users to access the transport service, the transport layer ***must provide some operations to application programs***, that is, a transport service interface. Each transport service has its own interface.
- The transport service is similar to the network service, but there are also some important differences.
- The main difference is that the network service is ***intended to model the service*** offered by real networks. Real networks can lose packets, so the ***network service is generally unreliable***. The (connection-oriented) ***transport service, in contrast, is reliable***.

- As an example, consider two processes connected by pipes in UNIX. They assume the connection between them is perfect. They ***do not want to know about acknowledgements, lost packets, congestion,*** or anything like that. What they want is a 100 percent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other.
- A second difference between the network service and transport service is whom the services are intended for. The network
  - service is used only by the transport entities. Consequently, the transport service must be convenient and easy to use.

Service Primitives

LISTEN

- To start with, the server executes a LISTEN primitive, typically by calling a library procedure that makes a system call that blocks the server until a client turns up.

CONNECT

- When a client wants to talk to the server, it executes a CONNECT primitive. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. The client's CONNECT call causes a CONNECTION REQUEST segment to be sent to the server.

SEND & RECEIVE

- Data can now be exchanged using the SEND and RECEIVE primitives. In the simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND.

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

Service Primitives

DISCONNECT

- When a connection is no longer needed, it must be released to free up table space within the two transport entities. Disconnection has two variants: asymmetric and symmetric.
- In the asymmetric variant, either transport user can issue a DISCONNECT primitive, which results in a DISCONNECT segment being sent to the remote transport entity.
- In the symmetric variant, each direction is closed separately, independently of the other one. When one side does a DISCONNECT, that means it has no more data to send but it is still willing to accept data from its partner.

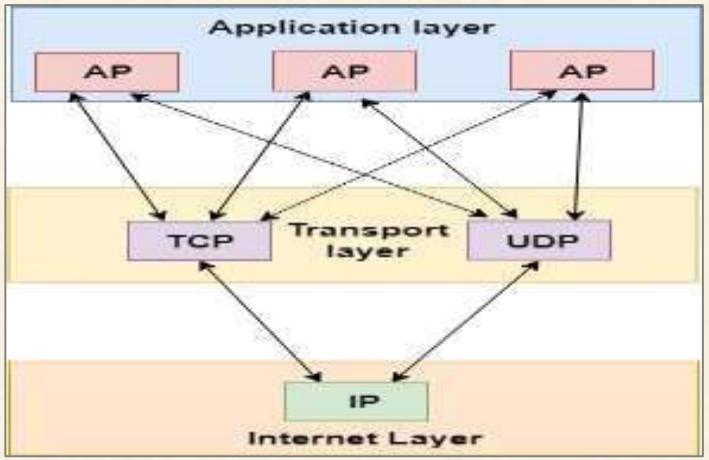
Services Provided

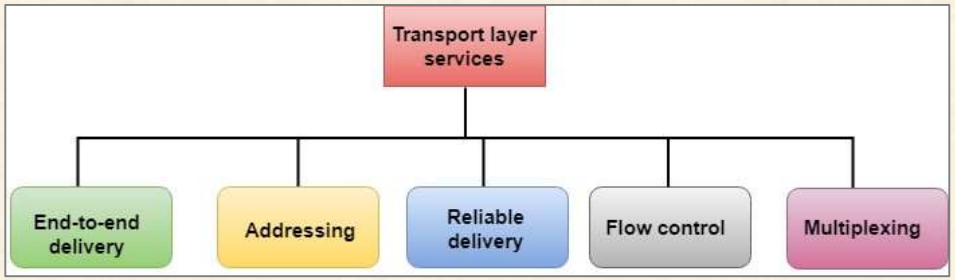
Services provided by the Transport Layer

- 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:

1. End-to-end delivery
2. Addressing
3. Reliable delivery
4. Flow control
5. Multiplexing





End-End & Reliable

End-to-end delivery

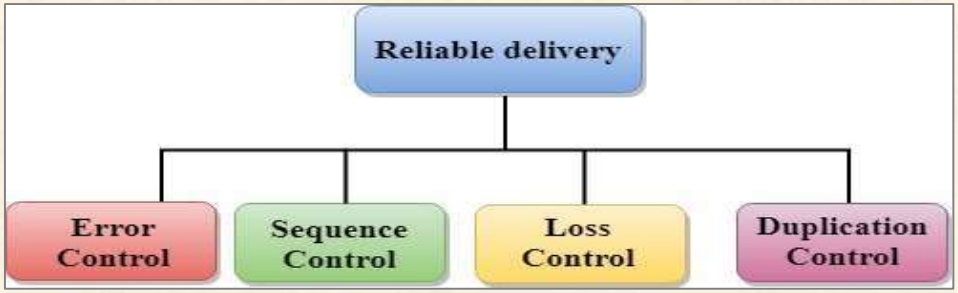
- The transport layer transmits the entire message to the destination. Therefore, it ensures the ***end-to-end delivery of an entire message*** from a source to the destination.

Reliable delivery

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

The reliable delivery has four aspects:

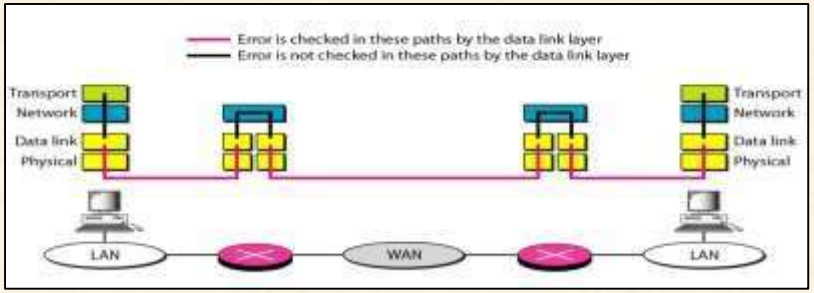
1. Error control
2. Sequence control
3. Loss control
4. Duplication control



Error Control

Error Control

- The primary role of reliability is **Error Control**. In reality, 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.



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.***

Flow Control

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

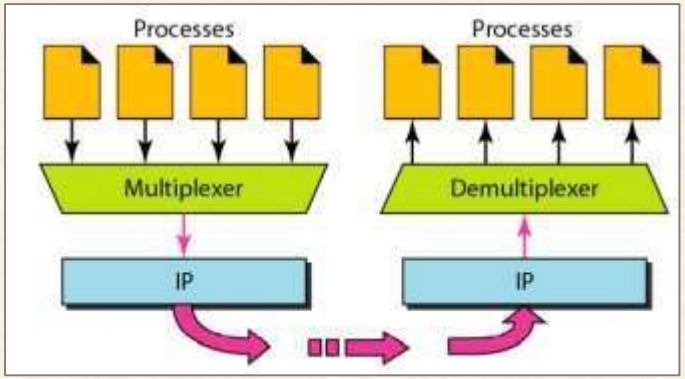
Multiplexing

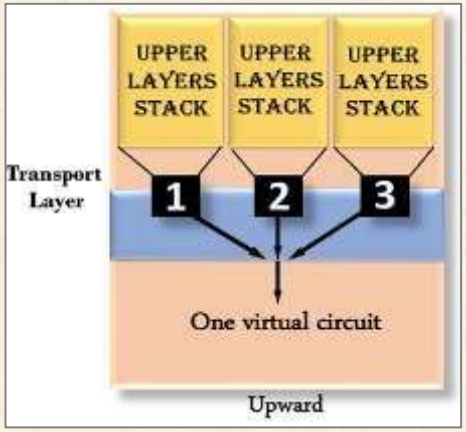
- The transport layer uses the multiplexing to improve transmission efficiency.

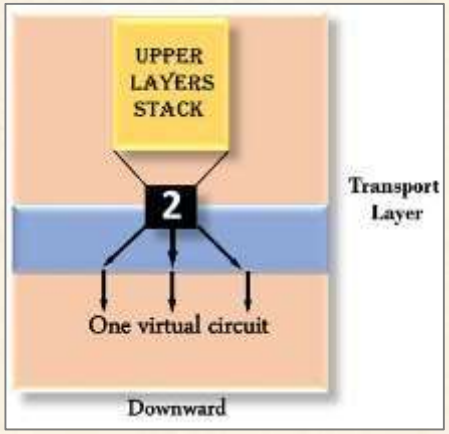
Multiplexing can occur in two ways:

- **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.***



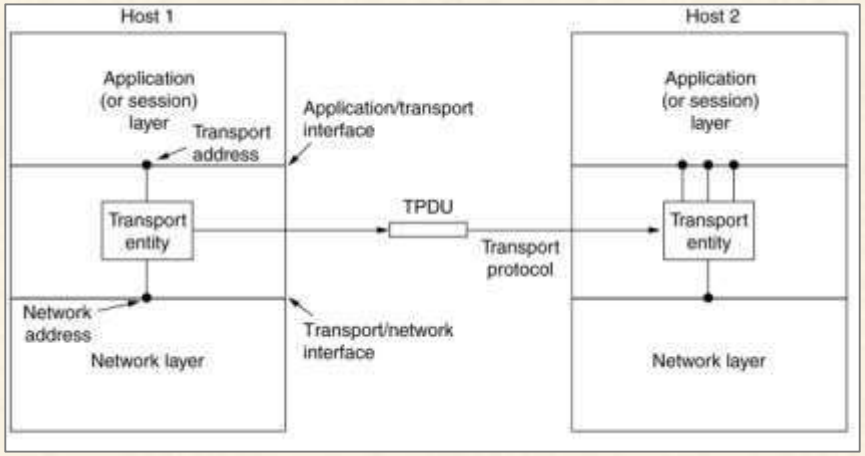




Addressing

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.

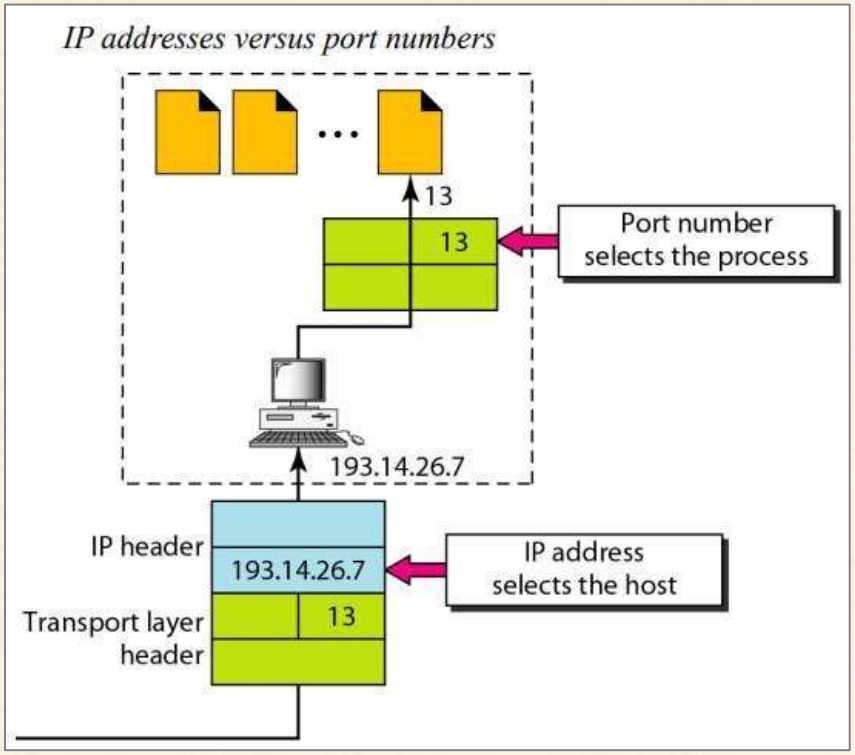


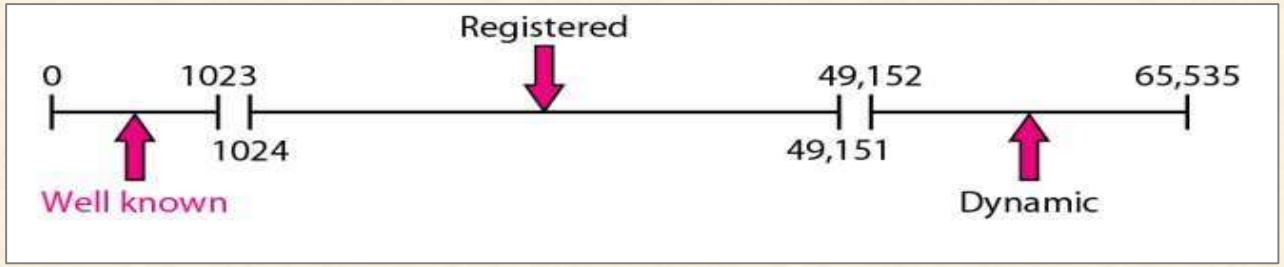
System Ports (0 – 1023)

User Ports (1024 – 49151)

Private/Dynamic Ports (49152 – 65535)

Addressing


IP addresses versus port numbers
13
13
Port number
selects the process
193.14.26.7
IP header
193.14.26.7
IP address
selects the host
Transport layer
header
13


Registered
0
1023
1024
49,152
49,151
65,535
Well known
Dynamic

Some Assigned Ports

Port Number	Process Name	Protocol Used	Description
20	FTP-DATA	TCP	File transfer---data
21	FTP	TCP	File transfer---control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP & UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP & UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol 3
123	NTP	TCP	Network Time Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP

Transport Layer Protocols

Elements of Transport Protocols

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

TCP Connection

TCP Connection

- TCP is connection-oriented. a connection-oriented transport protocol establishes a logical path between the source and destination. In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

Connection Establishment -

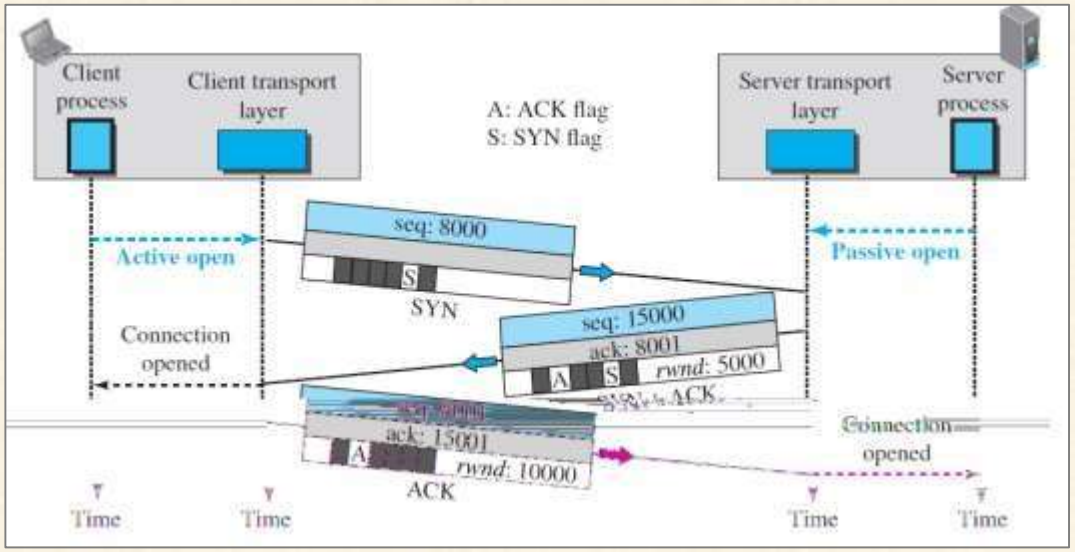
- TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.

Three-Way Handshaking:

- The connection establishment in TCP is called three-way handshaking. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a passive open. Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.
- The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP to connect to a particular server.TCP can now start the three-way handshaking process

TCP Connection

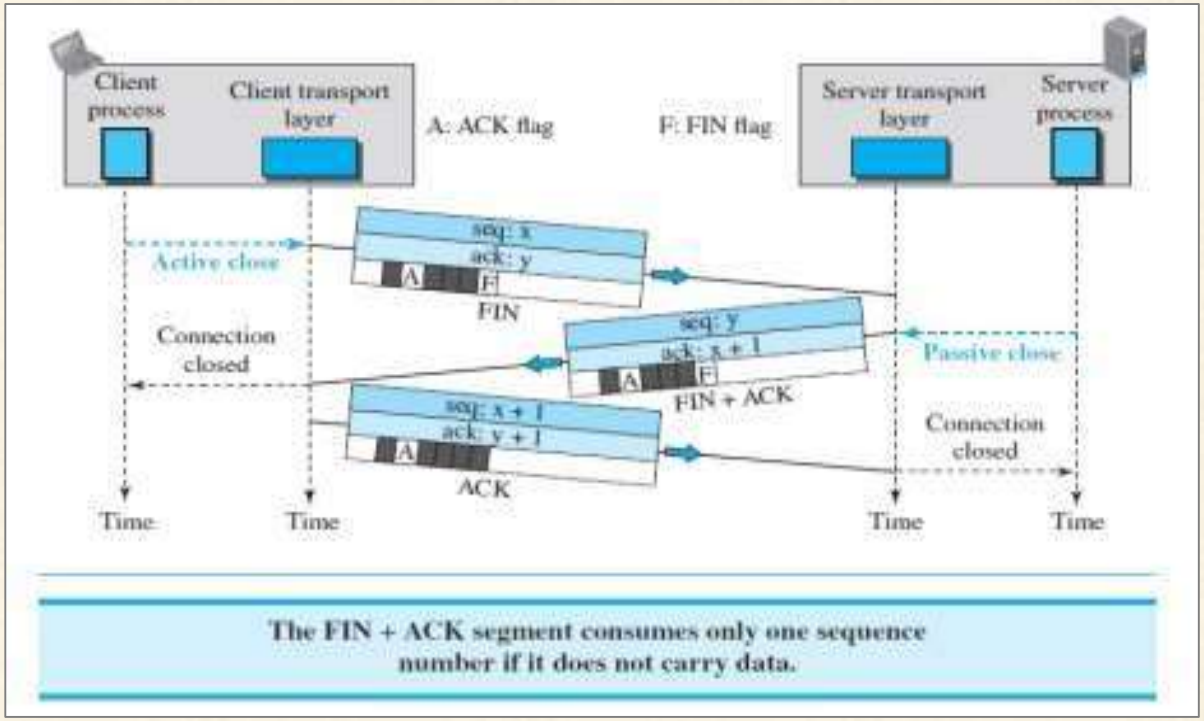
1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN). A SYN segment cannot carry data, but it consumes one sequence number.
2. The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. A SYN + ACK segment cannot carry data, but it does consume one sequence number.
3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. An ACK segment, if carrying no data, consumes no sequence number.



Connection Termination

Connection Termination

Using Three-Way Handshaking



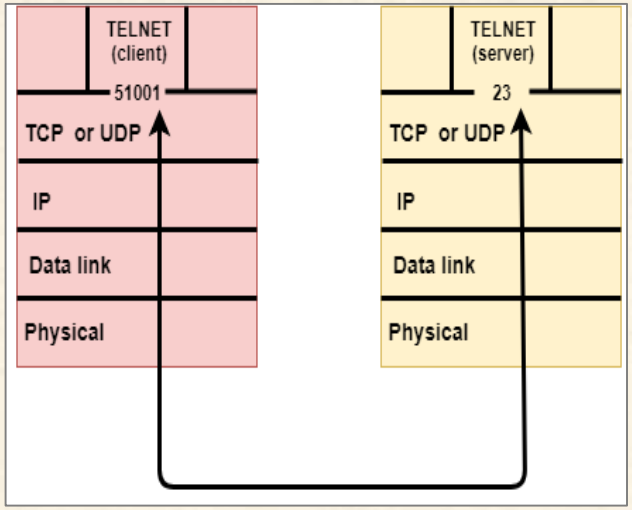
TCP Connection

SYN Flooding Attack:

- The connection establishment procedure in TCP is susceptible to a serious security problem called SYN flooding attack. This happens when one or more malicious attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses in the datagrams.
- The server, assuming that the clients are issuing an active open, allocates the necessary resources, such as creating transfer control block (TCB) tables and setting timers.
- TCP server then sends the SYN + ACK segments to the fake clients, which are lost. When the server waits for the third leg of the handshaking process, however, resources are allocated without being used. If, during this short period of time, the number of SYN segments is large, the server eventually runs out of resources and may be unable to accept connection requests from valid clients.
- This SYN flooding attack belongs to a group of security attacks known as a denial of service attack, in which an attacker
  - monopolizes a system with so many service requests that the system overloads and denies service to valid requests.

Transport Layer Protocols

- The transport layer is represented by ***two protocols:TCP and UDP.***
- The IP protocol in the network layer delivers a datagram from a source host to the destination host.
- Nowadays, the operating system supports multiuser and multiprocessing environments, an executing program is called a process. When a host sends a message to other host means that source process is sending a process to a destination process. The transport layer protocols define some connections to individual ports known as protocol ports.
- An IP protocol is a host-to-host protocol used to deliver a packet from source host to the destination host while transport layer protocols are port-to-port protocols that work on the top of the IP protocols to deliver the packet from the originating port to the IP services, and from IP services to the destination port.
- ***Each port is defined by a positive integer address, and it is of 16 bits.***



UDP

- UDP stands for **User Datagram Protocol**, is a connectionless protocol.
- UDP is a simple protocol and it provides non-sequenced transport functionality.
- 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.

To be Disscussed:

1. Well-Known Ports for UDP
2. User Datagram
3. Checksum
4. UDP Services

UDP: Well-Known Ports

1. **Well-Known Ports for UDP**

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

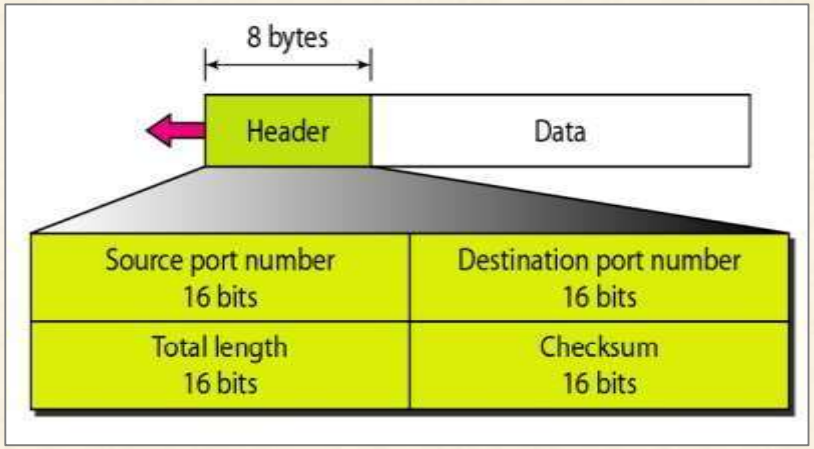
UDP: Datagram

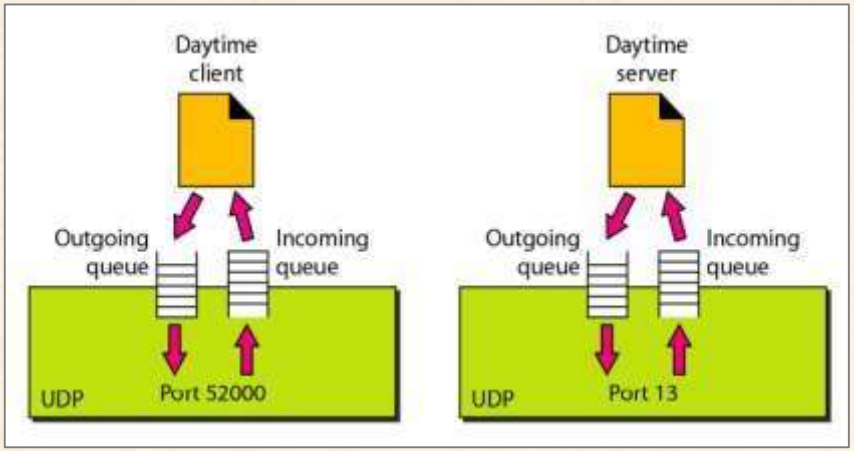
2. User Datagram

The user datagram has a 16-byte header which is shown below:

Where,

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

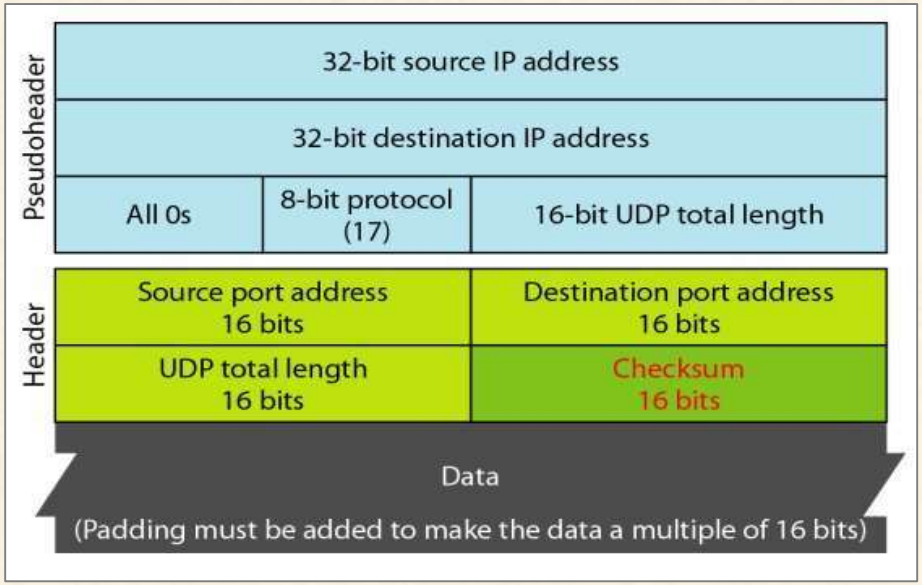




UDP: Checksum

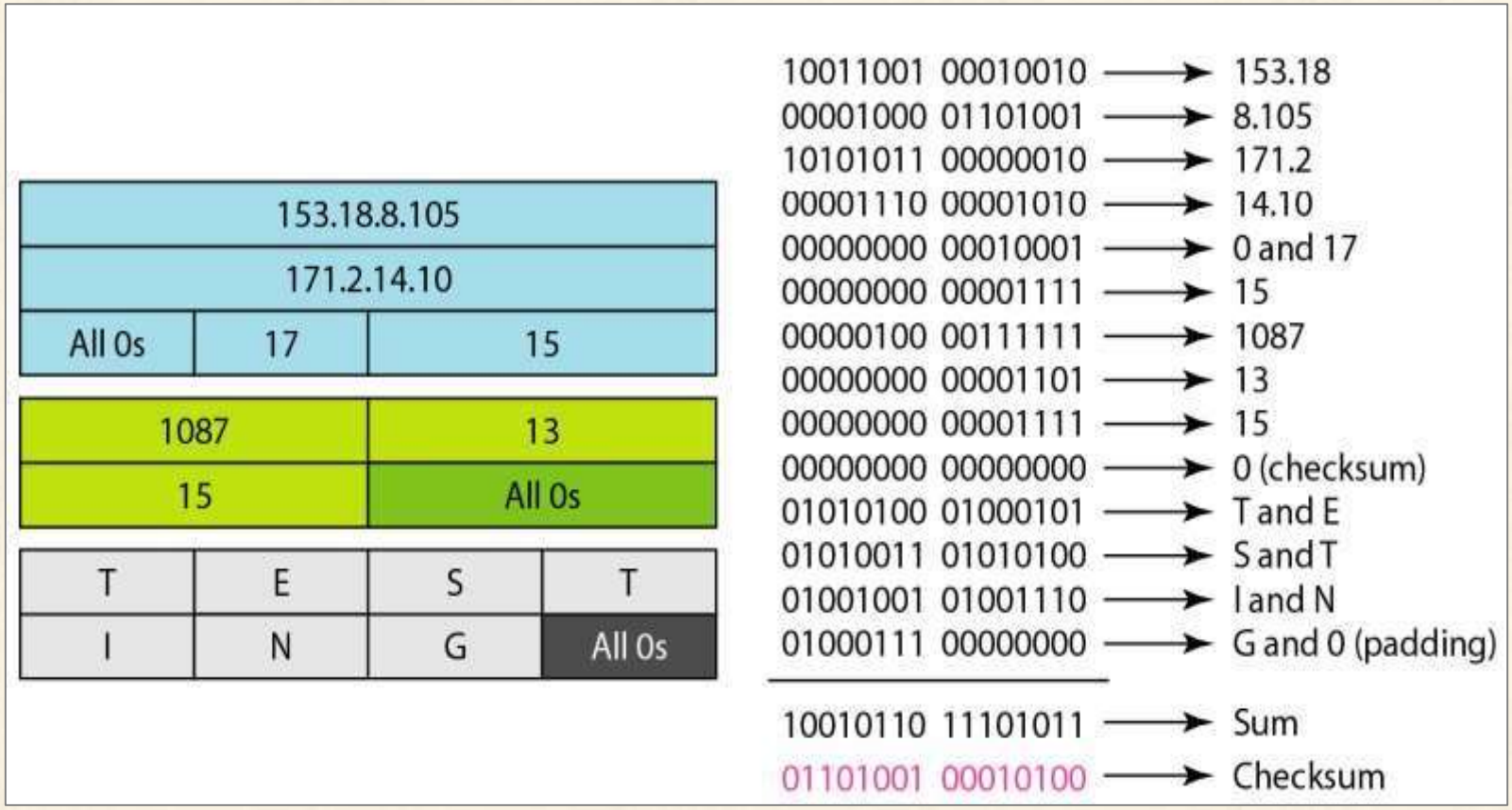
3. C hecksum

- The UDP checksum calculation is different from the one for IP and ICMP.
- Here the checksum includes three sections: ***a pseudo header, the UDP header, and the data*** coming from the application layer. The pseudo header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with O's.
- If the checksum does not include the pseudo header, a user datagram may arrive safe and sound.
- However, if the IP header is corrupted, it may be delivered to the wrong host.
- The protocol field is added to ensure that the packet belongs to UDP, and not to other transport-layer protocols.



UDP: Checksum

Checksum calculation of a simple UDP user datagram



UDP Services

UDP Services

Process-to-Process Communication -

- UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.

Connectionless Services -

- UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- The user datagrams are not numbered. Also, unlike TCP, there is no connection establishment and no connection termination. This means that each user datagram can travel on a different path.

Flow Control -

- UDP is a very simple protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages.The lack of flow control means that the process using UDP should provide for this service, if needed.

UDP Services

Error Control –

- There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded.The lack of error control means that the process using UDP should provide for this service, if needed.

Congestion Control –

- Since UDP is a connectionless protocol, it does not provide congestion control. UDP assumes that the packets sent are small and sporadic and cannot create congestion in the network.

Encapsulation and Decapsulation –

- To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages

UDP

Disadvantages of UDP protocol

- UDP provides basic functions needed for the end-to-end delivery of a transmission.
- It does not provide ***any sequencing or reordering functions*** and does not specify the damaged packet when reporting an error.
- UDP can discover that an error has occurred, but it ***does not specify which packet has been lost*** as it does not contain an ID or sequencing number of a particular data segment.

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 circui***t between sender and receiver for the duration of a transmission.
- This is a connection-oriented protocol that provides a ***reliable, full-duplex byte stream*** to its end users.
- TCP is an example of a stream socket that provides ***a bidirectional, reliable, and sequenced flow of data***, unlike UDP which is a datagram socket.

What makes TCP protocol reliable:

TCP has four important feature which makes it reliable:

1. Error control and
2. Flow control
3. Congestion control
4. Connection management

Error control is achieved by:

1. Acknowledgement number
2. Re transmission
3. Checksum
4. Sequence number

TCP

Acknowledgment Number:

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

For example:

- If the acknowledgment number is 1635, means all the bytes before this number are reached and the receiver expects bytes with 1635 as the next sequence number.
- If an acknowledgment number is not received, TCP automatically re-transmits the data(segment) and waits a longer amount of time.

Note:

- The maximum time it can keep trying re transmission is 4 to 10 mins, depending upon implementation.
- TCP does not guarantee that data will be received at other end, its just that it provides reliable delivery of data or reliable notification of failure.
- There is nothing called segment number in TCP, rather each segment is a collection of bytes and each bytes is associated with sequence number.

TCP

Retransmission:

- This is the heart of the TCP when it comes to reliability that is error control mechanism. If the packets is ***lost or damaged or corrupted or the ack itself is lost,*** TCP retransmits the data.

Retransmission takes place in two scenarios:

- ***Re transmission timer expires:*** that is it does not get the ack for the send bytes ***within stipulated time***.
- ***Fast re transmission:*** This happens when the sender receives three duplicate ACK, in this segment is re transmitted even before RTO. (Retransmission Time Out)

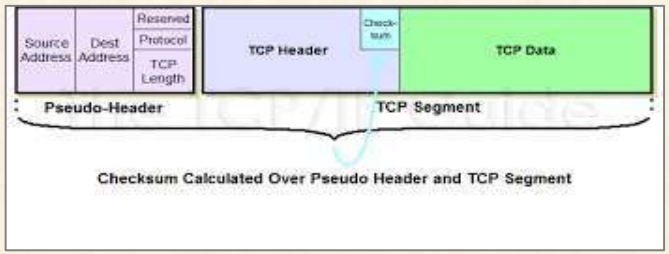
Checksum: This is one of the features of TCP along with acknowledgment and retransmission which is used for error control mechanisms in TCP.

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

  TCP header TCP Body Pseudo IP header

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

1. Source IP address
2. Destination IP address
3. Protocol
4. TCP segment size
5. Fixed of 8 bits



TCP

Sequence number:

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

Significance of sequence number:

Reassembly of packet at receiver side:

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

Discard of duplicate data:

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

Retransmission of lost or corrupted or for damaged data:

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

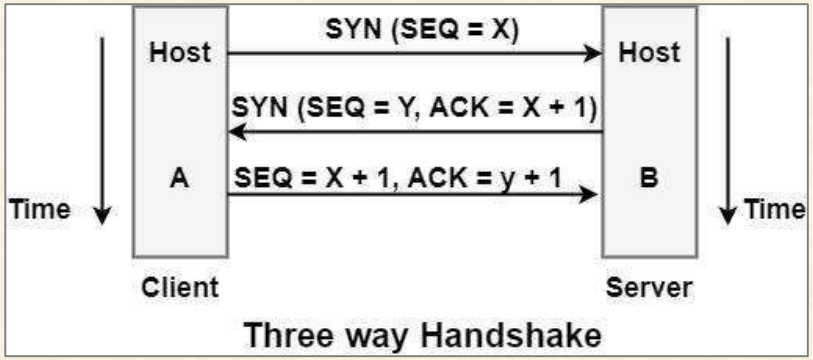
Flow Control:

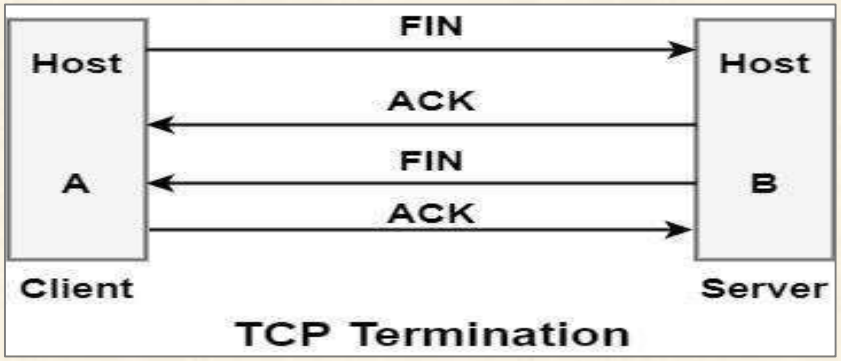
- TCP provides a mechanism called flow control by which it always tells its peer how many bytes of data it is willing to accept. This is called advertised window which reflects the buffer size of the receiver side so that sender cannot overflow the receiver buffer.
- This is also called **windowing mechanism.**

TCP Connection & Termination

The following steps occur:

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





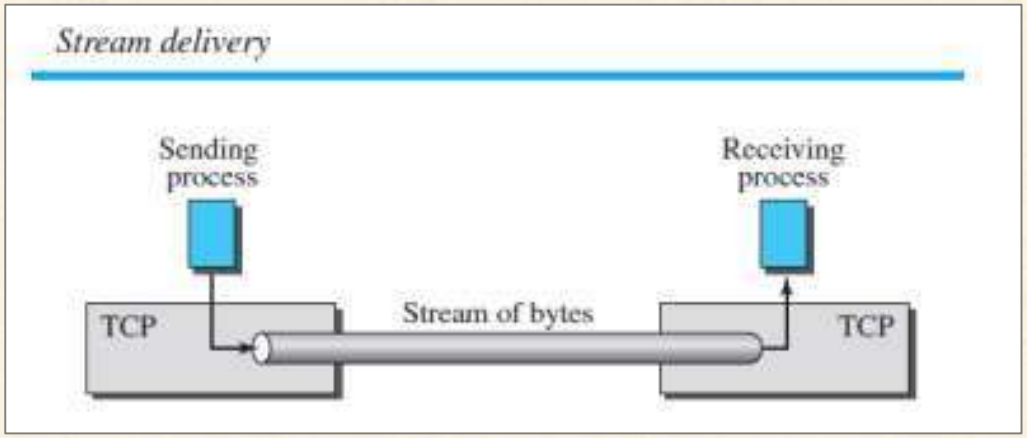
TCP Services

Process-to-Process Communication -

- As with UDP, TCP provides process-to-process communication using port numbers.

Stream Delivery Service -

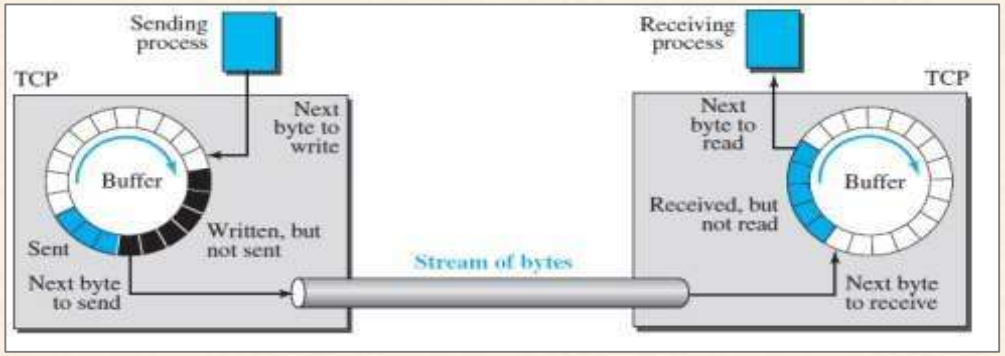
- TCP, unlike UDP, is a stream-oriented protocol. TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet. The sending process produces (writes to) the stream and the receiving process consumes (reads from) it.



TCP Services

Sending and Receiving Buffers –

- Because the sending and the receiving processes may not necessarily write or read data at the same rate, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. These buffers are also necessary for flow- and error control mechanisms used by TCP. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure:

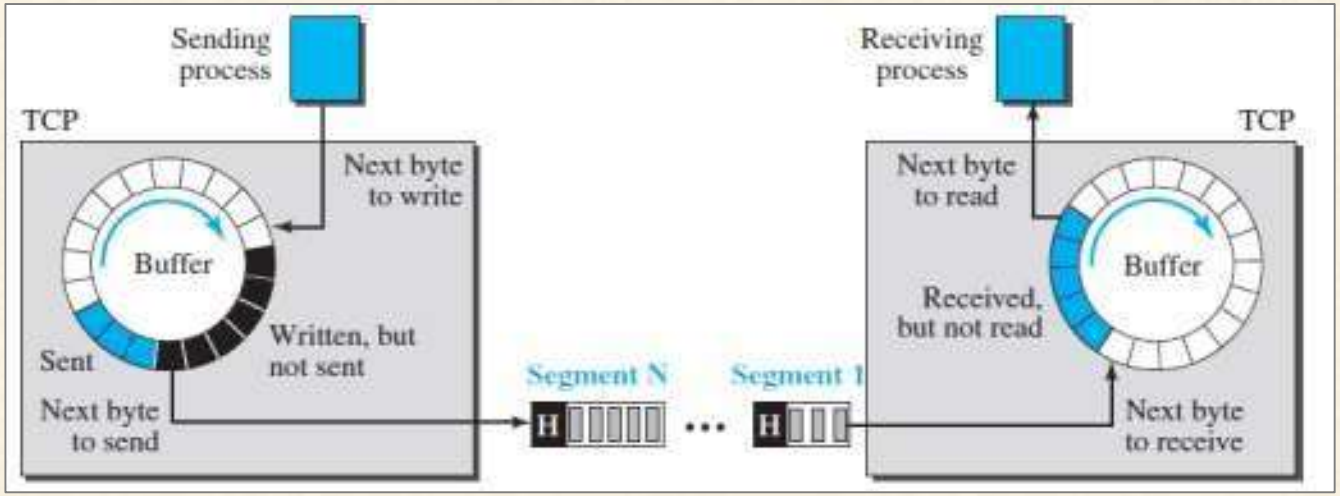


- At the sender, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The colored area holds bytes that have been sent but not yet acknowledged. The TCP sender keeps these bytes in the buffer until it receives an acknowledgment. The shaded area contains bytes to be sent by the sending TCP.
- The operation of the buffer at the receiver is simpler. The circular buffer is divided into two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

TCP Services

Segments -

- The network layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes. At the transport layer, TCP groups a number of bytes together into a packet called a segment.
- TCP adds a header to each segment (for control purposes) and delivers the segment to the network layer for transmission. The segments are encapsulated in an IP datagram and transmitted.
- This entire operation is transparent to the receiving process.



TCP Services

Full-Duplex Communication -

- TCP offers full-duplex service, where data can flow in both directions at the same time Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing -

- Like UDP, TCP performs multiplexing at the sender and demultiplexing at the receiver.

Connection-Oriented Service -

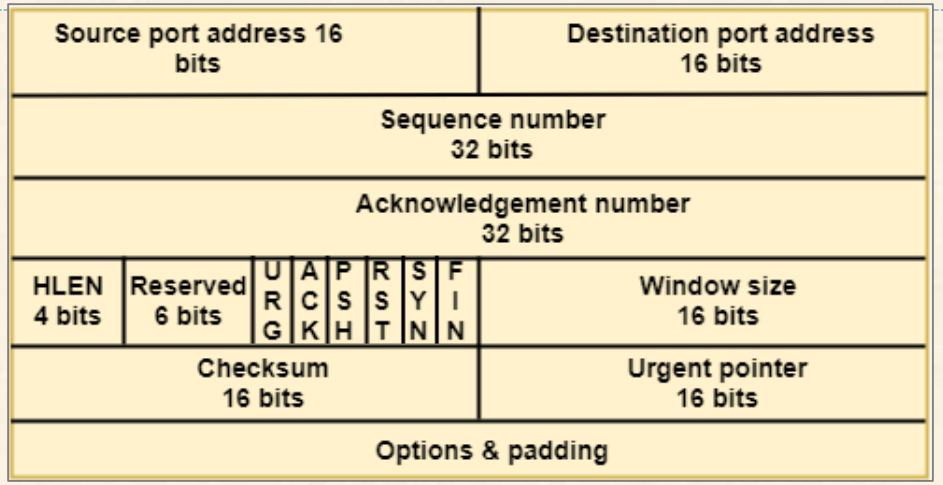
- TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1. The two TCP's establish a logical connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Reliable Service -

- TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data

TCP Segment



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.***

TCP Segment

- **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:

1. **URG:** The URG field indicates that the data in a ***segment is urgent***.
2. **ACK:** When ACK field is set, then it ***validates the acknowledgement number***.
3. **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.
4. **RST:** The reset bit is used to reset the TCP connection when there is ***any confusion occurs*** in the sequence numbers.
5. **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***.
6. **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***.

TCP Vs UDP

Basis for Comparison	TCP	UDP
Definition	TCP establishes a virtual circuit before transmitting the data.	UDP transmits the data directly to the destination computer without verifying whether the receiver is ready to receive or not.
Connection Type	It is a Connection-Oriented protocol	It is a Connectionless protocol
Speed	slow	high
Reliability	It is a reliable protocol.	It is an unreliable protocol.
Header size	20 bytes	8 bytes
acknowledgement	It waits for the acknowledgement of data and has the ability to resend the lost packets.	It neither takes the acknowledgement, nor it retransmits the damaged frame.

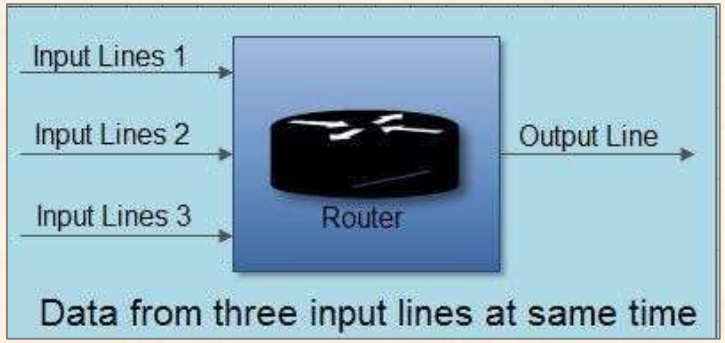
Congestion

- Congestion is an important issue that can arise in ***packet switched network***.
- Congestion is a situation in Communication Networks in which ***too many packets are present in a part of the subnet***, performance degrades.
- Congestion in a network may occur ***when the load on the network*** (i.e. the number of packets sent to the network) is greater than the capacity of the network (i.e. the number of packets a network can handle.).
- Network congestion occurs in case of ***traffic overloading***.

Causes of Congestion

The various causes of congestion in a subnet are:

- The input ***traffic rate exceeds the capacity*** of the output lines. If suddenly, a stream of packet start arriving on three or four input lines and all need the same output line. In this case, a queue will be built up. If there is insufficient memory to hold all the packets, the packet will be lost.

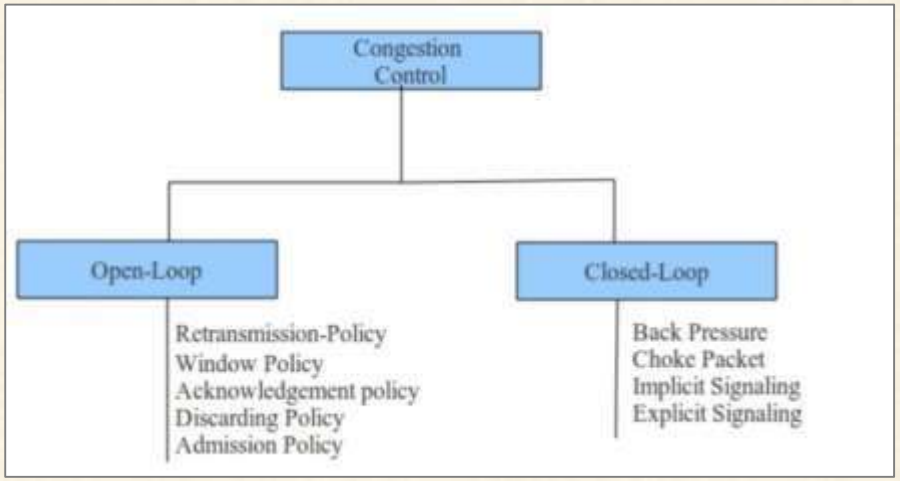


- The routers are ***too slow to perform*** bookkeeping tasks (queuing buffers, updating tables, etc.).
- The routers' buffer is too limited.
- Congestion in a subnet can occur if the ***processors are slow***. Slow speed CPU at routers will perform the routine tasks such as queuing buffers, updating table etc slowly.
- Congestion is also caused by ***slow links***.This problem will be solved when high speed links are used.

Congestion Problem

How to correct the Congestion Problem:

- Congestion Control refers to techniques and mechanisms that can either ***prevent congestion, before it happens, or remove congestion, after it has happened.***
- Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.



These two categories are:

1. Open loop

2. Closed loop

Open Loop Congestion

Open Loop Congestion Control

- In this method, policies are used to ***prevent the congestion before it happens***. Congestion control is handled either by the source or by the destination.

The various methods used for open loop congestion control are:

1. Retransmission Policy

- The sender ***retransmits a packet, if it feels*** that the packet it has sent is ***lost or corrupted***.
- However retransmission in general may increase the congestion in the network. But we need to implement ***good retransmission policy to prevent congestion***.
- The retransmission ***policy and the retransmission timers*** need to be designed to optimize efficiency and at the same time prevent the congestion.

2. Window Policy

- To implement window policy, ***selective reject window method*** is used for congestion control.
- Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this ***duplication may make congestion*** worse.
- Selective reject method ***sends only the specific lost or damaged packets***.

Open Loop Congestion

3. Acknowledgement Policy

- The acknowledgement policy ***imposed by the receiver*** may also affect congestion.
- If the receiver does not acknowledge ***every packet it receives it may slow down*** the sender and help prevent congestion.
- Acknowledgments also ***add to the traffic load*** on the network. Thus, by sending ***fewer acknowledgements we can reduce*** load on the network.

To implement it, several approaches can be used:

1. A receiver may send an acknowledgement ***only if it has a packet to be sent.***
2. A receiver may send an acknowledgement ***when a timer expires.***
3. A receiver may also decide to acknowledge ***only N packets at a time.***

4. Discarding Policy

- A router may ***discard less sensitive packets*** when congestion is likely to happen.
- Such a discarding policy may ***prevent congestion and at the same time may not harm*** the integrity of the transmission.

5. Admission Policy

- An admission policy, which is a ***quality-of-service mechanism,*** can also prevent congestion in virtual circuit networks.
- Switches in a flow first check the resource requirement of a flow before ***admitting it to the network.***
- A router can ***deny establishing*** a virtual circuit connection if there is congestion in the "***network or if there is a possibility of future congestion.***

Closed Loop Congestion

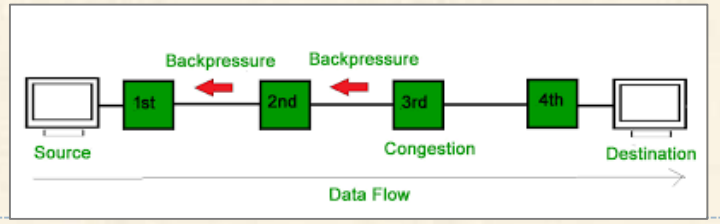
Closed Loop Congestion Control

- Closed loop congestion control mechanisms ***try to remove the congestion after it happens***.

The various methods used for closed loop congestion control are:

1. Backpressure

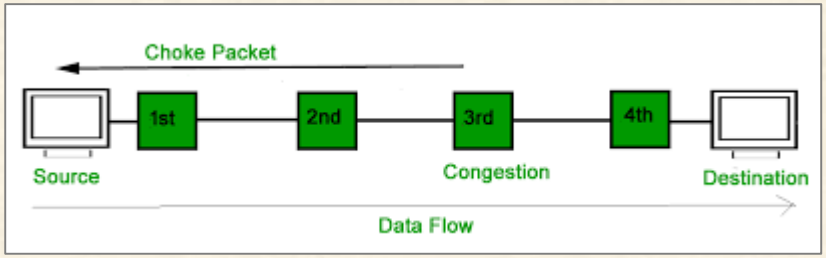
- Back pressure is a node-to-node congestion control that ***starts with a node and propagates, in the opposite direction*** of data flow.
- The backpressure technique can be applied only to virtual circuit networks. In such virtual circuit ***each node knows the upstream node*** from which a data flow is coming.
- In this method of congestion control, the congested node ***stops receiving data from the immediate upstream node*** or nodes.
- This may cause the upstream node on nodes to become ***congested, and they, in turn, reject data*** from their upstream node or nodes.
- As shown in fig node 3 is congested and it stops receiving packets and informs its upstream node 2 to slow down. Node 2 in turns may be congested and informs node 1 to slow down. Now node 1 may create congestion and informs the source node to slow down. In this way the congestion is alleviated. Thus, the pressure on node 3 is moved backward to the source to remove the congestion.



Closed Loop Congestion

2. Choke Packet

- In this method of congestion control, ***congested router or node sends a special type of packet*** called ***choke packet*** to the source to inform it about the congestion.
- Here, congested node ***does not inform its upstream node*** about the congestion as in backpressure method.
- In choke packet method, congested node sends a ***warning directly to the source station*** *i.e.* the intermediate nodes through which the packet has travelled are not warned.



3. Implicit Signalling

- In implicit signalling, there is ***no communication*** between the ***congested node or nodes and the source***.
- The ***source guesses*** that there is congestion somewhere in the network when it ***does not receive any acknowledgment***.
- Therefore the ***delay in receiving an acknowledgment*** is interpreted as congestion in the network.
- On sensing this congestion, the source slows down. This type of congestion control policy is used by TCP.

Closed Loop Congestion

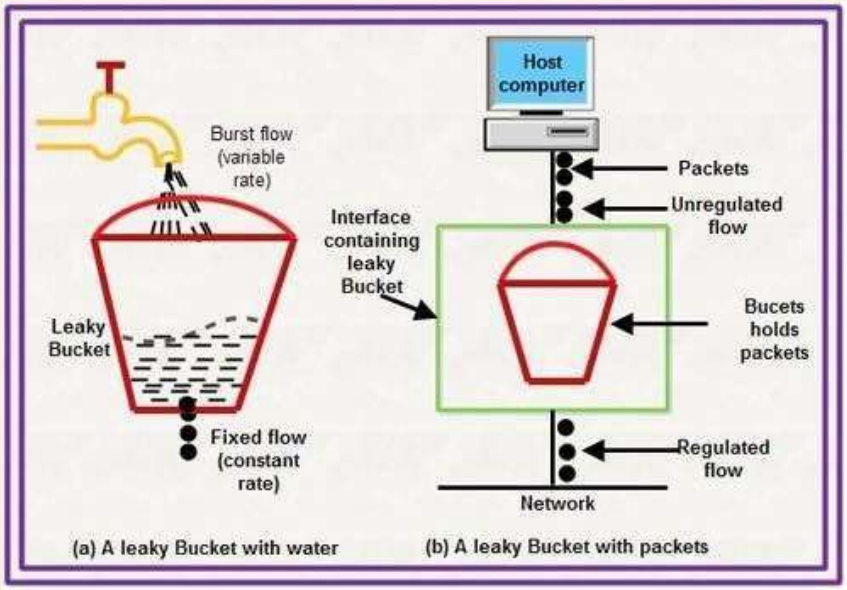
4. Explicit Signalling

- In this method, the ***congested nodes explicitly send a signal*** to the source or destination to inform about the congestion.
- Explicit signalling is different from the choke packet method. In choke packed method, a separate packet is used for this purpose whereas in explicit signalling method, the ***signal is included in the packets that carry data***.
- Explicit signalling can occur in either the ***forward direction or the backward direction***.
- In ***backward signalling***, a bit is set in a packet moving in the ***direction opposite*** to the congestion. This bit ***warns the source*** about the congestion and informs the source to slow down.
- In ***forward signalling,*** a bit is set in a packet ***moving in the direction of congestion***. This bit ***warns the destination*** about the congestion. The receiver in this case uses policies such as ***slowing down the acknowledgements*** to remove the congestion.

Congestion Control Algorithms

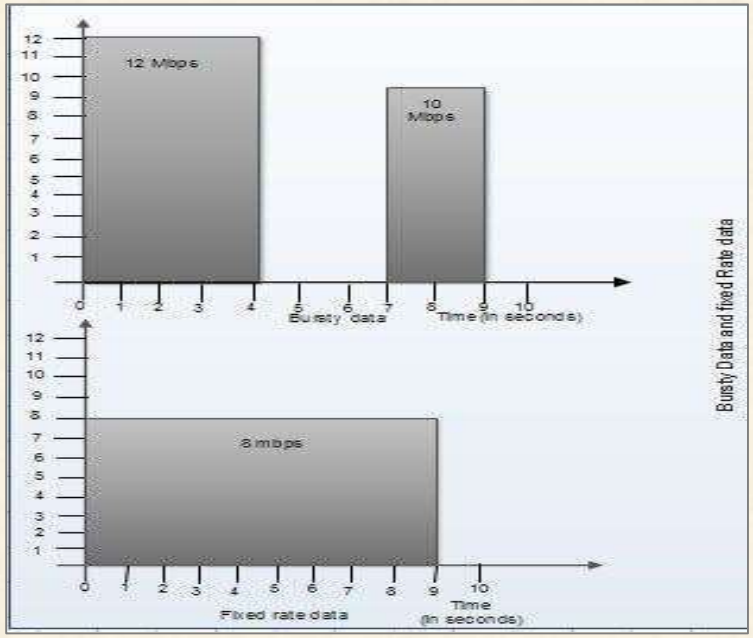
1. **Leaky Bucket Algorithm**

- It is a ***traffic shaping mechanism*** that controls the ***amount and the rate of the traffic*** sent to the network.
- A leaky bucket algorithm shapes ***bursty traffic into fixed rate traffic*** by averaging the data rate.
- Imagine a bucket with a small hole at the bottom.
- The rate at which the water is poured into the bucket is not fixed and can vary but it leaks from the bucket at a constant rate. Thus (as long as water is present in bucket), the rate at which the water leaks does not depend on the rate at which the water is input to the bucket.



Congestion Control Algorithms

- Also, when the bucket is full, any additional water that enters into the bucket spills over the sides and is lost.
- The same concept can be applied to ***packets in the network***.
- Consider that data is coming from the source at variable speeds. Suppose that a source sends ***data at 12 Mbps for 4 seconds***. Then there is no data for 3 seconds. The source again transmits data at a ***rate of 10 Mbps for 2 seconds***. Thus, in a ***time span of 9 seconds, 68 Mb data*** has been transmitted.
- If a leaky bucket algorithm is used, the ***data flow will be 8 Mbps for 9 seconds***.Thus constant flow is maintained.



Congestion Control Algorithms

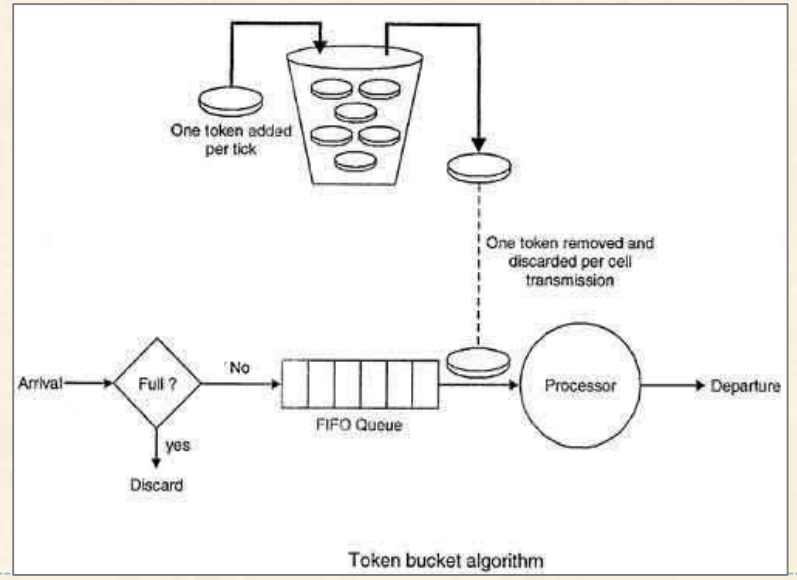
2.Token bucket Algorithm

- The leaky bucket algorithm allows ***only an average (constant) rate of data flow***. Its ***major problem*** is that it cannot deal ***with bursty data***.
- A leaky bucket algorithm does ***not consider the idle time of the host***. For example, if the host was idle for 10 seconds and now it is willing to sent data at a very high speed for another 10 seconds, the total data transmission will be divided into 20 seconds and average data rate will be maintained.The host is having no advantage of sitting idle for 10 seconds.
- To overcome this problem, a token bucket algorithm is used.A ***token bucket algorithm allows bursty data transfers.***

Congestion Control Algorithms

A ***token bucket algorithm allows bursty data transfers.***

- A token bucket algorithm is a modification of leaky bucket in which ***leaky bucket contains tokens***.
- In this algorithm, ***a token(s) are generated at every clock tick***. For a packet to be transmitted, system must remove token(s) from the bucket.
- Thus, a token bucket algorithm allows ***idle hosts to accumulate credit for the future in form of tokens***.
- For example, if a system generates *100 tokens in one clock tick and the host is idle for 100 ticks*. The bucket will contain 10,000 tokens.
- Now, if the host wants to send bursty data, it can consume all 10,000 tokens at once for sending 10,000 cells or bytes.
- Thus a host can send bursty data as long as bucket is not empty.



Token bucket algorithm

Quality of Service (QoS)

- **Quality of Service (QoS)** is a group of technologies that operate on a network to ensure that high-priority traffic and applications may be reliably carried out even when the network's capacity is constrained.

- Additionally, the QoS specifies that supporting priority for one or more flows will not fail other flows. A flow can consist of a packet from a particular application or an incoming interface as well as source and destination addresses, source and destination socket numbers, session identifiers, and packets.

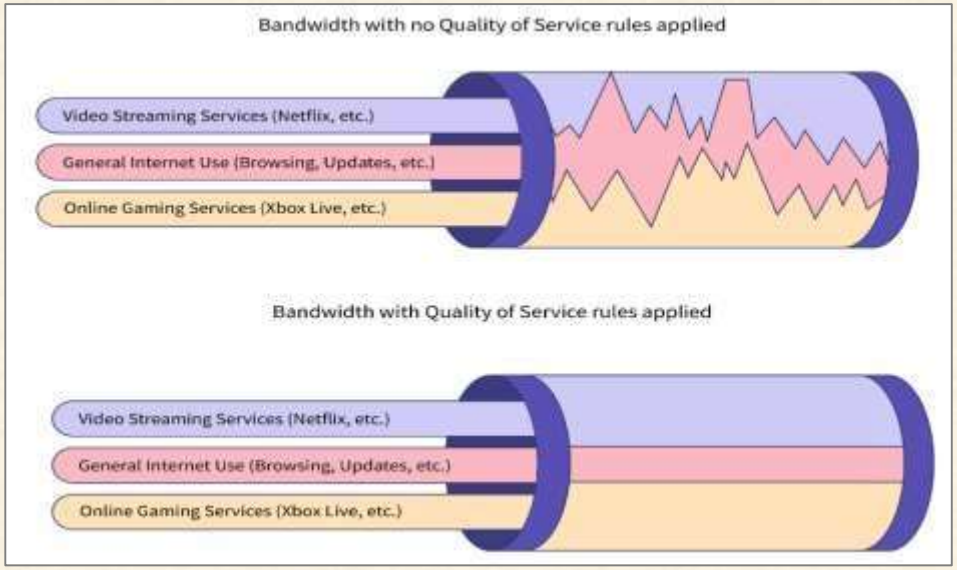
- The companies can avoid interruptions in real-time communications applications such as VoIP (voice over IP), AoIP (audio over IP), and others by using quality of service (QoS).

How does QoS in CN Work?

- QoS facilitates the manipulation of packet loss, postpone and jitter in your community infrastructure. Since we are operating with a finite quantity of bandwidth, our first order of enterprise is to become aware of what packages could advantage from handling those three things.
- Once community and alertness directors become aware of the packages that want to have precedence over bandwidth on a community, the following step is to become aware of that visitors. There are numerous approaches to become aware of or mark the visitors. Class of Service (CoS) and Differentiated Services Code Point (DSCP) are examples.
- We may utilize this information to set policies on those groups in order to give some data streams preferential treatment over others now that we can group data streams into different groups. Queuing is the term for this. The routing or switching device will advance these packets/frames to the front of the queue and transmit them right away, for instance, if voice traffic is tagged and a policy is developed to grant it access to the bulk of network bandwidth on a channel.
- However, if a typical TCP data transfer stream is given a lower priority designation, it will wait (be queued) until enough bandwidth is available to send. These lower priority packets/frames are the first to be dropped if the queues get overcrowded.
- QoS networking generation works through marking packets to become aware of provider types, then configuring routers to create separate digital queues for every utility, primarily based totally on their precedence.
- As a result, bandwidth is reserved for essential packages or websites which have been assigned precedence to get entry.

Why is QoS Important?

- QoS is mainly essential to assure the excessive overall performance of essential packages that require excessive bandwidth for real-time visitors.
- QoS facilitates agencies to save you the postponement of those touchy packages, making sure they carry out to the extent that customers require.
- QoS is more essential as community overall performance necessities adapt to the developing variety of human beings' usage of them.
- QoS is likewise turning more essential because the Internet of Things (IoT) keeps returning to maturity.
- QoS allows the statistics to circulate to take precedence withinside the community and guarantees that the statistics flow as fast as possible.

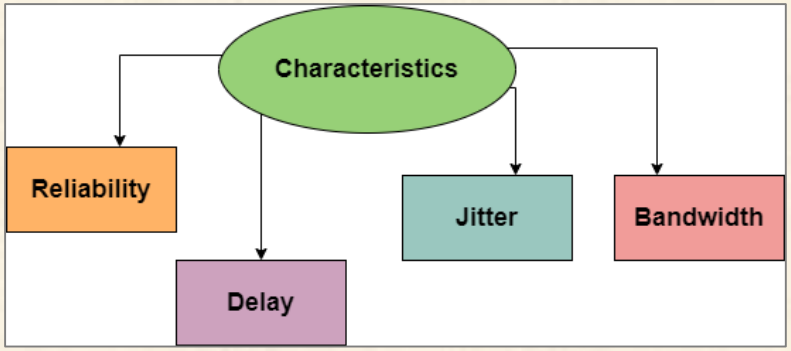


Flow Characteristics

Flow Characteristics

Given below are *four* types of characteristics that are mainly attributed to the flow and these are as follows:

1. Reliability
2. Delay
3. Jitter
4. Bandwidth



Communication Networks Quality Of Service (QOS). - YouTube

QoS Characteristics

- QoS is an overall performance measure of the computer network.

Important flow characteristics of the QoS are given below:

1. Reliability

- If a packet gets ***lost or acknowledgement is not received*** (at sender), the re-transmission of data will be needed. This decreases the reliability. the importance of the reliability can differ according to the application.

For example:

- E- mail and file transfer need to have a reliable transmission as compared to that of an audio conferencing.

2. Delay

- ***Delay of a message*** from source to destination is a very important characteristic. However, delay can be ***tolerated*** differently by the different applications.

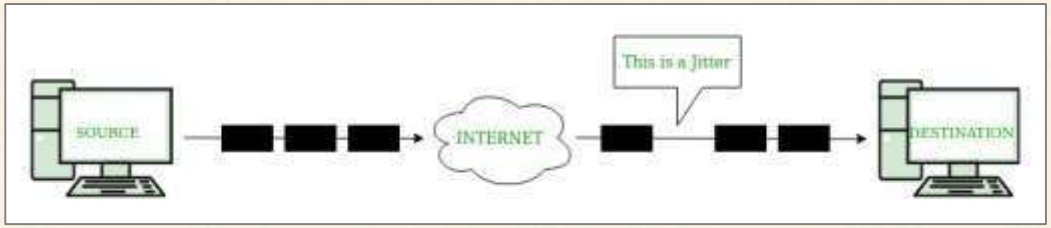
For example:

- The time delay ***cannot be tolerated in audio conferencing*** (needs a minimum time delay), while the time delay in the e-mail or file transfer has less importance.

QoS Characteristics

3. Jitter

- The jitter is the ***variation in the packet delay.***
- If the difference between delays is large, then it is called as **high jitter.** On the contrary, if the difference between delays is small, it is known as **low jitter.**



Example:

Case1: If 3 packets are sent at ***times 0, 1, 2 and received at 10, 11, 12.*** Here, the delay is same for all packets and it is acceptable for the telephonic conversation.

Case2: If 3 packets ***0, 1, 2 are sent and received at 31, 34, 39***, so the delay is different for all packets. In this case, the time delay is not acceptable for the telephonic conversation.

4. Bandwidth

- Different applications need the different bandwidth.

For example:

- Video conferencing needs more bandwidth in comparison to that of sending an e-mail.

Techniques Involved in QoS

Techniques that may be used for QoS are as follows-

1. Scheduling :

- Packets from one-of-a-kind flows arrive at a transfer or router for processing. An exact scheduling approach treats the one-of-a-kind flows truthfully and suitably. Several scheduling strategies are designed to enhance the exceptional of the provider.

Three of them here-

1. **FIFO Queuing** Packets wait in a buffer (queue) in first-in, first-out (FIFO) queuing until the node (router or switch) is prepared to process them. The queue will get full and new packets will be deleted if the average arrival rate exceeds the average processing rate.Anyone who has had to wait at a bus stop for a bus knows what a FIFO queue is like.
2. **Priority Queuing** Packets are first given a priority class in priority queuing. Each type of priority has its own queue. The first packets processed are those in the queue with the highest priority. The final packets processed are those in the lowest priority queue.The system continues to serve a queue until it is empty, it should be noted.
3. **Weighted Fair Queuing** The packets are still allowed to various queues and assigned to various classes in this method. The queues are, however, weighted according to their priority; a higher priority corresponds to a higher weight. The quantity of packets processed from each queue is determined by the associated weight, and the system processes packets in each queue in a round-robin method.

Three packets are processed from the first queue, two from the second queue, and one from the third queue, for instance, if the weights are 3, 2, and 1.All weights can be equal if the system does not give the classes any sort of priority.

Techniques Involved in QoS

2.Traffic shaping

- This technique which is also known as **packet shaping** is a **congestion control or management technique** that helps to regulate network data transfer by delaying the flow of least important or least necessary data packets.
- QoS is included in the service-level agreement when an organization signs it with its network service provider which guarantees the selected performance level.
- Traffic shaping is a mechanism to manipulate the quantity and the price of the visitors despatched to the network.

Two strategies can form visitors: Leaky Bucket and Token Bucket.

Implementation of QoS

Three of the following current models can be used to implement quality of service-

1. **Best Effort:**

- If we use this model, we must treat each data packet equally in terms of priority. However, since everyone has determined the priority order in this manner, there is no assurance that all data packets will be delivered, but every attempt will be made to do so. The best-effort approach is used when networks aren't configured with QoS regulations, or if their network infrastructure doesn't support QoS, it's important to keep in mind.

2. Integrated Services or IntServ:

- The bandwidth over a specified network path is reserved by this QoS approach. Applications request a reservation of network resources for themselves, while network devices simultaneously watch the packet flow to ensure that network resources are open to receiving packets. Keep in mind that the IntServ-capable routers and resource reservation protocol are required while implementing the Integrated Services Model.

This model uses a lot of network resources and is not very scalable.

3. Differentiated Services:

- In this QoS paradigm, network components like switches and routers are set up to handle various traffic types following various priority levels. Depending on its needs, a business can classify the network traffic. For instance, giving audio traffic a higher priority.

Quality of Service (QoS)

There are 2 types of Quality of Service Solutions:

1. Stateless solution

- Here, the server is not required to keep or store the server information or session details to itself.
- The **routers** maintain no fine-grained state about traffic, one positive factor of this is, that it's **scalable and robust.**
- But also, it has ***weak services*** as there is **no guarantee about the kind of performance delay** in a particular application which we encounter. In the stateless solution, the server and client are **loosely coupled** and can act.

2. Stateful solution:

- Here, the server is required to maintain the **current state and session information**, the **routers maintain per-flow state** as the flow is very important in providing the Quality-of-Service which is providing powerful services such as guaranteed services and high resource utilization, provides protection, and is much **less scalable and robust.**
- Here, the server and client are **tightly bounded.**

Advantages of QoS in CN

Major blessings of deploying QoS include:

1. **Unlimited software prioritization:** QoS ensures that businesses' maximum mission-essential packages will usually have precedence and the essential assets to attain excessive overall performance.

2. **Better aid management:** QoS allows directors to higher manipulate the organization's net assets. This additionally reduces fees and the want for investments in hyperlink expansions.

3. **Enhanced consumer experience:** The stop aim of QoS is to assure the excessive overall performance of essential packages, which boils right down to handing over the most excellent consumer experience. Employees experience excessive overall performance on their excessive-bandwidth packages, which allows them to be greater powerful and get their tasks carried out greater quickly.

4. **Point-to-factor site visitors management:** Managing a community is crucial, but site visitors are delivered, be it stop to stop, node to node, or factor to factor. The latter allows businesses to supply client packets so as from one factor to the following over the net without struggling with any packet loss.

Advantages of QoS in CN

5. Packet loss prevention:

- Packet loss can arise whilst packets of records are dropped in transit among networks. This can frequently be because of a failure or inefficiency, community congestion, a defective router, a free connection, or a terrible signal. QoS avoids the capability of packet loss with the aid of using prioritizing bandwidth of excessive-overall performance packages.

6. Latency reduction:

- Latency is the time it takes for a community request to head from the sender to the receiver and for the receiver to method it. This is generally stricken by routers taking longer to research data and garage delays because of intermediate switches and bridges. QoS allows businesses to lessen latency or accelerate the method of a community request with the aid of using prioritizing their essential software.

Disadvantages of QoS in CN

- In maximum companies, the QoS idea isn't carried out properly or is now no longer even carried out, reflecting some commercial enterprise problems.
- Frequent needs for enlargement of the net aid, generated with the aid of using unsatisfactory consumer experiences, can frequently be circumvented thru the software of manage mechanisms, which cost protection and availability.
- QoS prioritization is tool-centric, and we want a few measures which can be greater consumer-centric that could make certain that the decision is walking easily and the download isn't taking greater than essential.

Network Performance issues

One important issue in networking is the performance of the network – how good is it?

Network performance is measured in following fundamental ways

1. Bandwidth
2. Throughput
3. Latency (D elay)

Network Performance issues

1. **Bandwidth**

- ***Informal:*** Maximum amount of data that can be transmitted per second.
- ***Formal:*** The bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period of time.
- Bandwidth (bps) = Capability.

***Example*:** Gigabit Ethernet can provide a bandwidth of 1Gbps.

Bandwidth in Hertz

- A range of frequencies used to transmit signals which is measured in hertz.

2.Throughput

- ***Informal:*** Actual amount of data that passes through the medium.
- ***Formal:*** The throughput is a measure of how fast we can actually send data through a network. It depends on the Latency.
- Although bandwidth in bits per second and throughput seem the same, they are different.
- A link may have a bandwidth of 'B' bps, but we can only send 'T' bps through this link ***with T < B always***.

Network Performance issues

3. Latency (Delay)

- The latency or delay defines how long it takes for an **entire message to completely arrive at the destination** from the time the first bit is sent out from the source.

Latency is made of four components.

- Transmission delay
- Propagation delay
- Queueing delay
- Processing delay

Latency = Transmission delay + Propagation delay + Queueing delay + Processing delay

End Of UNIT-4