

CS & IT ENGINEERING

COMPUTER NETWORKS

TCP & UDP

Lecture No-17

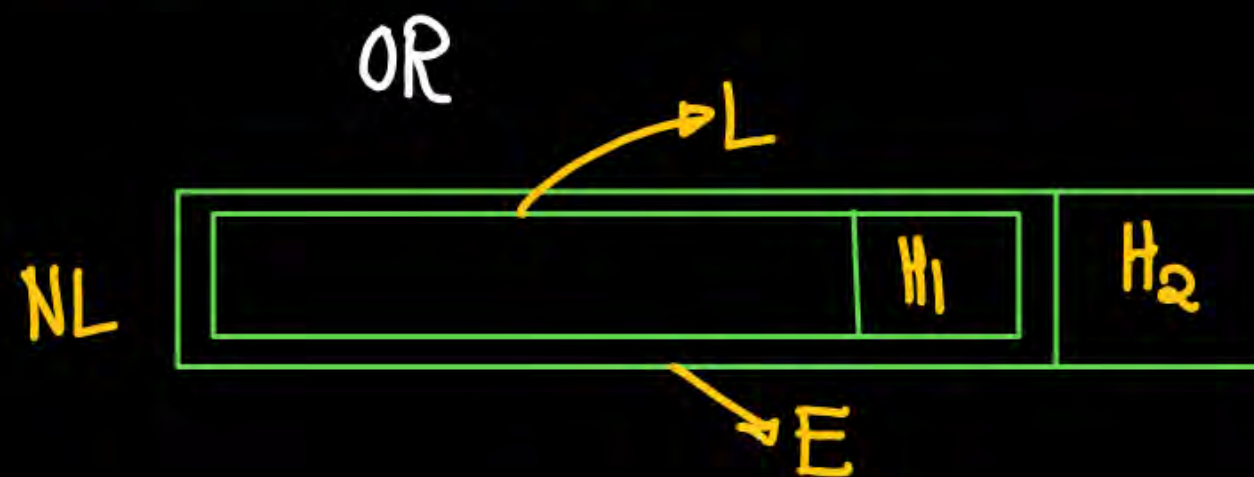
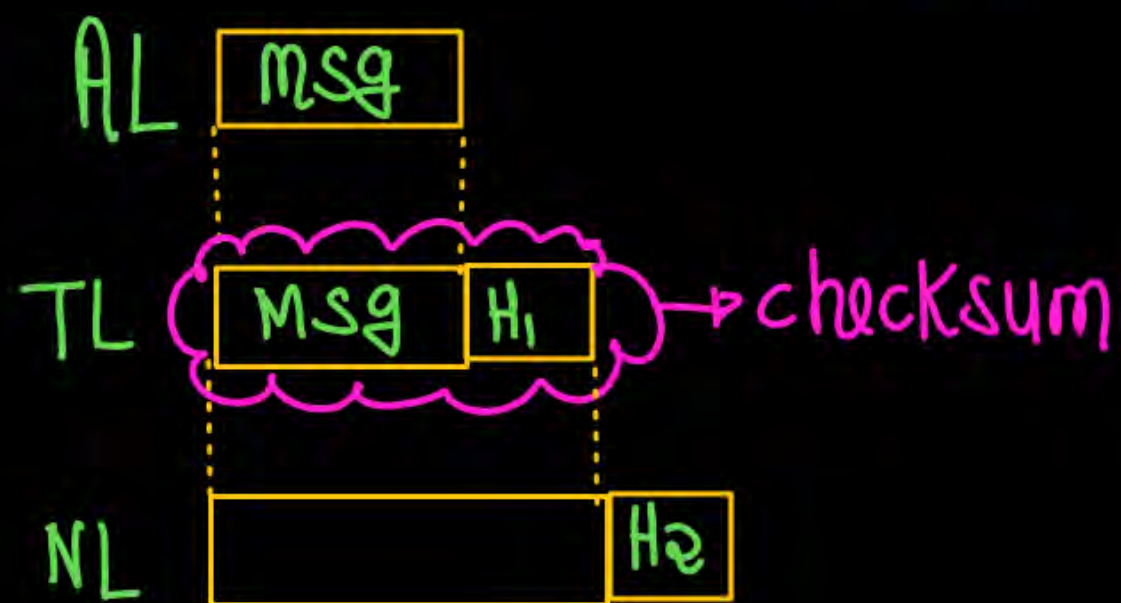


By- Ankit Doyla Sir

TOPICS TO
BE
COVERED

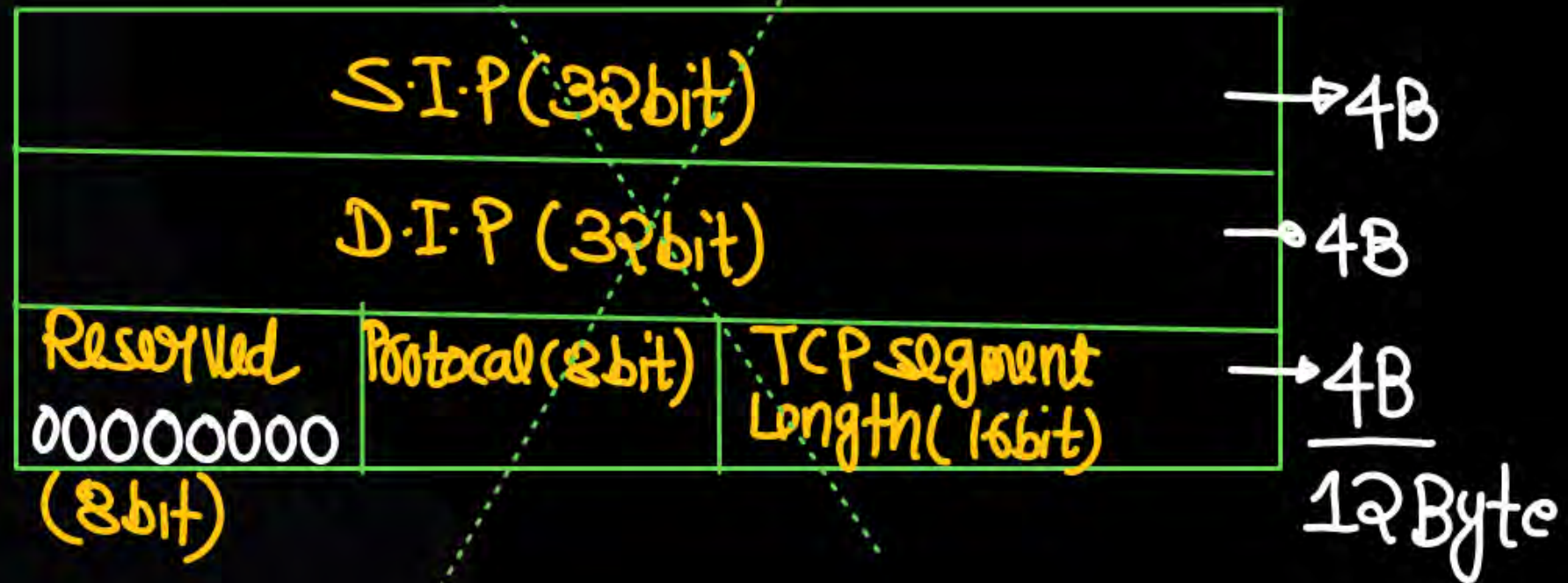
UDP HEADER

TCP Checksum



$$\text{TCP checksum} = \text{TCP data} + \text{TCP Header} + \text{IP Pseudo Header}$$

IP Pseudo Header



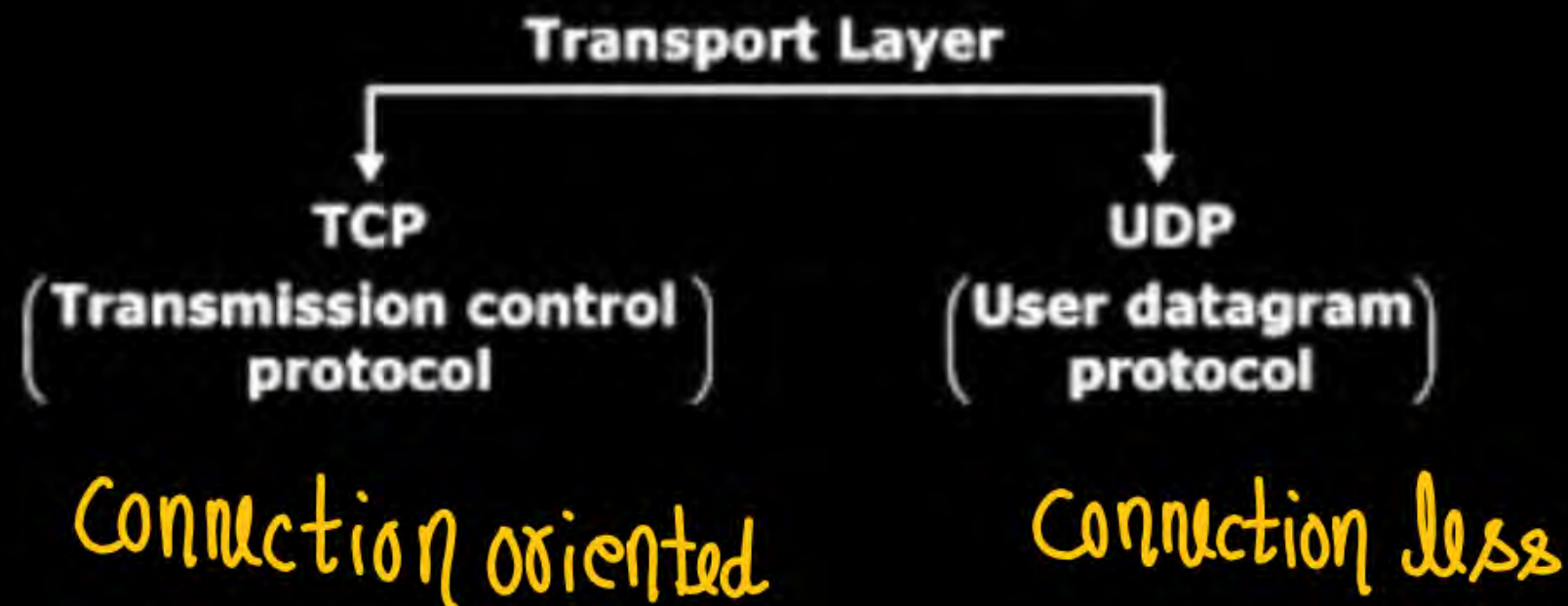


Hi, Janu . . .
.....
Flat No-902

Flat No-~~902~~ 903
Sakettech Ecocity
Sector 137, Noida

Transport Layer Protocol

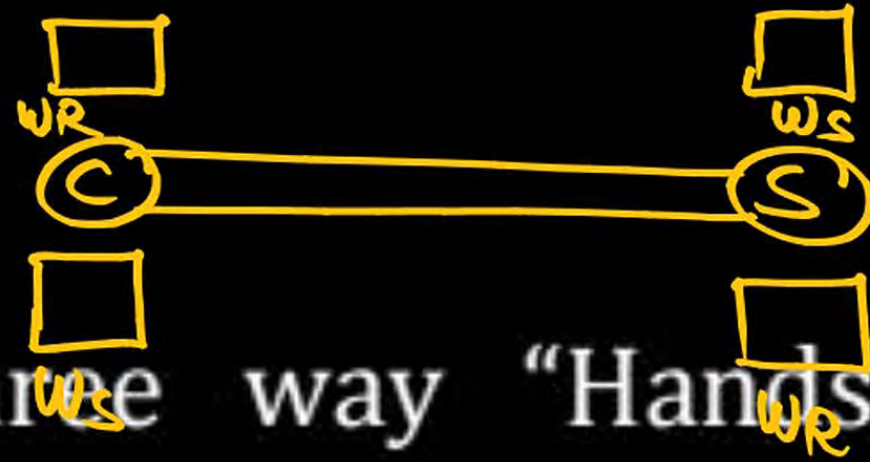
Transport Layer can be connection oriented or connection less.



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.

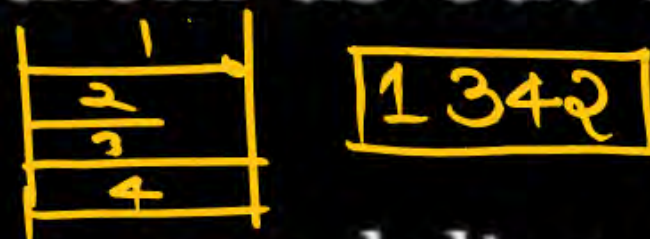
8. TCP Header size is 20 byte but if options are added it will become 60 byte.

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 data delivered to the process

^
order

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.

$$\text{Total length} = 16 \text{ bit}$$

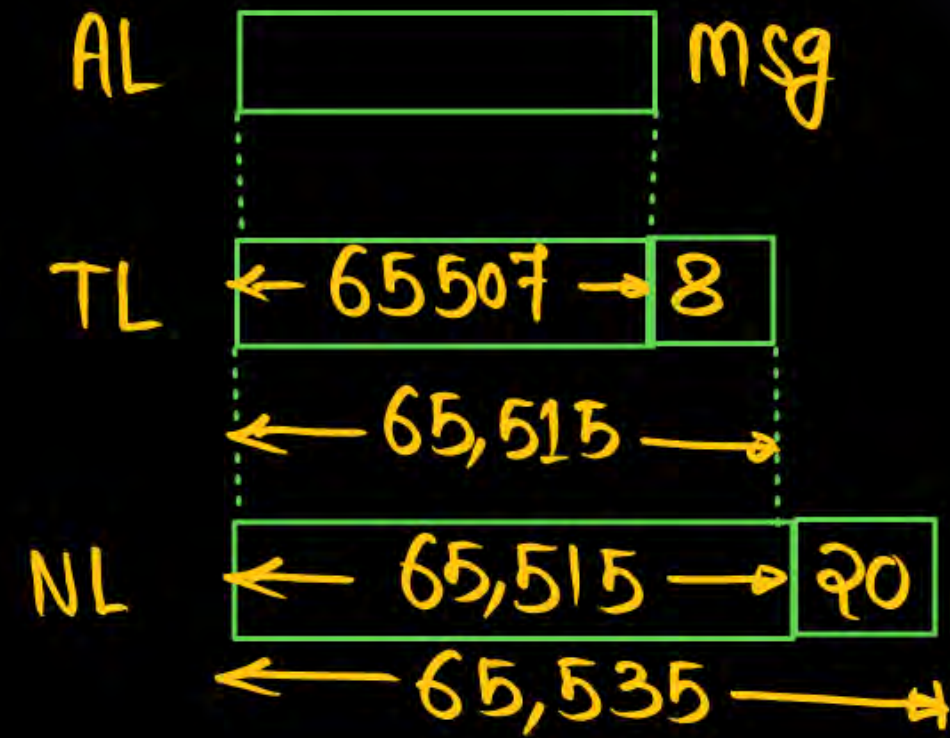
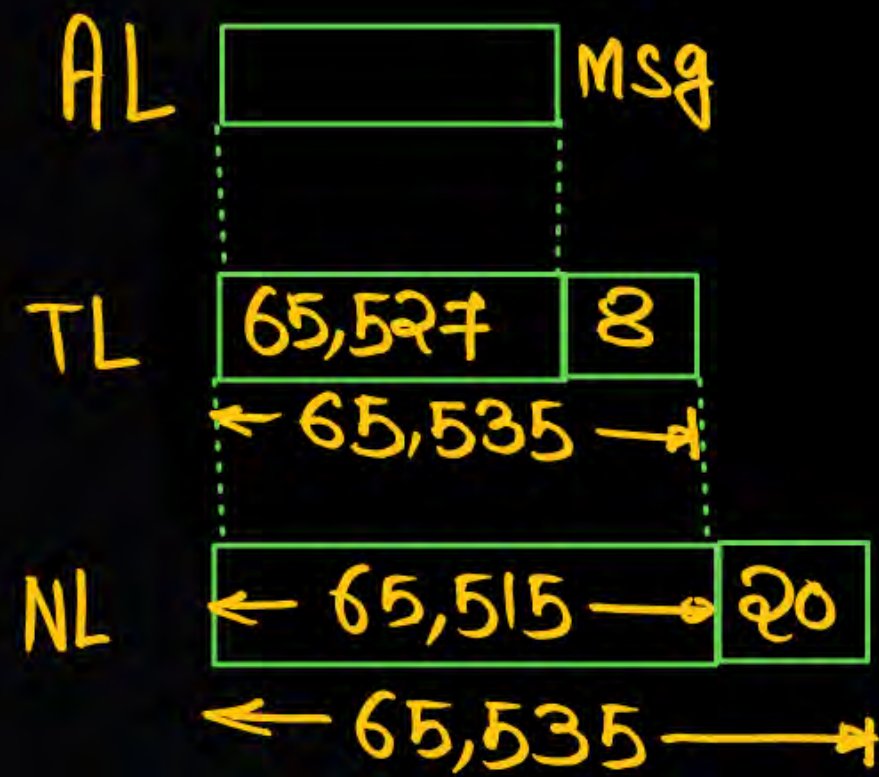
$$\text{Maximum Number} = 2^{16} - 1 = 65,535$$

$$\text{Total length} = \text{Data} + \text{Header}$$

$$65,535 = \text{Data} + 8$$

$$\text{Max UDP data size} = 65535 - 8$$

$$\text{Max UDP data size} = 65,527$$



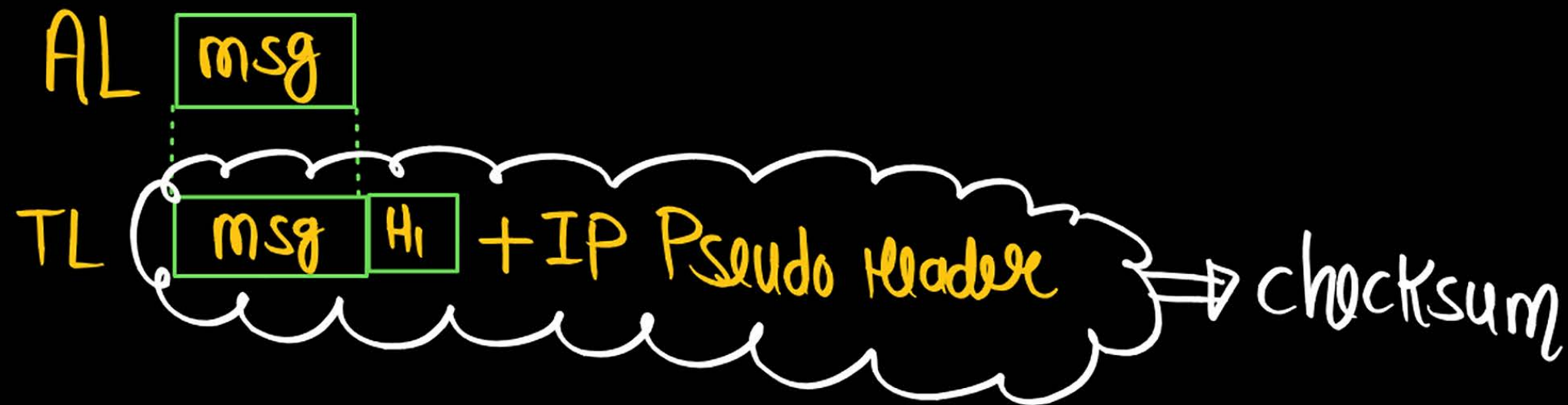
NOTE:-



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

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



S.I.P (32bit)			→ 4B
D.I.P (32bit)			→ 4B
Reserved (8bit)	Protocol (8bit)	UDP total length (16bit)	→ 4B
00000000			<u>12B</u>



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

S. Port	D. Port	TL	checksum
CB84	000D	001C	001C

- 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
- 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 ?

Dest Port No = 13 (Well known Port No) \Rightarrow 20-1023
 \downarrow
(Server)
Packet is moving From client to server

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

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

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

$$= 52,100$$

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

16^0

$$= 13 \times 16^0 = 13$$

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

$16^1 \quad 16^0$

$$= 1 \times 16^1 + 12 \times 16^0$$

$$= 16 + 12 = 28$$

$$Total\ length = Data + Header$$

$$28 = Data + 8$$

$$Data = 20\ Byte$$

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.
- 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	overhead is less
Keep track of order (sequence)	No order
Protocols: HTTP, FTP, SMTP, POP	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!**

