Assignment -1

Name = Subu Kumar Jaiswal

Roll No = 22054412

Sec = CSE-34

Subject = CN

Q1) a)

⇒ Given UDP header hexadecimal = CB84 0050 001F 001F

(Source, Destination, length)

I) - length of data = $(001F)_{16} = (31)_{10}$

$= 31 - 8$

$= 23$

$\boxed{\text{length of data} = 23}$

II) - Destination of port number = $(0050)_{16}$

$(0050)_{16} = 80$

$\boxed{\text{port number} = 80}$

b) Solution,

given :-

bandwidth = 512 mbps = 512×10^3 bps

propagation speed = 2×10^5 m/s

length of packet = 1kb = $8 \times 1024 = 8192$ bits

between 2 computers x and y.

$\Rightarrow$ Transmission delay $= \frac{L}{B} = \frac{\text{Length}}{\text{bandwidth}}$

$\Rightarrow$ propagation delay $= \frac{d}{V} = \frac{\text{distance}}{\text{propagation speed.}}$

$$T_t = \frac{8192}{512 \times 10^3} = \frac{8192}{512000} \simeq 0.016 \text{ sec.}$$

$$\boxed{\text{Transmission delay} = 16 \text{ ms.}}$$

assuming propagation delay = Transmission delay.

$$T_p = T_t$$

$$\frac{d}{v} = 0.016$$

$$\text{distance } (d) = v \times 0.016$$

$$d = 2 \times 10^9 \times 0.016$$

$$d = 32 \times 106 \text{ m} \Rightarrow \boxed{d = 3200 \text{ km}} \text{ :ans}$$

c) solution

TCP, header uses a 32-bit sequence number for given segment : ACK, SYN+ACK, FIN

TCP uses a 32-bit sequence number field for:

i) Uniquely Identify each bytes of data in a TCP - connection.

ii) provide a large enough sequence number space to avoid sequence number wrap around during a single connection.

Given segments :-

i) ACK - Consumes no sequence number, only Acknowledges the received data.

ii) SYN + ACK : consumes one sequence number for the SYNC flag and acknowledges the received SYN segment.

iii) FIN : consumes one sequence number, indicating the end of data stream.

⇒ All these and some others flags takes a lot of space in the TCP-header, that's why the TCP-header is of 32-bit.

d) ⇒ Given TCP header length field value of 0111

$(0111)_2 = (7)_{10}$

$\boxed{\text{TCP header length} = 7 \times 4 = 28 \text{ bytes.}}$

⇒ optional data = Actual header length - standard header length

$= 28 - 20 = 8$ bytes.

optional data in bits $= 8 \times 8 = 64$ bits.

$\boxed{\text{optional data} = 64 \text{ bits.}}$

c) Solution,

Go-Back-N protocol with $m=3$

window size $= 7$

$Sf = 62$, $Sn = 66$, $Rn = 64$

Sequence Number $= 2^m = 2^3 = 8$

[Sequence Number $= 0, 1, 2, 3, 4, 5, 6, 7$]

Go back-N $= 2^m - 1 = 2^3 - 1 = 8 - 1 = 7$.

$\therefore$ $Sf = 62 \rightarrow$ sequence from.

$Sn = 66 \rightarrow$ Next seq. number.

from 62 to 66

$\boxed{\text{Seq number} = 62, 63, 64, 65, 66}$

Q2

A) $\Rightarrow$ Given successive visits are : RTT of RTT2 ... RTT_n

... ANS Lookup Time = if n [DNS] Servey are visited and each visit incurs a round-trip [RTT]

Total DNS time : $RTT1 + RTT_2 + RTT_3 \ldots + RTT_n$

to calculate the total time elapsed.

Total time elapsed = ANS Lookup time + web page Retrieval time.

$$\boxed{\text{Total time} = \sum_{i=1}^{n} RTTi + RTT_0}$$

i) DNS lookup time : total time for the DNS lookup is the sum of the round time trip for each DNS visited.

$$\text{DNS lookup time} = RTT_1 + RTT_2 + \cdots + RTT_n$$
$$= \sum_{i=1}^{n} RTT_i$$

ii) web pages Retrieval time = after DNS lookup the ip-address obtained the time taken to retrieval the web page object

$$\boxed{\text{web page Retrieval time} = RTT_0}$$

$$\boxed{\text{total time} = \sum_{i=1}^{n} RTT_i + RTT_0}$$

B)

Ans) ⇒ Given protocol are :-

i) SMTP - simple mail Transfer protocol
ii) MIME - Multipurpose Internet mail Extensions.
iii) POP 3 - post-offic - protocol version -3

i) SMTP = protocol used for sending email, across networks. It operates over the TCP/IP protocol suite and is defined in several RFCs with RFC being the most notable, meaning it is it is a push protocol, meaning it is designed to send message from a client to server or among servers.

ii) MIME :- this protocol enhanced the functionality of email-system by enabling the exchange of diverse content types.

iii) POP 3 :- protocol used by email clients to retrieve emails from a group servers, used by client and scane email.

Q3

I) ⇒ Given a selective Repeat protocol with
sender window size = 4
Receiver window size = 4
sequence number : 0 to 15 (total 16 seq no)
packets are sent up to 10th seq ence no
Acknowledgement Received : sent upto 7 number.

I) sender's and receiver's window position.

i) sender window :- The sender has sent packets with seq number upto to Acknowledgemend have been received for packets 0 to 7.

ii) Receiver = The receiver has received packets upto a correctly but has least packet 8. The receive can accept packets with sequence no 10, 11, 12 and 13.

II) Response to packet loss and receiver has received up to sequence number 9 but packet 8-lost.

- Sender Response

⇒ i) Sending new packets: The sender can continue to send packet 8, 9, 10 and 11 since it has received all acknowledgement for packet.

ii) wait for Acknowledgement: The sender will wait for acknowledgement for packets 8, 9 and any subsequent packet sent (10, 11). It will not know that packets 8 is missing until it receiving a negative acknowledgement [NAK].

B)

⇒ selective Repeat protocol max window size:

sender : receiver

2^{m-1} : 2^{m-1}

Go-Back-N protocol max window size:

$2^m - 1$: 1

i) Selective Repeat protocol: This protocol allows both the sender and receiver to maintain a window size upto $(2^m - 1)$ and this smaller window size is necessary because selective Repeat [SR] can accept out-of-order frames and only send back the acknowledgement.

Examp:- if m=3 [seq no = 0 to 7] and [sender window = 4] it can send frame = 0, 1, 2, 3 now let us assume that [frame = 2] got the receiver can still accept frame 3 & frame 4 storing them and wait till [frame = 2] get's received.

ii) Go-Back-N protocol :- This protocol the sender multiple-frame before needing acknowledgements but the receiver can only accept frames in order. Example :-
If the sender has a window size of 3 and sends frame 0, 1, 2, but $\boxed{\text{frame} = 1}$ in lost, the receiver will discard frame 2 and 3. until it receives [frame = 1], thus the sender must retransmit frame starting from leading to potential in efficiency of transportation.

Q4) A) ⇒ A, TCP connection has been established between HOST A & HOST B

I) ⇒ The significance of SYN and ACK flag
⇒ Host-A initiates the connections by sending a TCP segment win [SYN] flag set to 1 and sequence number = 1000.
⇒ This is also known as 3-way Hand-shake, which establishes the connection and synchronizes the sequence number between HOSA and HOST-B.

II) After receiving HOST-B SYN+ACK segment, HOST-A would set set = 1201. as it indicates HOST-B Acknowledgement number in the SYN+ACK segment.

$\boxed{\text{Next sequence number} = 1201}$

III)
⇒ HOST A sends a TCP segment with [FIN=1] indicating its intention to close/terminate the connection.

⇒ HOST-B sends a response with ACK segment acknowledging Host-A termination request or Host-B can also sends its own FIN-flag=1 and send it to Host-A.

B) and) ⇒ TCP header fields typically ranges from 20-60 bytes.

* TCP header components:-

i) source-port → 16 bits.
ii) Destination port → 16 bits.
iii) seq number → 32 bits.
iv) Acknowledgement number [ACK] - 32 bits.
v) data offset → 4 bits.
vi) Reserved - 3 bits.
vii) flags → URG/ACK/PSH/RST/SYN/FIN - 9 bits.
viii) window size → 16 bits
ix) checksum → 16 bits.
x) urgent pointer → 16 bits.
xi) option → since it is option so, it is variable in length.

TCP - header

i) Source port number :- Allows the receiving host to know to know which application is sending the data.

ii) Destination port number :- similar to source port.

iii) Sequence number = Essential for recording segments at the receiving end and ensuring data is processed.

iv) Acknowledgement Number :- Specifies the next byte the sender expects to receive.

v) flags :- control flags -

↳ 1) - URG :- Urgent
2) - ACK :- Acknowledgement
3) - PSH :- Push
4) - RST :- Reset
5) - SYN :- Synchronize
6) - FIN :- Terminate Connection

These flags control the flow of data and manage the connection data.

vi) Urgent pointer :- Indicates the end of Urgent data, that needs to be transmitted.

⇒ Data - section :- This is attached to the TCP - Header and contains the actual data that needs to be transmitted the length of that section can vary depending upon the urgent data.

Q5

A) Given, Link speed = 1 Gbps.

RTT = 20ms = 0.02 sec.

packet size = 1000 - bytes

Bandwidth delay product (BDP) = Link speed × RTT.

$BDP = 1 \times 10^9 \times 0.02$

$\boxed{BDP = 2 \times 10^7}$

No of packets $= \frac{BDP}{packet\ size} = \frac{2 \times 10^7}{1000}$

$\boxed{No.\ of\ packets = 2 \times 10^4}$

To achieve 80% channel utilization, the window size should be atleast 80% of the BDP.

Maximum window size $= 0.8 \times 2 \times 10^4 = 16000$

$\boxed{\therefore Ans: 16000\ packets\ channy}$

B) Difference between :-

Centralized P2P network.	Decentralized P2P network.
i) Uses a central server to coordinate the communicate between peers.	i) peers communicate directly with each other without central server.
ii) peers connects to the central server	ii) peer uses a distributed search mechanism.
iii) if central server fails the entire network fails.	iii) If a peer leaves the network, it only affects the resources it was providing.

⇒ given DHT [Distributed Hash Table]

m = 4 (bit identifier space)

node identifier = 18

binary equivalent $(18)_{10} = (10010)_2$

⇒ The identifier space is divided into $2^4 = 16$ equal segment.

- The identifier space can be divided as follows:

- Segment 0: 0000 - 0111 [0-7]
- Segment 1: 1000 - 1111 [8-15]

Since the hash identifier 18 (10010) falls in the second segment (1000-1111) the node with the identifier closest to (18) - will be responsible for storing the resource.

⇒ Centralized P2P - networks rely on the central server for coordination, while decentralized network distribute control among peers. In a DHT - based networks with [m=4], node with hash identifier = 18, would be located in the second segment of the identifier space.