CS & IT ENGINEERING

COMPUTER NETWORKS

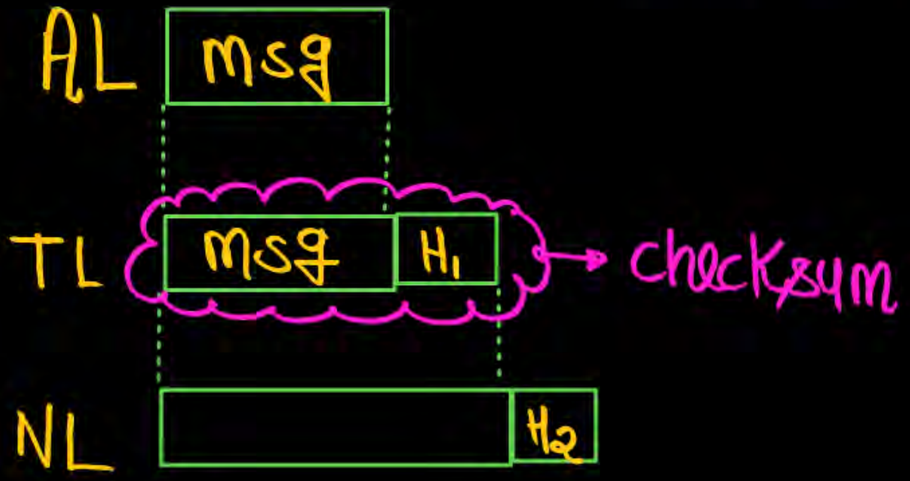
TCP & UDP

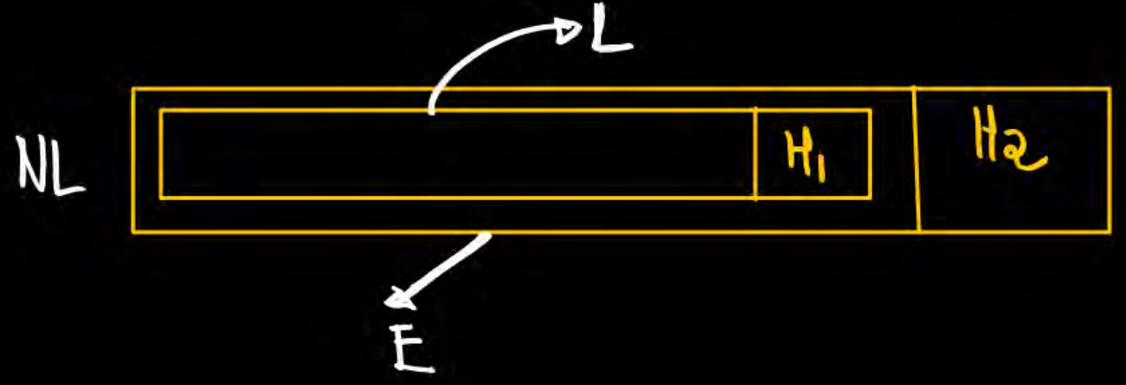
Lecture No-14

By- Ankit Doyla Sir

TCP Checksum

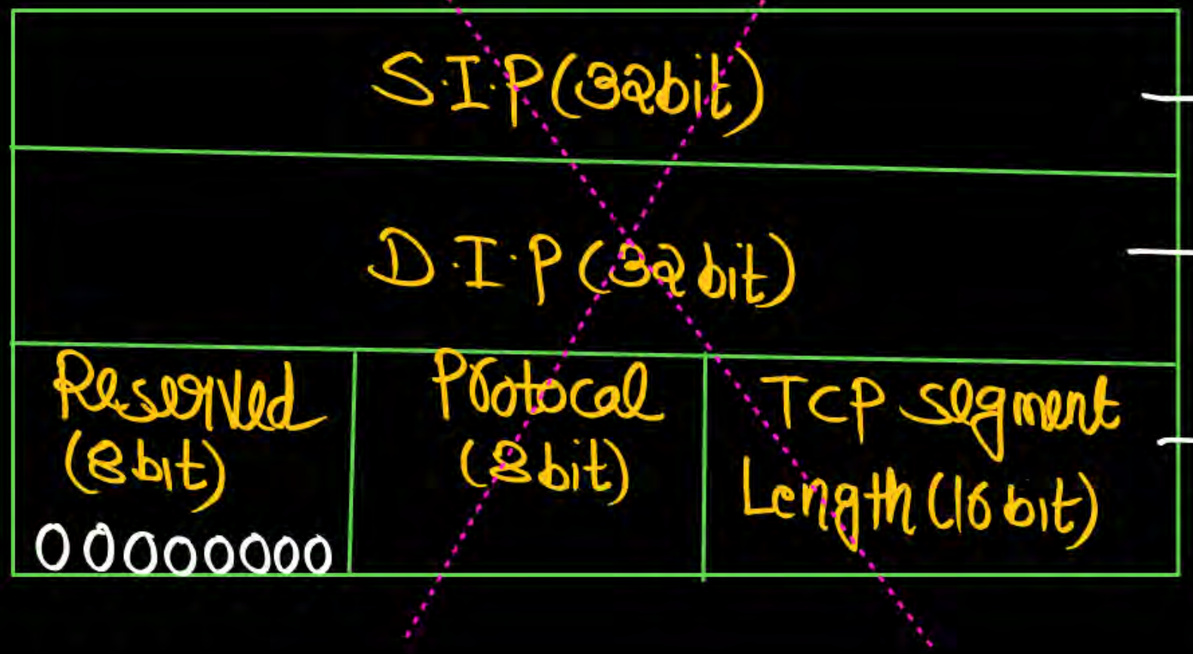
TCP Checksum





TCP checksum = TCP data + TCP Header + IP Pseudo Header

IP Pseudo Header

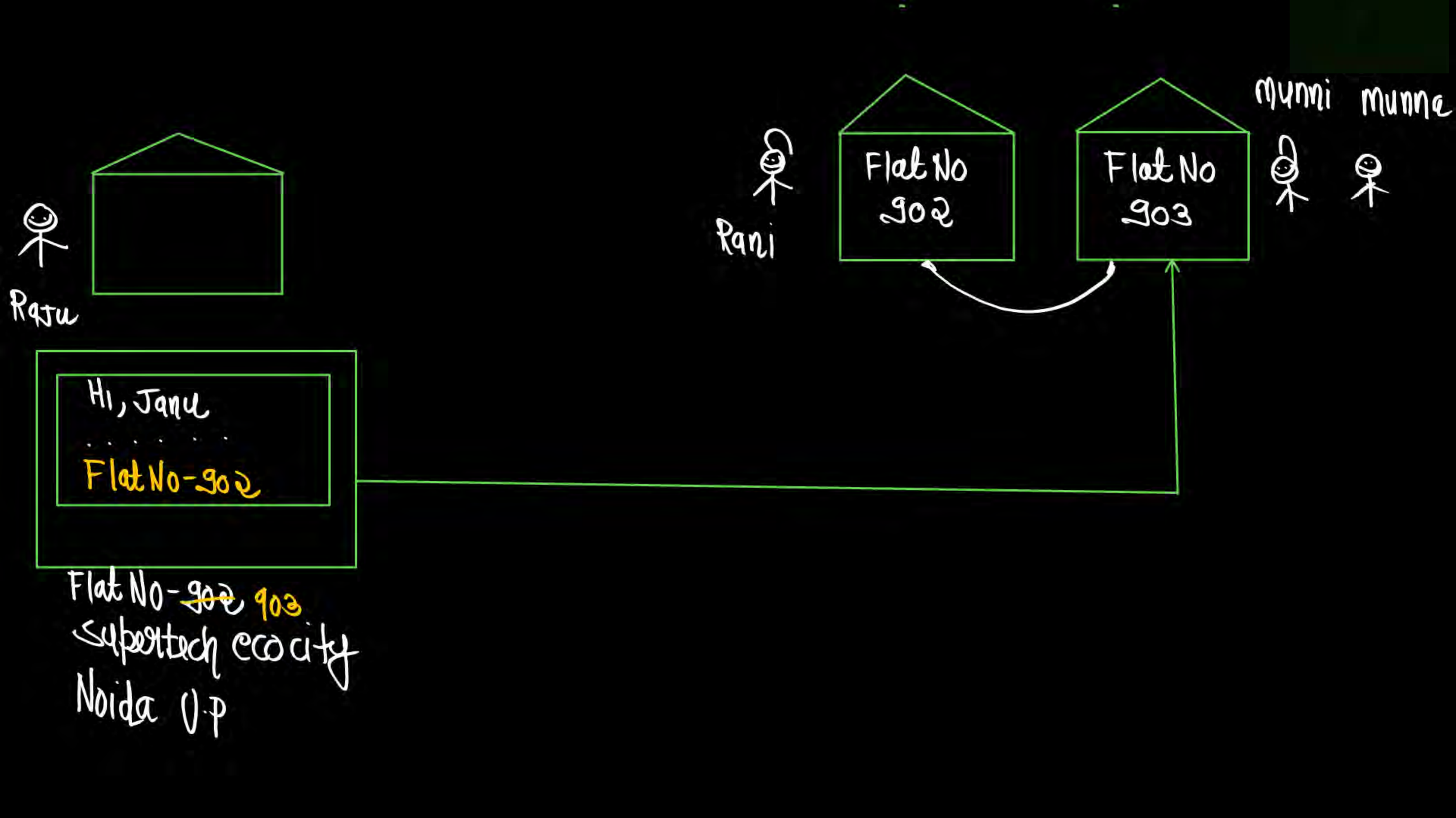


→ 4B

→ 4B

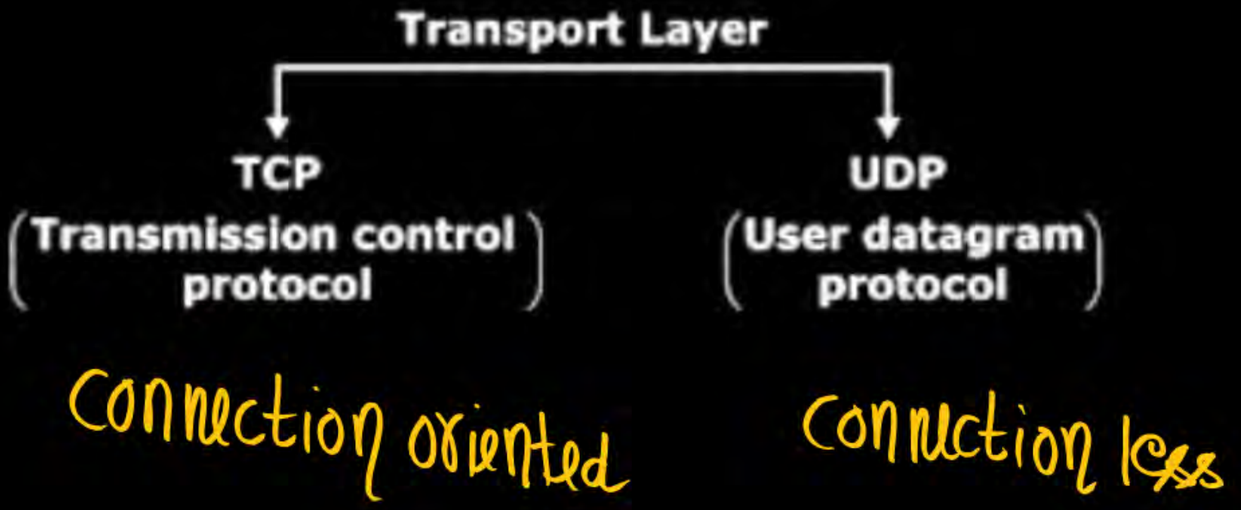
→ 4B

12Byte


Munni
Munna
Flat No 902
Flat No 903
Rani
Raju
Hi, Janu
. . . . . .
Flat No-902
Flat No- ~~902~~ 903
Supertech eco city
Noida U.P

Transport Layer Protocol

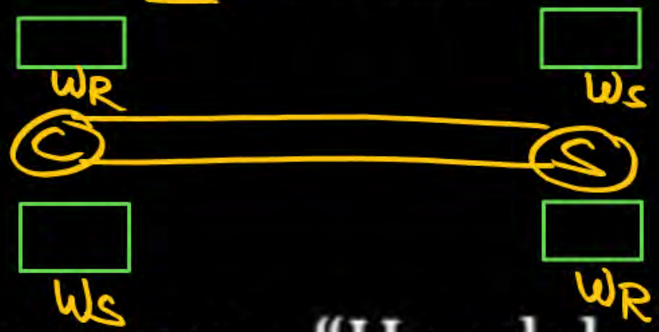
Transport Layer can be connection oriented or connection less.



Transport Layer Protocol

1. TCP is reliable process to process delivery of entire message.
2. TCP is a connection oriented.
3. TCP connection are full duplex and point to point.
4. TCP connection has 3 phases
    i. Connection establishment
    ii. Data transfer
    iii. Connection termination

5. Each TCP connection is associated with Four window.



6. TCP user three way "Handshake" to establish TCP connection.
7. TCP is not useful for Broadcasting and Multicasting.
8. TCP Header size is 20 byte but if options are added it will become 60 byte.
9. TCP provide end to end error control and flow control.
10. Data will be received at the destination in order.

Out of order segment:

TCP implements today do not discard out of order segment. They stored them temporarily and flag them as out of order until the missing segment arrive.

Note, however, that out of order segment never delivered to process.

TCP guarantee that data are delivered to the process In order.

NOTE:-

Data may arrive out of order and be temporarily stored by receiving TCP, but TCP guarantee that No out of order data delivered to the process

UDP

1. UDP is message oriented connection less Datagram protocol.
2. It is unreliable Transport protocol.
3. It does not provide Flow control and Error control & congestion control
4. It does not add anything to the services except process to process delivery of data.
5. Header is simple and fixed in size i.e. 8 byte

UDP Header


Source port (16 bit)
Destination port (16 bit)
Length (16 bit)
Checksum (16 bit)
0000000000000000
32 bit → 4B
32 bit → 4B
8Byte

Source port Address

This is a 16 – bit field that defines the port number of the application program in the host that is sending the segment.

Destination port address

This is a 16 – bit field that defines the port number of the application program in the host that is receiving the segment.

Total length = 16 bit

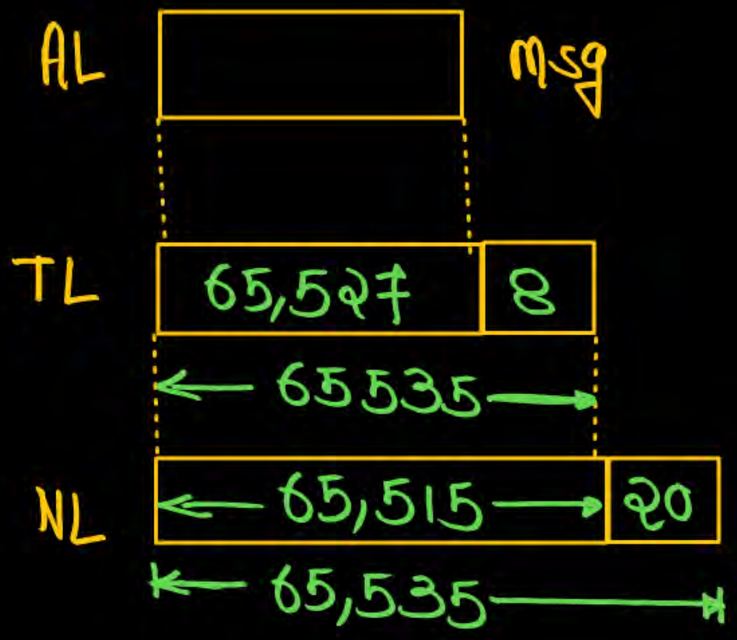
Maximum Number $\longrightarrow 2^{16} - 1 = 65,535$

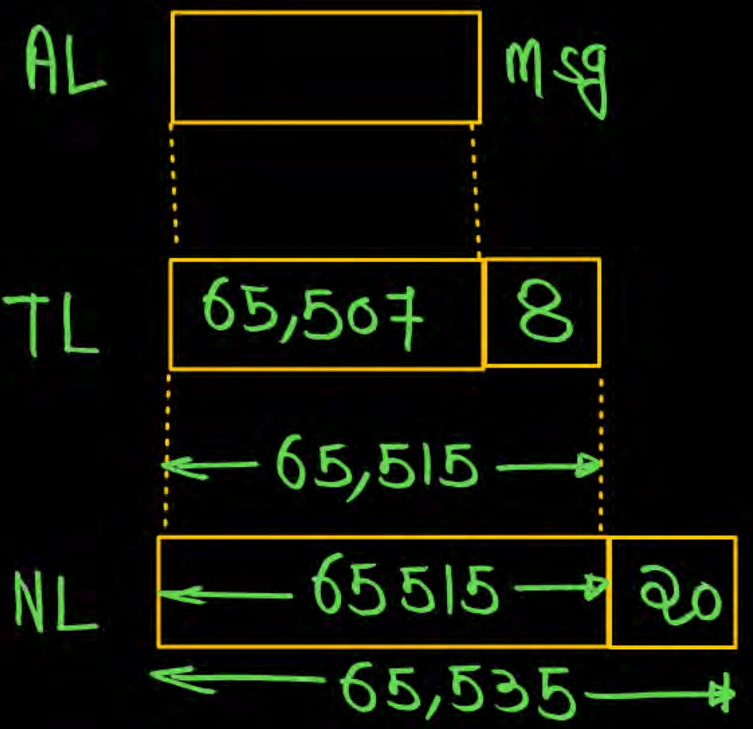
Total Length = Data + Header

65,535 = Data + 8

Maximum UDP data size = 65,535 − 8

Maximum UDP data size = 65,527


AL
msg
TL
65,527
8
65535
NL
65,515
20
65,535


AL
msg
TL
65,507
8
65,515
NL
65515
20
65,535

NOTE:-

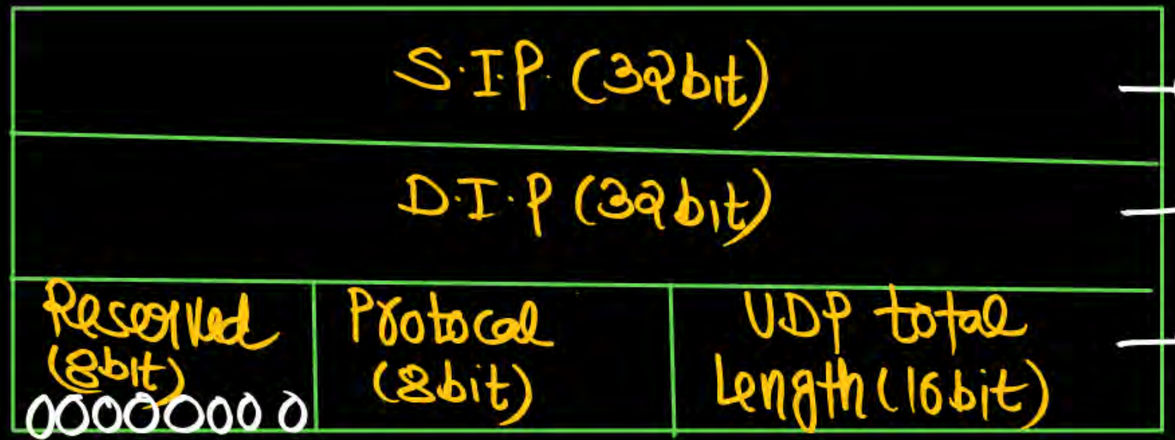
1. Practically data size at Transport layer for UDP is maximum 65,507 Byte
2. Only those processes sending short message less than 65,507 byte (65535 minus 8 byte for UDP header and minus 20 byte for IP header) can use UDP.

Checksum (16bit)

UDP Checksum includes three sections: a pseudo header, UDP header , and the data coming from application layers

UDP checksum=UDP data + UDP header +IP pseudo header

IP Pseudo Header



Q. The following is the content of a UDP header in hexadecimal format.

CB84 | 000D | 001C | 001C

S.Port | D.Port | TL | checksum

S.Port: 13 (Well known → Server)

D.Port: 52,100 (client)

a. What is the source port number ? Ans: 52,100

b. What is the destination port number ? Ans: 13

c. What is the total length of the user datagram ? Ans: 28 Byte

d. What is the length of the data ? Ans: 20 Byte

e. Is the packet directed from a client to a server or vice versa ?

Ans: Packet is moving From client to server

D.Port = 13 (well know Port No [0-1023]) → server

S.Port = $(CB84)_{16}$

$16^3\ 16^2\ 16^1\ 16^0$

$= 12*16^3 + 11*16^2 + 8*16^1 + 4*16^0$

$= 52100$

D.Port = $(000D)_{16}$

16^0

$= 13*16^0 = 13$

Total length = $(001C)_{16}$

$16^1\ 16^0$

$1*16^1 + 12*16^0 = 28$

Total length = Data + Header

data = Total length − Header

$= 28 - 8$

data = 20 Byte

NOTE:-

1. When destination port number is well known port number then data is moving from client to server.
2. When Source port number is well known port number then data is moving from server to client

Note:-

Unlike TCP, the checksum calculation is not mandatory in UDP. No error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting

Optional inclusion of checksum:

The sender of UDP packet can choose not to calculate the checksum. In this case the checksum field is filled with all 0s before being sent.

Q. What value is sent for the checksum in each one of the following hypothetical situations?

1. The sender decides not to include the checksum.
2. The sender decides to include the checksum, but the value of the sum is all 1s.
3. The sender decides to include the checksum, but the value of the sum is all 0s.

Solution:

1. The value sent for the checksum field is all 0s to show that the checksum is not calculated.
2. When the sender complements the sum, the result is all 0s; the sender complements the result again before sending. The value sent for the checksum is all 1"s. The second complement operation is needed to avoid confusion with the case in part 1.
3. This situation never happens because it implies that the value of every term included in the calculation of the sum is all 0s, which is impossible; some fields in the pseudoheader have nonzero values.

Need of UDP

Why UDP ?

I. The application that required one request one reply. TCP is not suitable hence we use UDP. DNS

II. Application that required constant dataflow TCP is not suitable hence we use UDP.

III. Application that required multimedia data transfer we can not use TCP hence we use UDP.

IV. Application that required fastness and then reliability TCP is not suitable hence we use UDP.

V. UDP used for management process such as SNMP (simple N/w management protocol)

Why UDP ?

VI. UDP is used for some route updating protocol such as RIP

VII. For broadcasting & multicasting application TCP is not suitable hence we use UDP

VIII. VIII. UDP is normally used for interactive & real time applications

IX. IX. UDP is suitable for a process with internal flow –and error control mechanisms. For example, the Trivial File Transfer protocol(TFTP) process include flow and error control. It can easily use UDP

TCP

Vs

UDP

TCP	UDP
Dynamic Header(20-60 byte)	Fixed header(8 byte)
End to end Flow control	No flow control
Error control(Checksum mandatory)	No error control(Checksum is optional)

TCP	UDP
Connection-oriented	Connectionless
Reliability in delivery of msg	Not reliable
Sequence Number.	No sequence number.
Ack no.	No ack no.
Overhead is high (20B - 60B)	overhead is less (8 Byte Header)
Keep track of order (sequence)	No order
Protocols: HTTP, FTP, SMTP, POP, IMAP	Protocol: DNS, SNMP, TFTP, NFS,RIP ,BOOTP ,DHCP , All real time and multimedia protocols

Note:-

Client server application such as DNS uses the services of UDP because a client need to send a short request to server and to receive a quick response from it. The request and response can each fit in one user datagram. Since only one message is exchanged in each direction.

Note:-

A client-server application such as SMTP, which is used in electronic mail, cannot use the services of UDP because a user might send a long e-mail message, which could include multimedia (images, audio, or video). If the application uses UDP and the message does not fit in one user datagram, the message must be split by the application into different user datagrams. Here the connectionless service may create problems. The user datagrams may arrive and be delivered to the receiver application out of order. The receiver application may not be able to reorder the pieces.

THANK YOU!