* Transport Layer :-

TL provides end to end or process to process Communication.

→ TL protocol can be either Connectionless or Connection-oriented

- A Connection Less TL treats each segment as an Independent packet and delivers it to the TL at the destination machine.
- A Connection oriented TL makes a Connection with the TL at the destination m/c first before delivering the packet. After all data is transferred.

→ There are some Common protocols -

1) UDP – Connectionless + Unreliable
2) TCP / SCTP – Connection-oriented + Reliable

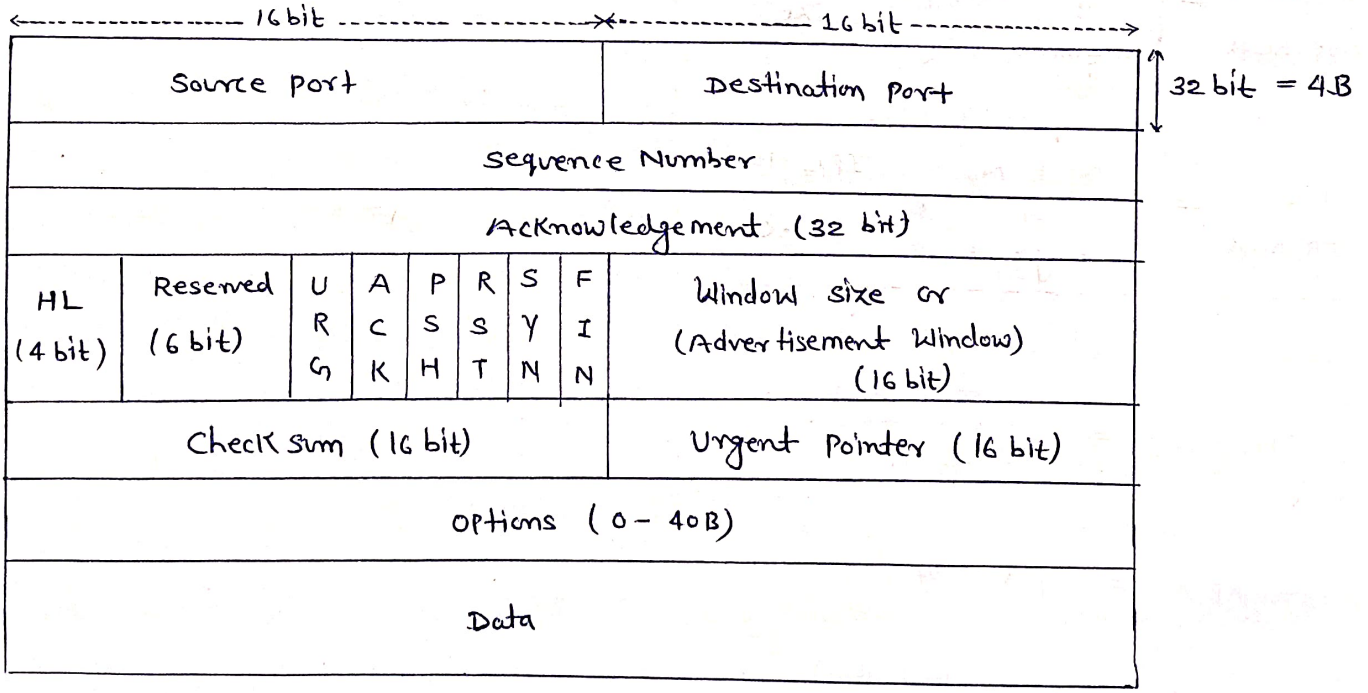
- **Addressing the port No. -**

→ To identify the process inside a host
→ Port No. is 16 bit Integer, Range = (0 to $2^{16}-1$)
→ IANA has divided the port numbers into three ranges:

A) Well Known → (0 – 1023)
B) Registered → (1024 – 49151)
C) dynamic (private) → (49151 – 65535)

SMTP = 25
HTTP = 80
FTP = 20, 21
DNS = 53
POP3 = 110
IMAP-4 = 143

* TCP Header :-



Minimum Header size = 20B

Maximum Header size = 60B

- HL (Header Length) = 4 bit field
  - ↳ max = 1111 = 15
  - Scaling factor = 4

if Header size = 40B

$$HLF = \frac{40}{4} = 10 = \underline{1010}$$

[Header size = (20B - 60B)
  HL = (0 - 15)

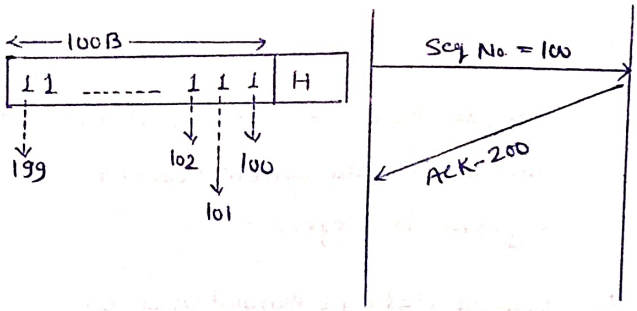
- Source port Address:
  - ↳ Port No. of application program in the Host that is sending the segment.

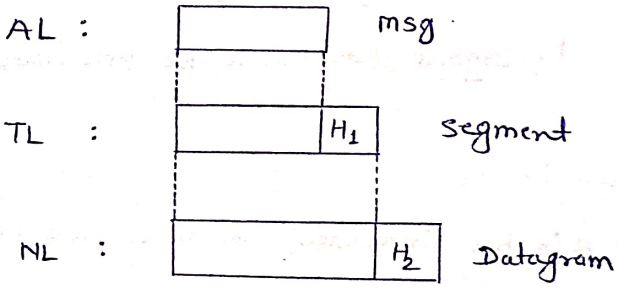
- Destination port Address:
  - ↳ Port No. of application program in the receiver that is receiving the segment.

TCP is a Byte stream protocol, i.e. every byte is associated with sequence Number.

- Sequence Number:
  - → This is 32 bit field defines the sequence Number of first data byte.

- Acknowledgement No. -
  - → This is a 32 bit field that defines the sequence no. of the next expected byte.
  - → If Receiver has successfully received the byte Number x from other Party, It returns x+1 as the ACK No..





data size at TL = Total Length (IP) - IP(H) - TCP(H)

Q. HL = 10, Total Length = 1000 } IP Header

HL = 5, Seq No = 100 } TCP Header

ACK No. ?

Ans :- data size at TL = 1000 - 40 - 20

= 940

ACK No. = 940 + 100

= 1040

Note :-

1) TCP suggests that do not start sequence Number with 0.

2) Choose any Random sequence Number from 0 to $2^{32}-1$.

- URG bit :-

→ Used to treat cretain data on the Urgent basis.

→ if URG bit = 1

↳ Indicate Receiver that certain amount of data within current segment is urgent.

↳ Urgent data is pointed out by evaluating the urgent pointer field.

↳ Urgent data has to be prioritized.

- ACK bit :-

→ ACK bit Indicates Wheather ACK No. field is valid or Not.

→ For all TCP segment except request segment, ACK bit is set to 1.

- PSH bit :-

→ Used to push the entire buffer immediately to the receiving application.

- RST bit :-

→ Used to Reset the TCP Connection.

→ RST = 1 : terminate the Connection

- SYN bit

→ Used to synchronize the sequence Numbers.

→ Syn = 1

↳ Indicates receiver that the Seq. No. Contained in TCP Header is the Initial seq. Number.

- FIN Bit

→ Used to Terminate the TCP Connection

- Options -

1) Time stamp

2) Window size extension

3) Parameter Negotiation

4) Padding

* Wrap around time :-

→ Time taken to use up all the 2^{32} seq. No. is called Wrap around time.

→ depends upon the BW of network.

if BW = x Bytes / sec.

1 sec. ⟶ x Byte transfer

x Byte ⟶ 1 sec.

x seq. No. ⟶ 1 sec.

1 seq No. ⟶ $\frac{1}{x}$ sec.

2^{32} seq No. ⟶ $\frac{2^{32}}{x}$ sec.

Wrap Around time

→ In modern Computers :

Life time of a Tcp segment is

180 sec. = 3 min

if x = 1 MBPS = 10^6 bytes/sec.

WAT = 4294.96 sec.

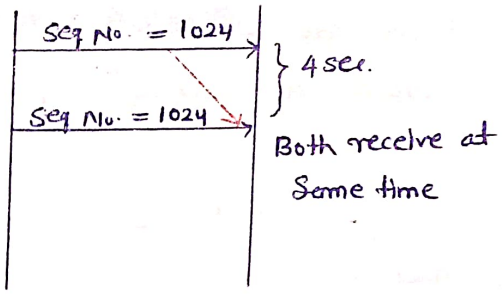
Life time (LT) = 180 sec

WAT > LT (✓)

if x = 1GBPS = 10^9 Byte / sec.

WAT = 4.294 sec.

WAT < LT (Problematic)



Soln : Same sequence No. will not repeat with in Life time.

So We have to increase seq. No.

BW = 10^9 Byte / sec

1 sec. ⟶ 10^9 seq. No.

180 sec. ⟶ 180×10^9 seq. No.

Min. seq. No. required to Avoid Wrap Around time With in Life time =

180×10^9

= LT x BW

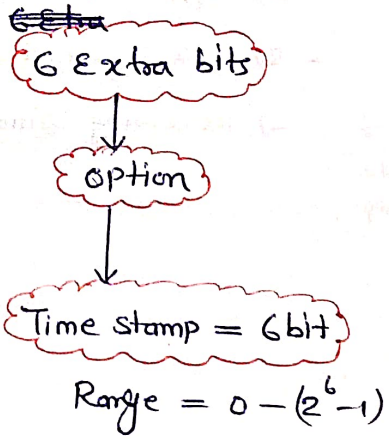
Min. No. of bits required in seq. No. field = $\lceil \log_2 (LT * BW) \rceil$

$= \lceil \log_2 (180 \times 10^9) \rceil$

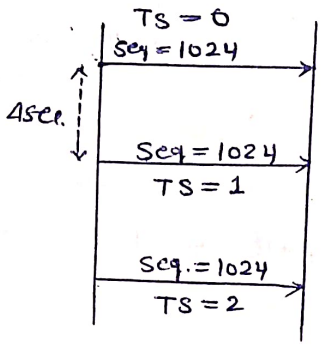
= 38 bit

Extra bits = 38 − 32 = 6 bits

• BW must be in Byte / sec.



→ So there are $2^6 = 64$ sets of 2^{32} Seq. No., to Send 2^{38} seq. No.



* Phase of TCP Connection :-

1.) Connection Establishment phase
2) Data Transfer phase
3) Connection Termination phase

1) Connection Establishment phase :-

- 3-Way Hand shake :-

Step-01 : (SYN)

→ Client Wants to establish a Connection With a Server, so it Sends a segment With SYN (synchronize sequence Number)

→ Request segment Contains the following information in TCP Header-

1.) Initial seq. No.
2) SYN = 1
3) max segment size (MSS)
4) Receiving Window size

data only

Step-02 : (SYN + ACK)

→ After Receiving the request segment, Server responds to the client by sending reply.

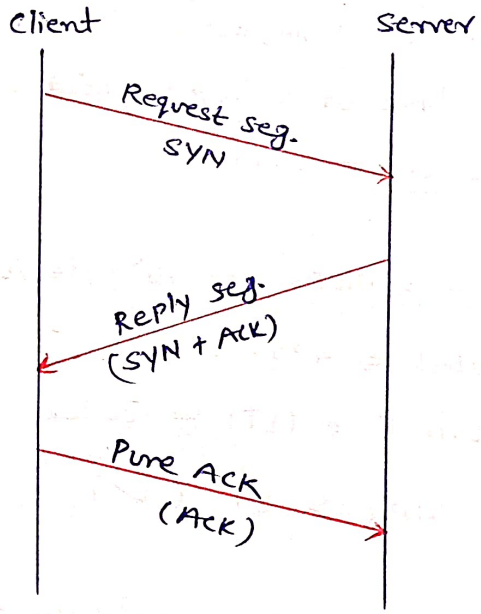
→ Reply segment Contains -

1) Initial seq. No.
2) SYN = 1
3) MSS
4) Receiving Window size
5) ACK No.
6) ACK = 1

Step-03: (ACK)

→ After Receiving the Reply segment client acknowledges the response of server.

→ It send pure ACK.

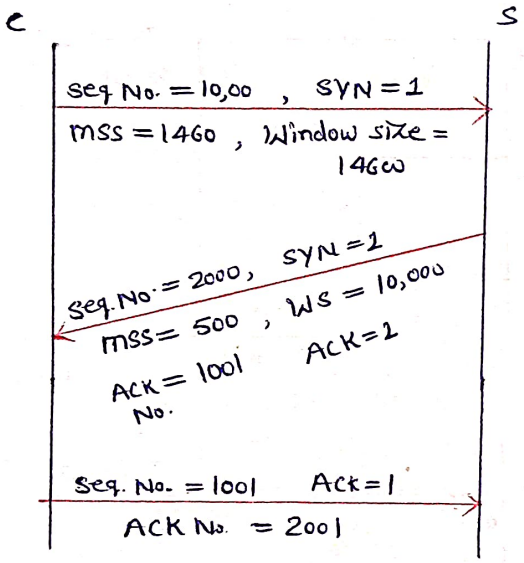


- SYN = 1 : Consume 1 Seq. No.
- FIN = 1 : Consume 1 Seq. No.
- ACK = 1 : Consume no seq. No.
- 1 databyte : Consume 1 seq No.

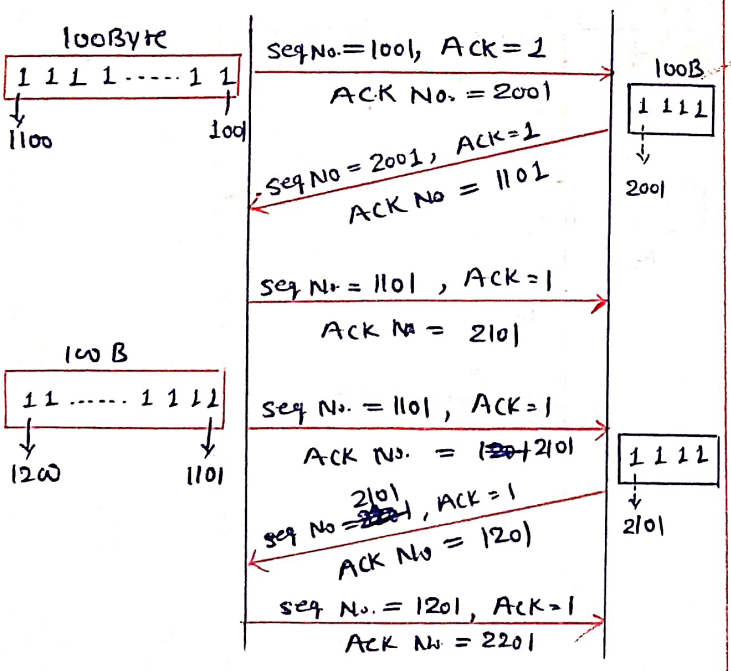
SYN	ACK	
1	0	→ Request
1	1	→ Reply
0	1	→ Pure ACK or data
0	0	→ Data

2) Data Transfer phase :-

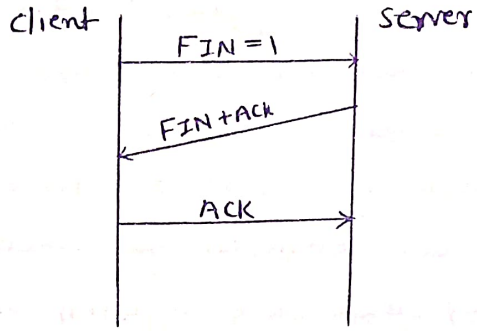
- Connection Establish



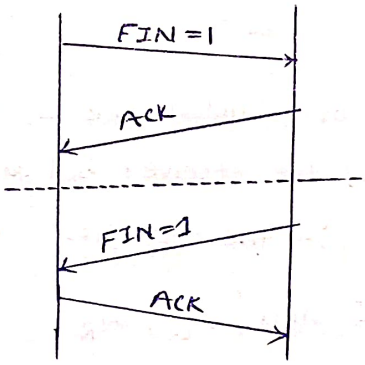
- Data Transfer



3) Connection Termination



(3 Way Hand Shake)



* TCP Congestion Control :-

In TCP sender Window size and Receiver Window size are equal.

→ TCP React to Congestion by reducing the sender Window size.

$$W_s = \min\{\text{Network Capacity}, \text{Receiver Capacity}\}$$

$$\Downarrow$$

W_c (Congestion Window)

$$\boxed{W_s = \min\{W_c, W_R\}}$$

→ An Internet is a Combination of networks and Connecting devices (e.g. Routers). A Packet from a sender may pass through several routers before reaching its final destination.

→ If the Router receives Packets faster than it can process, Congestion might occur and some packets could be dropped. When a packet does not reach the destination, no ACK is sent for it. Then Sender retransmit the packets, This may create more Congestion and more dropping of Packets.

* Congestion Window :-

→ In TCP Sender's Window size is determined not only by the receiver but also by Congestion in the network.

→ $W_s = \min\{W_c, W_R\}$

- TCP Congestion Policy :-

Consist of following 3 Phases -

1) Slow start
2) Congestion Avoidance
3) Congestion detection

1) Slow start phase (Exponential Increase)

→ W_c = maximum segment size

→ After Receiving Each ACK, Sender increases the Congestion Window size by 1 mss. (or double After 1RTT)

→ Size of Congestion Window increases exponentially.

→ Start with $W_c = 1$
↓
2
↓
4
↓
8
⋮

if W_R = 1024 Segment

Threshold (TH) = $\frac{1}{2} W_R$

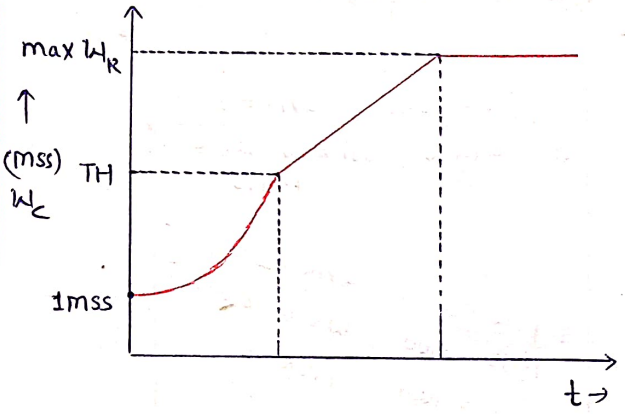
TH = 512 Segment

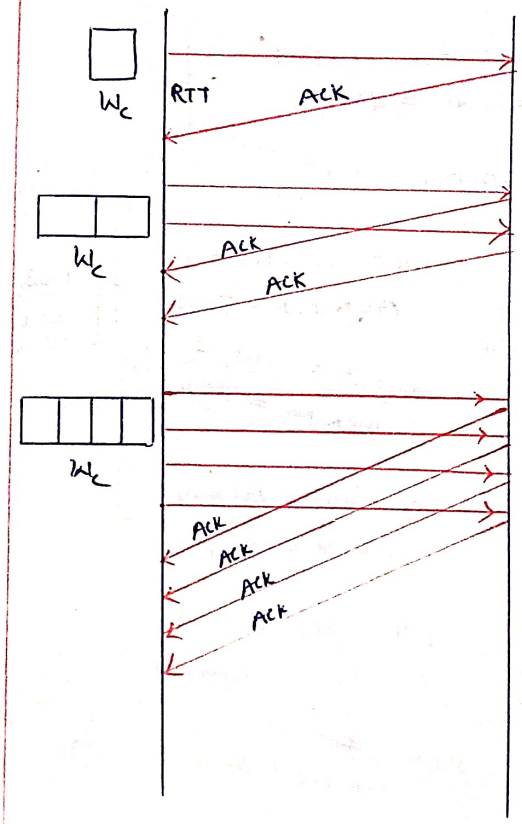
W_c : 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 513, 514, 515, 516, 517 ..... 1024, 1024, 1024

2) Congestion Avoidance (Additive Increase)

→ After the threshold, sender increases the Congestion Window size Linearly to avoid Congestion.

$W_c \leftarrow W_c + 1$





In Congestion Avoidance –

- After 1 RTT the Congestion Window will be increased by one only.
- if An ACK arrives : $W_c = W_c + \frac{1}{W_c}$

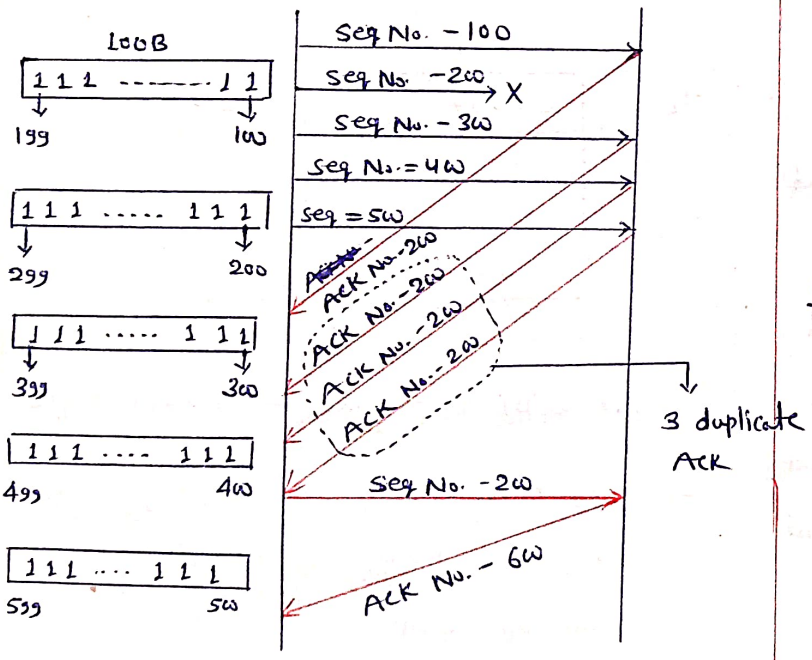
3) Congestion detection :-

→ Can be detected in two Ways:

1) Time out
2) 3 duplicate ACK

- **3- duplicate ACK :-**

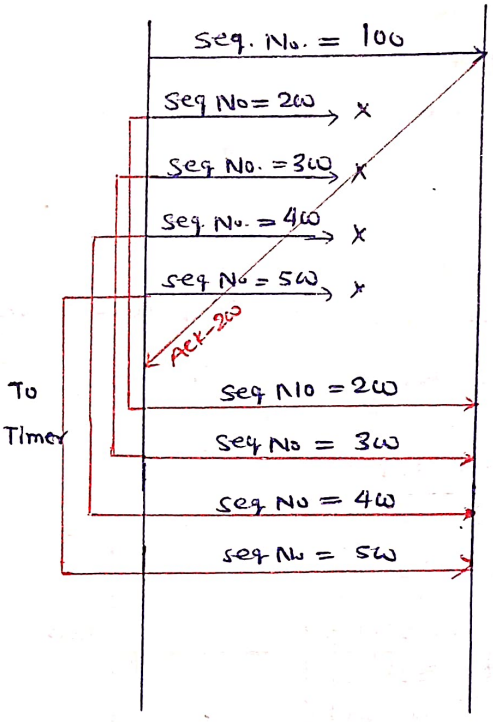
↳ This indicate mild Congestion Condition. In this Case the new Threshold Value is set to half of the current Window size and next transmission start from new threshold value and algorithms enters in a Congestion Avoidance phase



- **Time out Timer :-**

↳ Time out expires before receiving the ACK for a segment.

↳ This Case suggests the stronger Possibility of Congestion in network.



→ In this Case, the new threshold value is set to half of the Current Window size and next Transmission starts from one segment and Algorithm enters in a slow start phase.

Exa :- W_R = 128KB, MSS = 1KB

No. of segments = $\frac{128}{1}$ = 128

$TH = \frac{1}{2}(W_R \text{ Segments})$

= 64

W_c : 1 2 4 8 16 32 64 65 66 67 128

If i got 3 duplicate ACK at 68

New TH = 34

⇒ 1 2 4 8 16 32 64 65 67 68 34 35

36 37 38 1 2 4 8 16 19 20

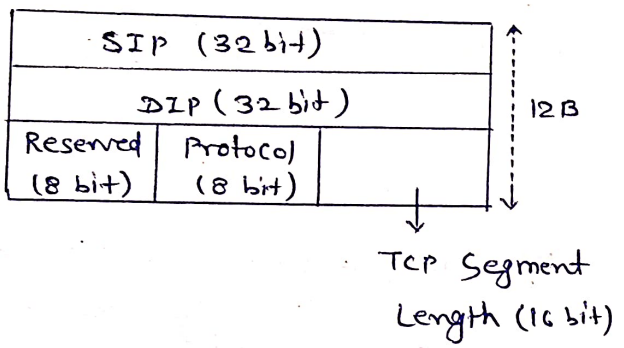
To Time Expire

↳ New Threshold value = 19

* Tcp checksum :- (16 bit)

→ Computed on Tcp data, Tcp Header and Ip pseudo Header.

• Ip Pseudo Header :-



Note :-

1) Tcp is not useful for Broad Casting and multi-Casting.

2) Tcp implements today do not discard out of order segment, They stored them temporarily and flag them as out of order untill the missing segment arrive.

✓Tcp guarantee that data are delivered to the process in order.

* UDP (user datagram protocol)

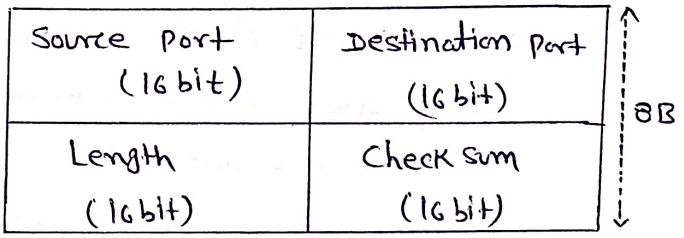
→ UDP is a message oriented Connection less Datagram protocol.

→ Unreliable Transfer protocol.

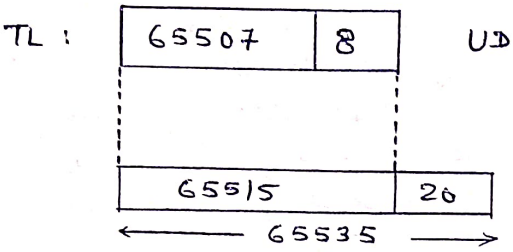
→ does not provide Err Control, flow Control and Congestion Control.

→ It does not add anything to the services except process to process delivery.

→ Header is simple and fixed in size i.e. 8byte



• Total Length :- (16 bit)



Max UDP data size = 65507 Byte

→ only those processes Sending short msg less than 65507 byte Can use UDP.

- checksum : (16 bit)

→ UDP Checksum includes 3 sections :

1) Pseudo Header (IP)
2) UDP Header
3) data Coming from AL.

IP Psudo Header for UDP

SIP (32 bit)		
DIP (32 bit)		
Reserved (8 bit)	Protocol (8 bit)	UDP total Length (16 bit)

Note :-

1) Unlike TCP, the checksum Calculation is not mandatory in UDP.
No Err and flow Control provided by UDP.
Hence UDP depends on IP and ICMP for Err Reporting.

↳ If UDP choose not to Calculate checksum, then
checksum = 0

Need of UDP :-

1) The application that required one request and one reply. TCP is not suitable. (E.g. - DNS)
2) Application that required Constant data flow, UDP is suitable.
3) Application that required multimedia data Transfer.
4) for Broadcasting and multicasting Application
5) Application that required fastness and then reliability.
6) UDP used for management process such as SNMP (simple N/W management process)
7) UDP used for some Route updating protocol as RIP.
8) UDP is normally used for Interactive real time applications
9) UDP is suitable for a process with internal flow and Err Control.
Ex- TFTP

TCP	UDP
⇓	⇓
Seq No ✓	No Seq No.
ACK No. ✓	No ACK Number
overhead ↑	overhead ↓
Keep track of order	No order
↓	↓
HTTP, FTP, SMTP, POP, IMAP	DNS, SNMP, TFTP, NFS, RIP, BOOTP, DHCP