CN:-

Transport layer service:-

The Transport layer provides end-to-end communication blw processes running on different hosts. It ensures reliable, ordered, and effient data delivery.

1. process-to-process delivery:-

Transport layer uses port numbers to deliver data to the correct application process (eg; HTTP → port 80, DNS → port 53).

This is also called end-to-end communication.

2. Connection-oriented and connectionless services:-

*TCP provides connection-oriented services with a three-way handshake before data transfer.

* UDP provides connectionless services, sending data without establishing a connection.

3. Reliable Data Transfer:

- Ensures error detection, error recovery, ack.
- Achieved using sequence numbers, Acks, timeouts, etc.
- Mainly supported by TCP.

4. flow control:-

Prevents the sender from overwhelming the receiver.

TCP uses sliding window mechanism to match sender speed with receiver capacity.

5. Error control :-

Detects and corrects error using checksums, Acks, NAKs and retransmissions.

Ensure data integrity during transmission.

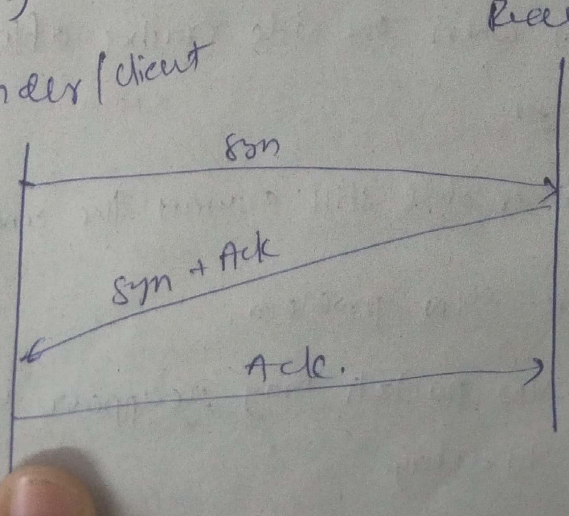
6. Multiplexing :-

* Allows multiple applications to share the same network connection.
* Incoming data is delivered to the correct process using port number.

(2) Connection Establishment in Transport layer.

The T.L, mainly using TCP, establishes a reliable connection blw sender and receiver before data transfer. This process ensures that both sides are ready for communication and agree on initial sequence numbers. The widely used method is the Three-way Handshake

→ Three-way handshake.



1. SYN (Synchronize).

* client sends a SYN segment to the server.
* To request to start a connection and sends intial sequence Number (ISN).
* state changes to SYN-Sent.

2. SYN + Ack.

* server responds with SYN+Ack segment.
* It Ack client's SYN and sends its own ISN.
* server moves to SYN-Received state.

3. Ack.

* client sends a Ack to ack the server's ISN.
* connection is established.
* Both parties enter established state and begin data transfer.

* Issues involved in connection Establishment:

(i) Half-open Connection Problem.

→ occurs when one side crashes after sending a messages.

→ The other side still assumes the connection is active

(ii) Duplicate SYN problem.

→ Old SYN packets may reappear in the network due to delay.

→ could mistakenly create a new connection.

(iii) Timeout Issues:-

→ If SYN or Ack is lost, the handshake fails.

→ TCP uses retransmission mechanism to resend lost SYN packets.

(3) Crash Recovery Mechanism in the T.L.

Crash recovery refers to the techniques used by the T.L - mainly TCP, to ensure reliable communication even when a system crashes or reboots during data transmission.

1. Detecting lost state after crash.

When a host crashes, it loses info such as:

- Sequence numbers.
- Acknowledgements

After reboot, the host must reconstruct this state or reset the connection.

2. Use of sequence Numbers:-

TCP uses sequence numbers to detect duplicates or missing data after recovery.

On restart, the host uses intial Sequene Number (ISN) to avoid confusion b/w old and new data.

3. Handling Half-open connections:-

A half-open connection occurs when:

- one side crashes.
- The other side continues to send data.

4. Timeout and Retransmission:-

If a sender receives no ack:

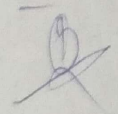
- It assumes either crash or network failure
- uses timeout mechanism to retransmit data.
- After several tries with no reply, it aborts the connection.

5. Recovery after Reboot.

When the crashed host restarts:

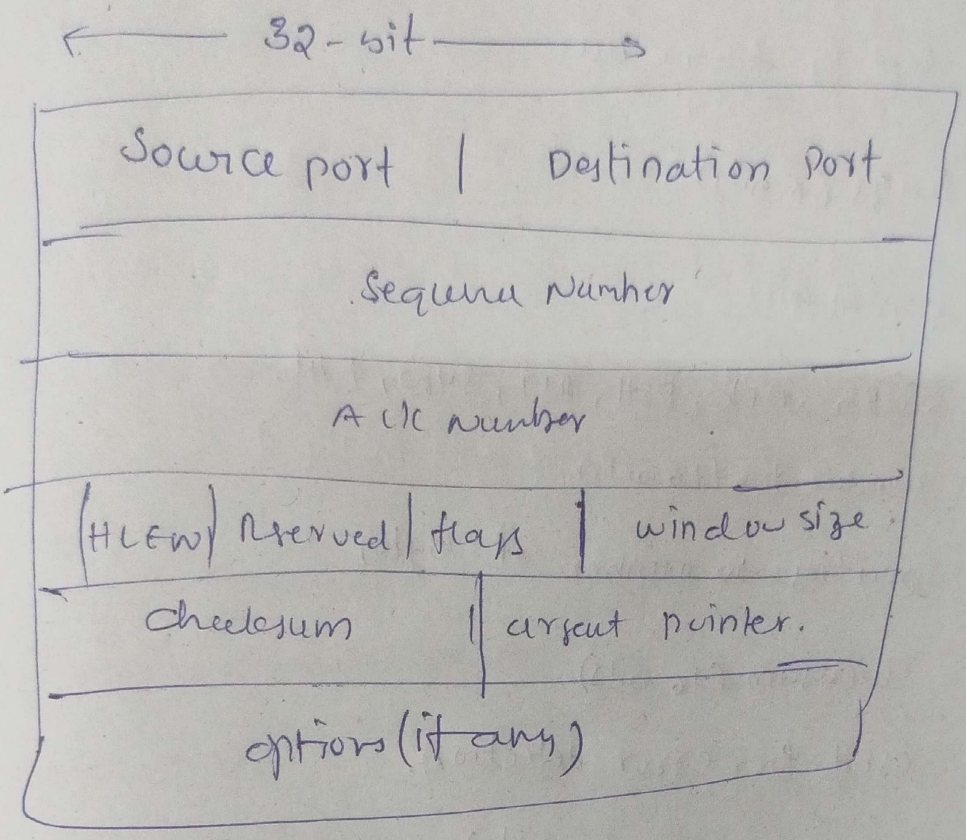
- It discards old connection info.
- Sends or responds with RST segments to clear old connections.
- Establishes a new connection using three-way handshake

(4). Two-Army problem:-



(5) TCP Header with neat diagram:-

The TCP header contains several fields used for reliable, connection-oriented communication.



1. Source port:- (16 bits)
   Identifies the sending application process.

2. Dest. port. (16 bits)
   Identifies the receiving application process.

3. S.N (32 bits)
   Indicates the byte number of the first data byte in the segment.

4. Ack N. (32 bits).
   Specifies the next bytes expected from the sender.
   Used for reliability.

5. Header length (HLEN - 4 bits)

Indicates the length of the TCP header.

6. Reserved (6 bits)

Kept for future use.

7. Flags (6 bits).

control bits.

. URG, Ack, PSH, RST, SYN, FIN.

8. window size (16 bits).

Specifies the number of bytes.

9. Checksum (16 bits).

used for error detection.

10. urgent pointer (16 bits).

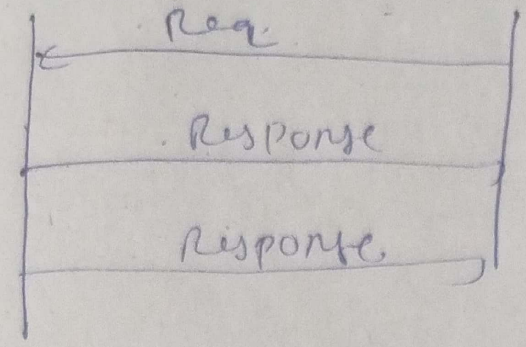
Points to urgent data when URG flag is set.

11. options.

used for MSS, window scaling, timestrap, etc.

⑥ UDP (User Datagram protocol).

It is a connectionless, lightweight, and fast transport layer protocol used for applications where speed is more important than reliability.



1. connection less service :-

. UDP does not establish a connection before sending data

. There is no handshake, which reduces delay.

. Each message is sent independently.

2. Unreliable Transport:

UDP does not provide reliability:

. No ack

. No retransmissions.

. No guarantee of packet delivery, order.

3. Faster Data Transfer.

Since UDP avoids flow control, error recovery, and congestion control, it offers:

. Low latency.

. High throughput.

. Minimal overhead.

Applications :-

Used in applications that need speed, low delay, and real-time performance, such as:

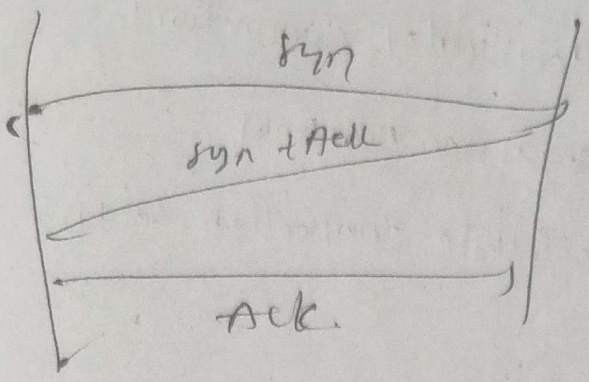
- video streaming. (youtube, etc)
- online gaming.
- DNS look ups.
- DHCP.
- TFTP, SNMP.

→ Adv:-
- very fast.
- low overhead.
- Suitable for broadcast and multicast.
- Ideal for real-time services.

→ Dis adv:-
- no gaarntee of delivery.
- No sequencing.
- No reliability mechanisms.

(7). 3-way handshake:-

TCP uses a three-way handshake to establish a reliable connection blw sender and reciever. The purpose is to synchronize sequence numbers, exchange intial parameters, and ensure that both sides are ready for communication.



step 1: SYN.

. The client intiatie the annection.

. It sends a SYN segment to the server.

. This segment contains the client's intial sequence number (ISN = x).

- client → server . SYN, seq = x.

step 2: SYN + Ack.

. The server receioves the SYN and agree to connect.

. It sends back a SYN + Ack segment.

- The ISN = y.

- server → client : SYN, seq = y, + Ack = x+1.

Setps 3: Ack.

. The client rec. the SYN + Ack.

It sends a final Ack segment to the server.

client → server : Ack, seq = x+1 + Ack = y+1.

Adv:-

- Avoids old duplicates connections.
- Synchronize sequence numbers.
- provides reliable connection setup.

(8) Why TCP and UDP called End-to-End protocols.

Transport layer protocols like TCP and UDP are called end-to-end protocols because:

1. communication occurs b/w two end systems (hosts):-

They deliver data from one process on a source host to another process on a destination host.

2. Use of Port numbers:-

TCP/UDP uses port numbers to identify specific applications.

3. provide services Directly to Applications.

T. L services are provided b/w end hosts;

4. Routers Do NOT participate:-

Intermediate routers only forward packet; they do not manage TCP/UDP state.

All control happens only at endpoints.

Diff" Bw

feature	TCP	UDP
Type	connection oriented	Connectionless
Reliability	Reliable (Ack	unreliable (no Ack)
ordering	Ensured, in-order delivery	No ordering.
Error control	Yes, strong	Limited (Checksum only)
flow control	Yes (Sliding window)	No f.c
Speed	Slower due to overhead	Faster, low latency.
Header size	20 bytes	8 bytes.
use cases	web, Email.	Streaming, Gaming, DNS.

(9). Error control :-

Error control ensures that the data received at the destination is correct, complete, and free from corruption.

Transport layer maintains reliability through mechanisms :-

1. checksum - Detect corrupted data.
2. Ack and NAck - Receiver tells sender whether data arrived correctly.
3. sequence numbers : Detect missing or duplicate packets.

TCP uses strong error control.

UDP " only checksum.

Flow control :-

* It prevents the sender from sending data faster than the receiver can process.

mechanisms :-

1. Sliding window protocol (TCP).

Receiver controls the window size → tells sender how much data can be sent.

2. Receiver Advertised window (rwnd).
Ensure the sender transmits only what the receiver can handle.

* It avoids buffer overflow.
* It prevents data loss.
* Maintains smooth communication.

→ UDP does not support flow control.