COMPUTER NETWORK

Page No.
Date : /

Transport Layer Unit-4

ONE SHOT + 3PYQ Solutions

Topics :-

→ Transport layer & its functions
→ Process to Process delivery.

* → TCP and UDP [Header format] [2022-23, 2021-22]
* → TCP Vs UDP

* → TCP numerical [2018-19]
* → Three-way Handshake.

→ TCP window management System.
→ Flow Control & Retransmission.

→ Multiplexing & Demultiplexing.

→ QoS & its techniques to improve it.

* → TCP Congestion Control.

Subscribe + Join Telegram

Page No.:
Date: / /

Transport Layer

It is the second layer in the TCP/IP model and the fourth layer in the OSI model. It is an end to end layer used to deliver messages to a host.

Working of Transport layer

The transport layer takes services from the Application layer and provides services to the Network layer.

At the sender's side: It receives data from Application layer and then performs Segmentation, divides the actual message into segments, adds to source and destination's port numbers into the header of the Segment, and transfer the message to the Network layer.

At the receiver's side: It receives data from the Network layer, reassembles the segmented data, reads its header, identifies the number, and forwards the message to the appropriate port in the Application layer.

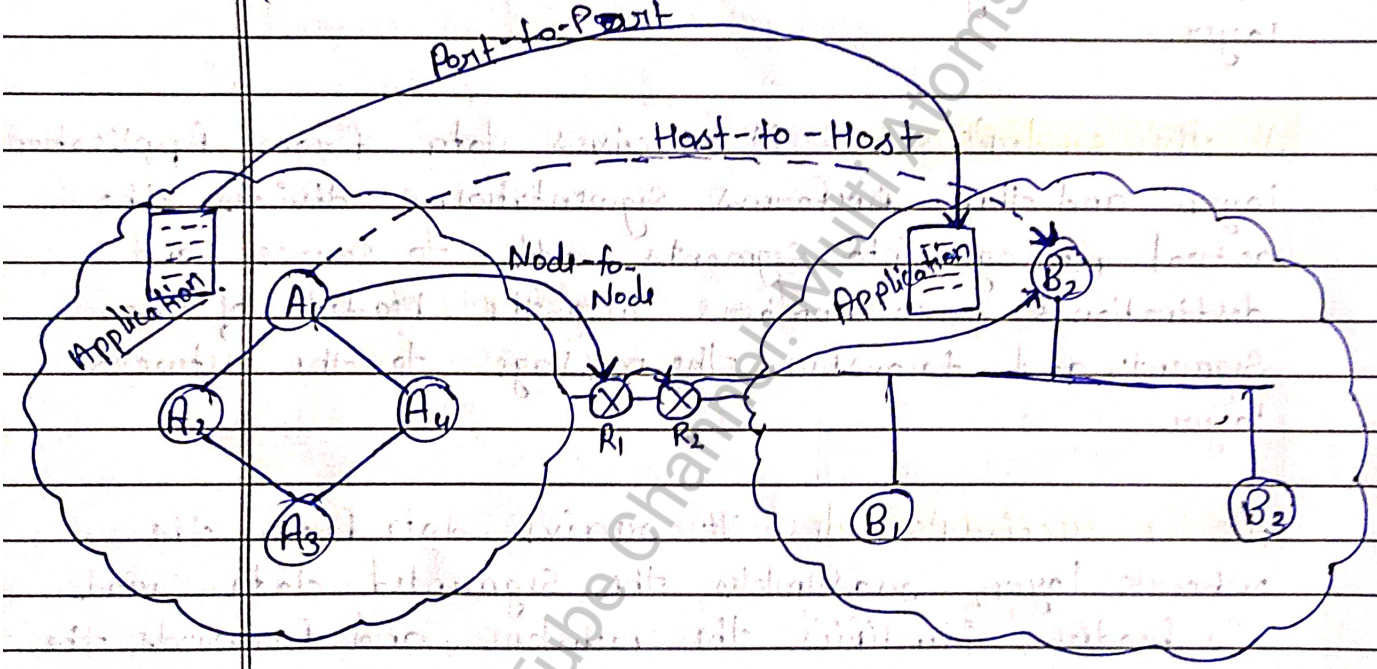
Functions of Transport layer:

1. Process to Process Delivery
2. Multiplexing & Demultiplexing
3. Congestion Control
4. Data Integrity & Error Correction
5. Flow Control.

Page No.
Date / /

Process-to-Process delivery / Port to port / Node to Node

→ A process is an application program running on a host.
→ The data link layer performs a node-to-node delivery.
→ The network layer performs host-to-host delivery.
→ The transport layer is responsible for the delivery of a packet, part of a message, from one process to another.



→ Transport Layer Protocols

① TCP [Transmission Control Protocol]

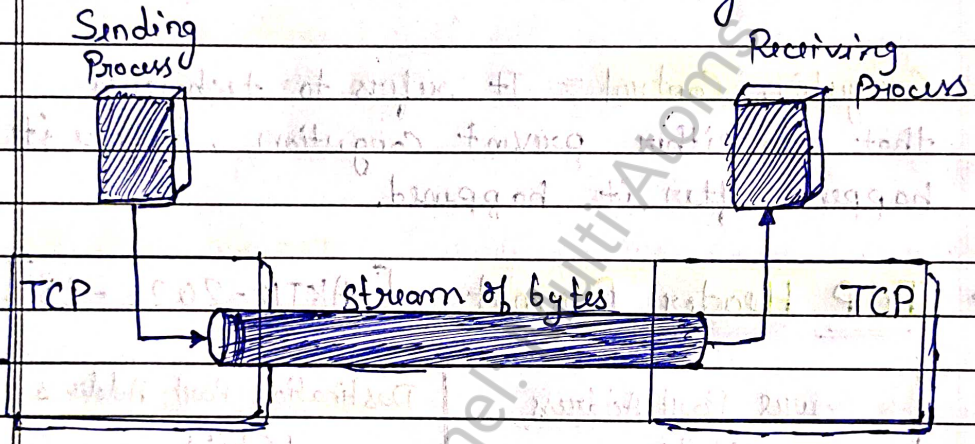
② UDP [User Datagram Protocol]

Page No.:
Date : / /

Transmission Control Protocol

→ TCP is Connection-oriented protocol.

Services TCP provides
- → Byte streaming
- → Connection Oriented
- → Full Duplex
- → Piggy backing
- → Error Control
- → Flow Control
- → Congestion Control



* **Byte Streaming** = TCP sends data as a continous, ordered stream of bytes, simplifying data handling by hiding packet details from application.

* **Connection Oriented** = It establishes a connection before data transfer begins and maintains it until the transfer is complete.

* **Full-Duplex** = TCP processes can send & receive both at the same time.

Page No.
Date / /

* **Piggybacking** :- means sending an acknowledgment along with data in the same packet to save time & resources.

* **Error Control** = TCP ensures data is sent and received accurately by detecting errors, requesting retransmission, and confirming successful delivery.

* **Flow Control** = manages the rate of data transmission between sender and receiver to prevent overwhelm & ensure smooth communication.

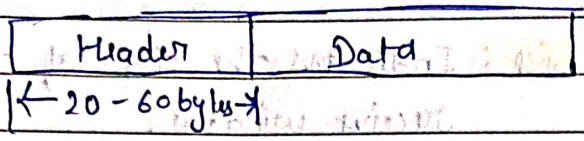
* **Congestion Control** = It refers to techniques that can either prevent congestion, before it happens, after it happened.

→ **TCP Header Format** [AKTU-2022-23]

Source Port Address 16 bits								Destination Port Address 16 bits
Sequence number 32 bits								
Acknowledgment number 32 bits								
HLEN 4 bits	Reserved 6 bits	U R G	A C K	P S H	R S T	S Y N	F I N	Window size 16 bits
Checksum 16 bits								Urgent pointer 16 bits
Options & Padding								

Page No.:
Date:

→ The Segment consists of a 20- to 60- byte header, followed by data + header.



Segment

1. **Source Port** : A 16-bit number identifying the application that TCP segment originated from within the sending host.
2. **Destination Port** : A 16-bit number identifying the application that TCP segment is destined for on a receiving host.
3. **Sequence number** : Identifying the current position of the first data byte. main the sequence of data.
4. **ACKnowledgement no.** : Identifying the next data byte that the Sender expects from the receiver.
5. **HLEN** : length of the header. (multiply by 4)
6. **Reserved** : Reserved for future use. Must be '0'.
7. **Control bit** or **Flags** : one or more of these bits can set at a time.
   i) **URG** : The value of urgent pointer field is valid.
   ii) **ACK** : The value of acknowledgement field is valid.
   iii) **PSH** : Push the data
   iv) **RST** : Reset the data.

Page No.
Date / /

(v) SYN: Synchronize sequence numbers.
(vi) FIN: No more data from sender.

8) Window Size: Indicates the size of the sender's receive window.

9) Checksum: Used for error-checking / control the header and data.

10) Urgent Pointer: If the URG flag is set, this field points to the sequence number of the urgent data.

11) Options: There can be upto 40 bytes of optional information in the TCP header [Maximum Segment Size, Timstamp, etc.].

AKTU - 2018-19

Q. The following is the dump of a TCP header in hexa decimal format:
05320017 00000001 00000000 500207ff 0000000

(I) what is the Sequence number?
(II) what is the destination port number?
(III) what is the acknowledgement number?
(IV) what is the window size?

Hexadecimal no. represent in 4bits.

Page No.:
Date: / /

* Sequence Number (32 bits)

Hexadecimal value = 00000001 = 1×16^0
Decimal Value = 1

* Destination Port number (16 bits) =

Hexadecimal value = 0017 = $(1 \times 16^1 + 7 \times 16^0)$
Decimal Value = 23

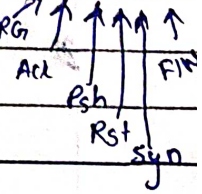
* Acknowledgement number (32 bits)

Hexadecimal Value = 00000000
Decimal Value = 0

* Window Size (16 bits)

Hexadecimal value = 07ff = $(7 \times 16^2 + 15 \times 16^1 + 15 \times 16^0)$
Decimal value = 2047

* Source Port no. = ? [1330]
* HLEN = ? [5 x 4] = 20
* Reserved/Flags = ? / Type of Segment = [SYN] 000010
* Checksum = ? [0]
* Urgent pointer = ? = [0]

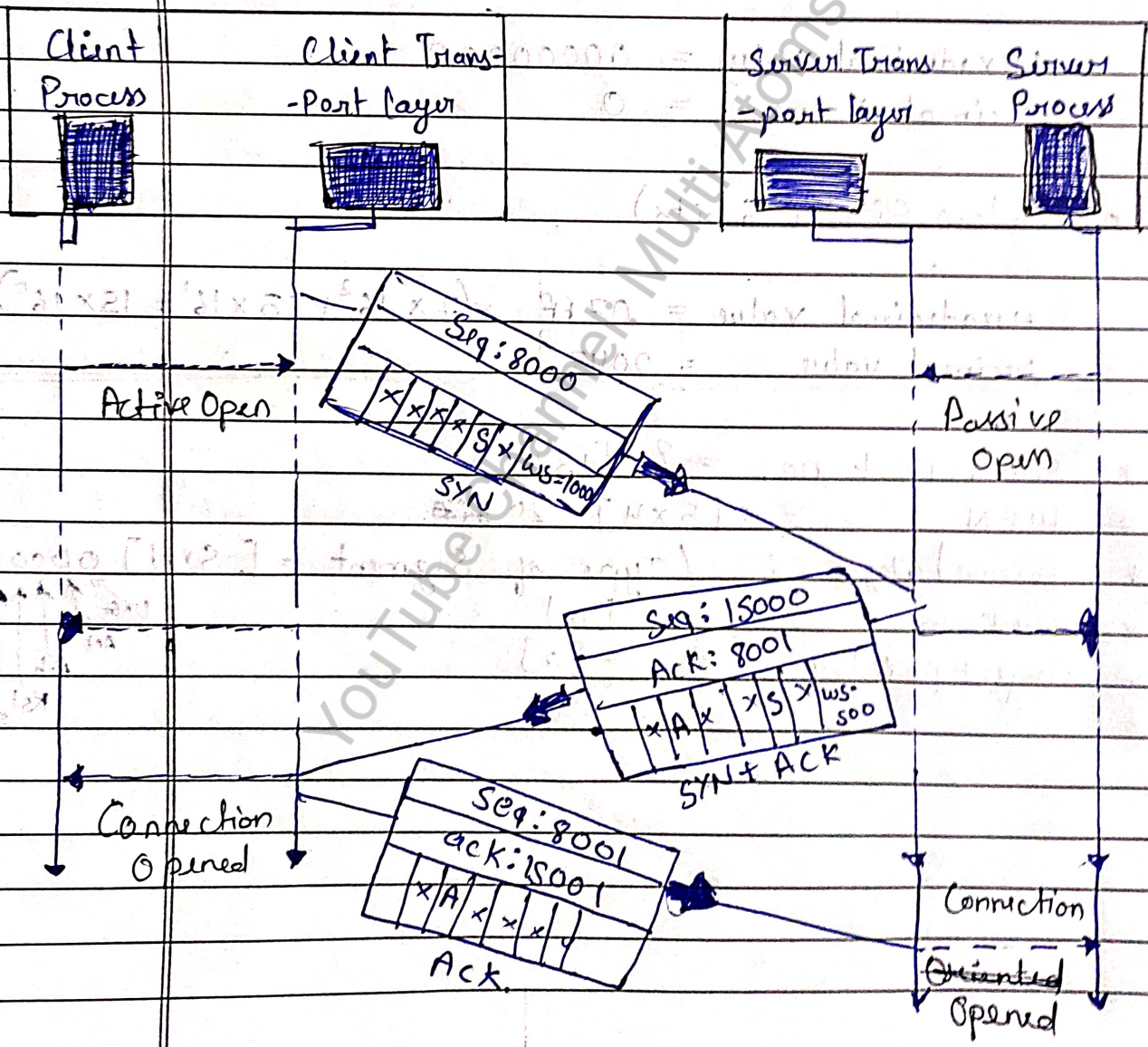


Page No.
Date / /

A TCP Connection

→ The connection establishment in TCP is called three-way handshaking.

→ 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.



Page No.:
Date : / /

* User Datagram Protocol

→ The User Data protocol is a Connectionless, unreliable transport protocol.

→ It does not add anything to the services of IP except for providing process to process communication instead of host to host communication.

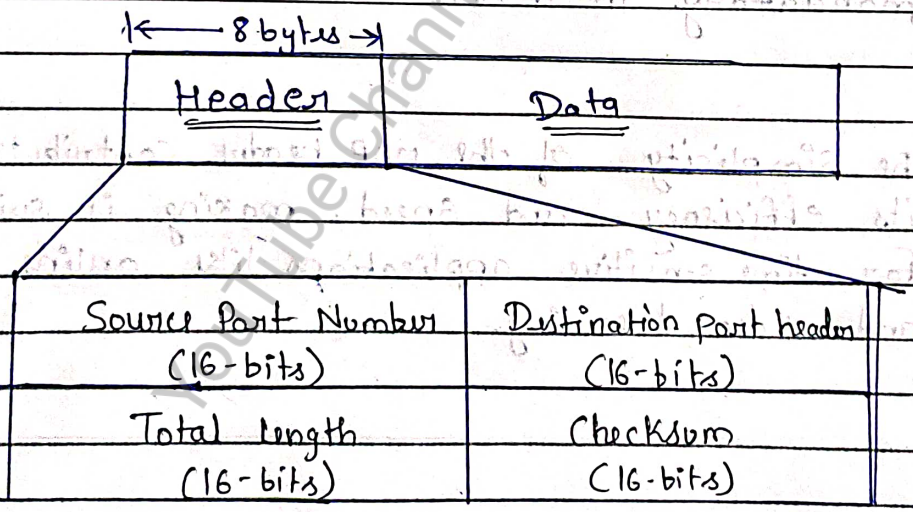
Why to use UDP

→ Low Latency - for real time application. [online Games]

→ Broadcast and Multicast.

→ Data loss is acceptable & Suitable for small packet Exchanges.

* UDP Header [AKTU-2022-23]



Page No.
Date: / /

→ The UDP header is divided into the following four 16-bit fields:

1. Source port: The port number of the sender. This is optional in some cases, but when used it, it helps to receiver to send a reply.

2. Destination Port: The port number of the receiver. This field is essential for directing the packet to the correct application on the destination machine.

3. Length: The length of the entire UDP datagram, including the header + data. mini (8 bytes) to max (65,535 bytes).

4. Checksum: Used for error-checking the header and data. The field is optional in IPV4 but mandatory in IPV6.

→ The simplicity of the UDP header contributes to its efficiency and speed, making it suitable for time sensitive applications like online gaming and video streaming.

Page No.:
Date : / /

[AKTU - 2022-23, 2021-22]

Q. Diff. b/w TCP and UDP

Feature	TCP	UDP
Connection Type	Connection - oriented	Connection - less
Reliability	Reliable (Acknowledge, Retransmission)	Unreliable (no guarantee of delivery)
Flow Control	Yes	No
Congestion Control	Yes	No.
Header Size	Minimum 20 bytes	8 bytes
Error checking	Extensive	Basic
Data Order	Ensures Order	No order guarantee
Connection Establishment	Three - way handshake	None
Use Cases	HTTP / HTTPS, FTP, SMTP, SSH	DNS, VOIP, online gaming, live streaming.

Page No.
Date

* TCP window management System

⇒ Window Management in the context of TCP refers to the process of controlling the flow of data between sender and receiver to ensure efficient and reliable data transmission.

→ It involves managing the amount of data that can be sent before receiving an acknowledgement.

→ This mechanism helps in optimizing network performance & preventing congestion.

→ During the TCP three-way handshake, both the Sender and receiver.
→ The receiver advertises it window size.

Example Scenario:

1. Initial state:
- Receiver advertises a window size of 5000 bytes.
- Sender can send up to 5000 bytes of data without waiting for an acknowledgement.

2. Data Transmission
- Sender sends 5000 bytes.
- Receiver processes data and sends an acknowledgement, updating the window size (3000 bytes).

3. Sliding the Window:

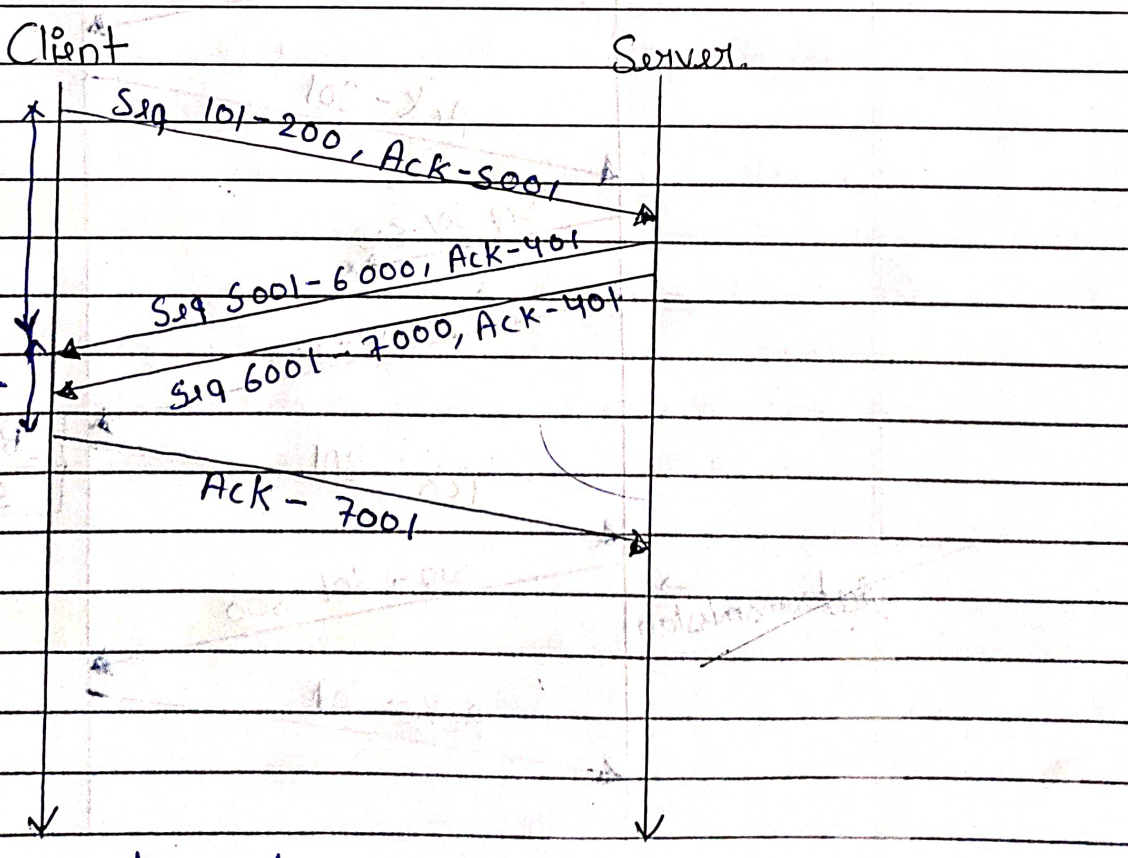
→ Upon receiving the acknowledgement, the sender slides the window forward and sends the next segment of data.

4. Congestion Event:

→ If congestion is detected, the sender reduces the Congestion window size, limiting the data rate until the network stabilizes.

* Flow Control & Retransmission

→ Flow Control is a technique used to prevent the Sender from overwhelming the receiver with too much data quickly & match the rate at which the Sender transmits data which the receiver can process & buffer it.



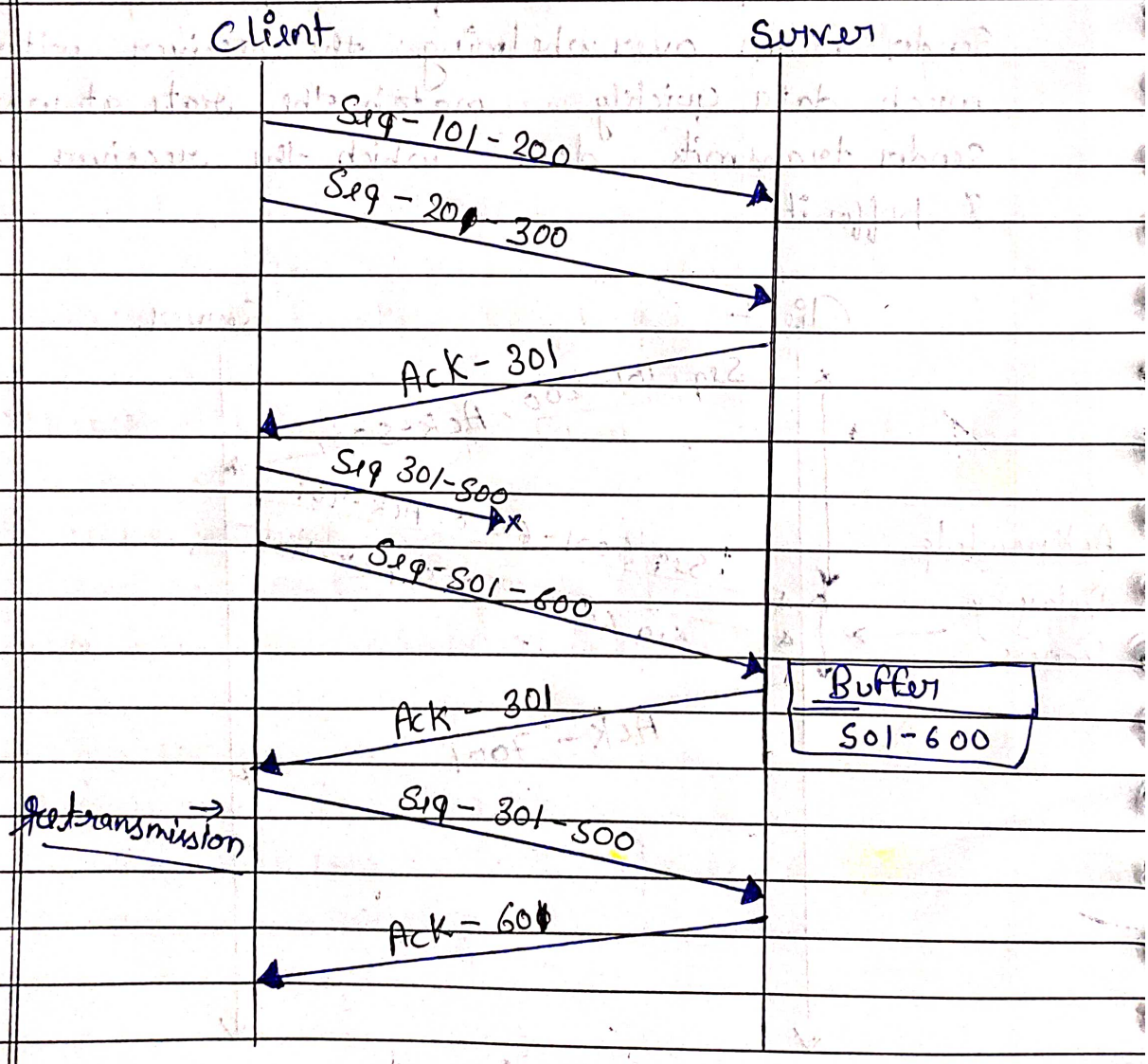
Normal

Page No.
Date: / /

Retransmission is a mechanism to ensure that lost or corrupted packets are resent, maintaining the reliability of TCP connections.

→ If the sender does not receive an ack. for a packet within a specified time, it assumes the segment is lost and retransmit it.

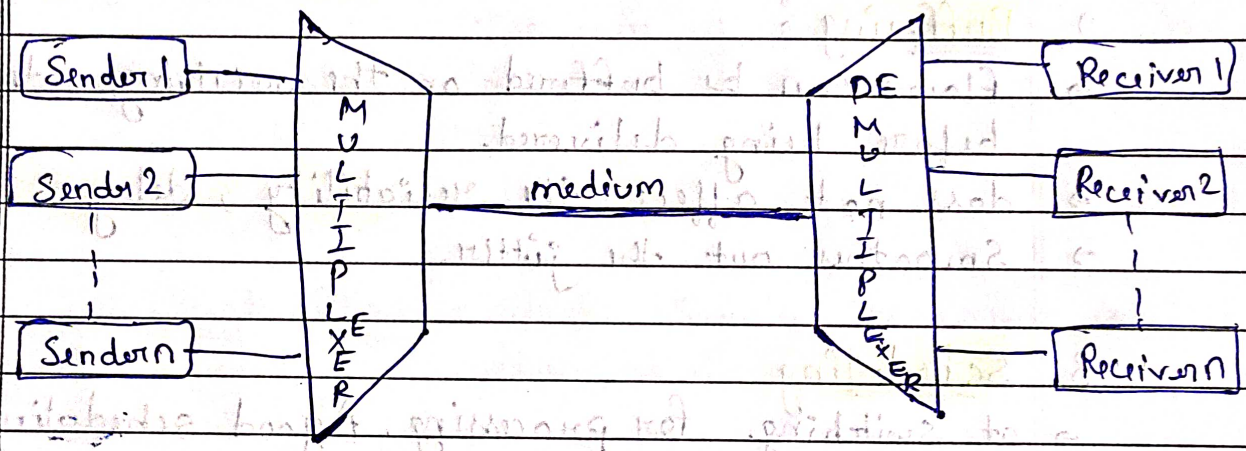
→ Use buffer for storing segments after the same segment lost.



Page No.:
Date: / /

Checkout Unit-1

* **Multiplexing** : It is the process of collecting data from different application processes running on the Sender's end and then gathered data with the header and then transmitting the whole to the receiver. The main purpose of multiplexing is to choose one of the n input lines and transmit it to the output.



Multiplexer Demultiplexer

* **Demultiplexing** : Working of Demultiplexing is just the reverse of the multiplexing process and demultiplexing delivers the segment received from the received from the receiver to the correct process of the application layer.

→ **QoS** → **Quality of Services** → Reliability, Delay, Jitter, Bandwidth.

→ Unit-3

Techniques for improving QoS:

1. **Over Provisioning:**
   - → Increase the capacity of router, Buffer space and bandwidth

2. **Buffering:**
   - → Flows can be buffered on the receiving side before being delivered.
   - → does not affect the reliability, delay
   - → Smoothes out the jitter.

3. **Scheduling**
   - → at switching for processing. A good scheduling technique treats the diff. flows in a fair and appropriate manner.
   - → FIFO and Priority Queuing.

4. **Traffic Shaping**
   - → It is a mechanism to control the amount and the rate of the traffic sent to the network.

   i) Leaky bucket
   ii) Token bucket

Page No.:
Date:

* TCP Congestion Control

→ Congestion occurs if the load offered to any network is more than its capacity.

→ TCP uses a congestion window & a congestion policy that avoid congestion.

Congestion Window - If the network cannot delivery the data as fast as sent by the Sender, it inform to sender to Slow down.

Congestion policy

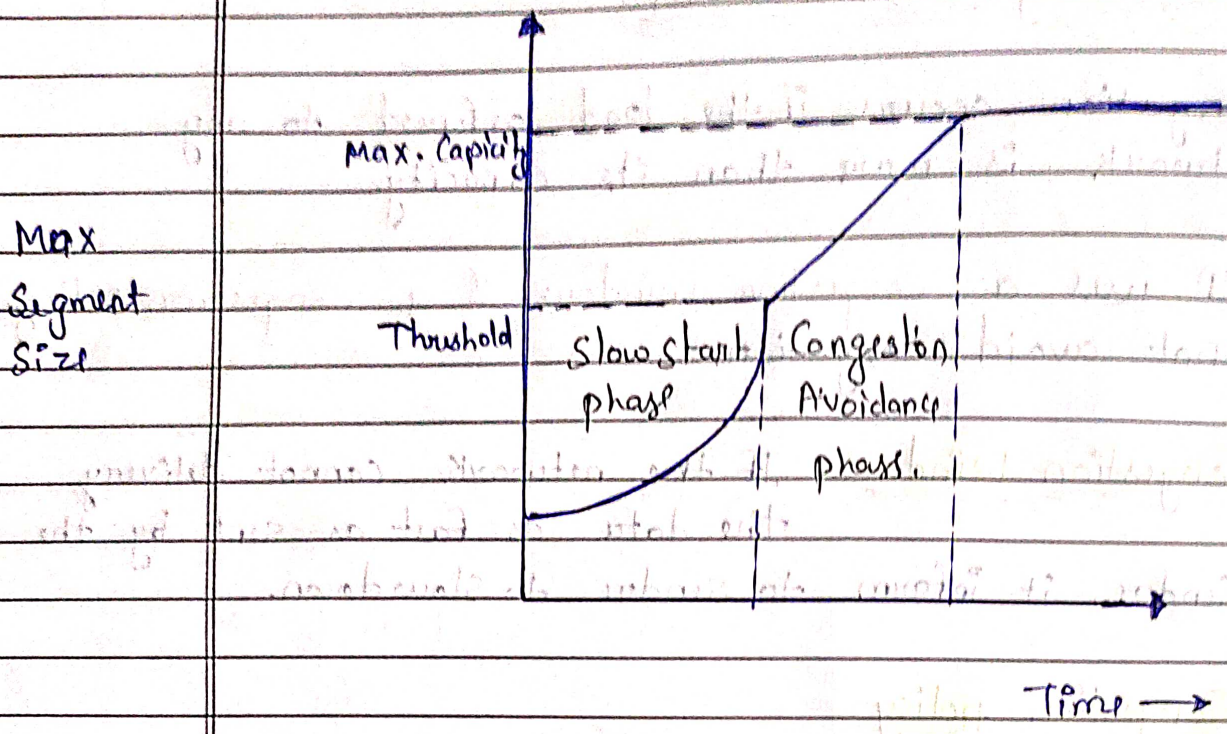
1. Slow Start Phase - Starts slowly increment is exponential to threshold.

2. Cogestion Avoidance phase - After reaching the threshold, increment is by '1'.

3. Congestion Detection phase - Sender goes back to Slow start phase or Congestion avoidance phase.

Case 1 : Retransmission due to Timeout - goes back to Slow start phase.

Case 2 : Retransmission due to 3 Duplicate Acknowledgement - goes back to Congestion avoidance phase.



Unit-4 Completed