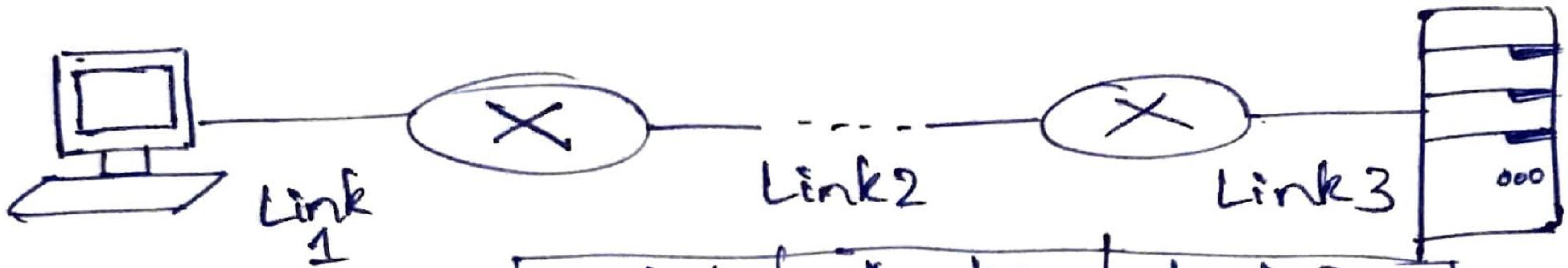
CS358 Midsem Assignment

Name: M. Maheeth Reddy
Roll: 1801CS31
Date: 25-02-2021

Answers

Ans1



	Link 1	Link 2	Link 3
Transmission Rates	1000Mbps	10Mbps	10Mbps
Link Length	1Km	1000Km	2Km

Mbps → Megabits per second

Given,

Length of packet = 8000 bits

Speed of Light propagation delay = $3 \times 10^8 \, ms^{-1}$

Transmission Delay (T_d) ~~is Amount~~ $= \frac{L}{R}$

L → length of packet R → Transmission Rate

Propagation Delay$(P_d) = \frac{d}{s}$

d → distance b/n routers s → speed of propagation

Total Delay at a Link $= (T_d + P_d)$ at the link.

(a) $$\text{Total Delay} = \left.\begin{matrix}\text{Total Delay}\\ @\text{ Link }1\end{matrix}\right\} + \left.\begin{matrix}\text{Total Delay}\\ @\text{ Link }2\end{matrix}\right\} + \left.\begin{matrix}\text{Total Delay}\\ @\text{ Link }3\end{matrix}\right\}$$

$$= (T_d + P_d)_{Link1} + (T_d + P_d)_{Link2} + (T_d + P_d)_{Link3}$$

$$= \left(\frac{L}{R}\right)_{L1} + \left(\frac{d}{s}\right)_{L1} + \left(\frac{L}{R}\right)_{L2} + \left(\frac{d}{s}\right)_{L2} + \left(\frac{L}{R}\right)_{L3} + \left(\frac{d}{s}\right)_{L3}$$

$$\Rightarrow \text{Total Delay} = \frac{8000}{1000\times10^6} + \frac{1\times10^3}{3\times10^8} + \frac{8000}{10\times10^6} + \frac{1000\times10^3}{3\times10^8} + \frac{8000}{10\times10^6} + \frac{2000}{3\times10^8}\ s$$

All values taken from above table in Pg 1

$$= 1.133\times10^{-5}\,s + 4.133\times10^{-3} + 8.067\times10^{-4}\,s$$

$$\Rightarrow \boxed{\text{Total Delay} = 4.95103\times10^{-3}\,s}$$

(b) Total Delay of Link 1 = Transmission Delay at Link 1 + Propagation Delay at Link 1

$$= \frac{L}{R} + \frac{d}{s} = \frac{8000}{1000\times10^6} + \frac{1\times10^3}{3\times10^8} = 1.133\times10^{-5}\,s$$

$$\Rightarrow \boxed{\text{Total Delay of Link 1} = 1.133\times10^{-5}\,s}$$

(c) Transmission Delay of Link 2 = $\frac{L}{R}$ → Length of Packet, → Transmission Rate of Link 2

$$= \frac{8000}{10 \times 10^6} = \boxed{8 \times 10^{-4} s}$$

$$\boxed{\text{Transmission Delay of Link 2 is } 8 \times 10^{-4} s}$$

(d) Total delay of Link 2 =

Transmission Delay at Link 2
+ Propagation Delay at Link 2

$$= \left(\frac{L}{R}\right) + \left\{\frac{d}{s}\right\} = (8 \times 10^{-4} s) + \left\{\frac{1000 \times 10^3}{3 \times 10^8} s\right\}$$

$$= (8 \times 10^{-4}) + \{3.33 \times 10^{-3}\} s$$

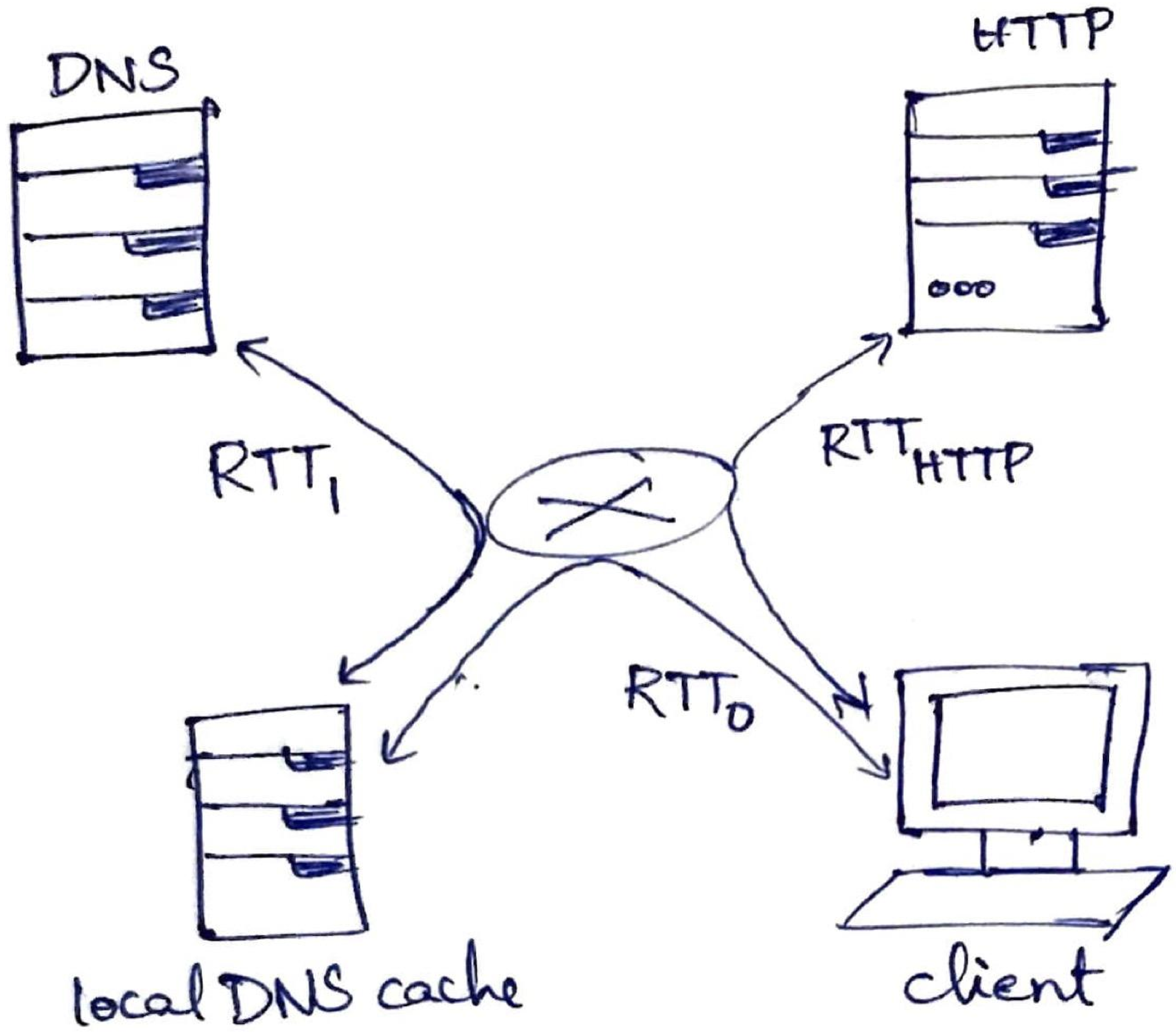
$$\Rightarrow \boxed{\text{Total delay of Link 2} = 4.133 \times 10^{-3} s}$$

(e) Transmission Delay of Link 3 = $\frac{L}{R}$ → Length of Packet, → Transmission Rate of Link 3

$$= \frac{8000}{10 \times 10^6} s = \boxed{8 \times 10^{-4} s}$$

$$\therefore \boxed{\text{Transmission Delay of Link 3 is } 8 \times 10^{-4} s}$$

Ans2



(a) Transmission time for HTML object is zero.

Time taken to receive object for the client
= Time to get IP address (DNS query)
+ Time to establish TCP connection
+ Time for response over TCP

Time to get IP address = Time delay @ local DNS cache
+ Time delay @ second DNS server

⇒ Time to get IP address = 4+31 ms = 35 ms

Time to establish TCP connection = RTT_{HTTP} = 94 ms

Time for response over TCP = RTT_{HTTP} = 94 ms

⇒ Total time taken for client to receive object

$= 35 + 94 + 94$ ms $= 223$ ms

∴ $\boxed{\text{It takes 223 ms for the client to receive object}}$

(b)

We know, time to fetch IPaddress $= RTT_0 + RTT_1$

$= \underline{35ms}$

Time elapsed from click on link until base object and all 6 additional objects are received

= Time to fetch IPaddress
+ Time to fetch base object by TCP
+ Time to fetch 6 additional objects

Time to fetch base object by TCP $= 2 \times RTT_{HTTP} = 2 \times 94$ ms

$= \underline{188 \text{ ms}}$

Time to fetch the 6 additional objects by TCP will be

$6 \times 2 \times RTT_{HTTP} = 12 \times 94$ms $= \underline{1128\text{ms}}$ since, this a non-persistent connection and handshakes will be done multiple times.

$\Rightarrow$ Total time taken $= (35 + 188 + 1128)$ ms
$= 1351$ ms

Non-persistent Serial HTTP

Hence, it takes 1351 ms from click on link until base object and all 6 additional objects are received.

(c) Time to fetch IP address $= RTT_0 + RTT_1$
$= \underline{35 ms}$

Time to get base object $= 2 \times RTT_{HTTP}$
$= 2 \times 94 ms = \underline{188 ms}$

We have 5 parallel TCP connections. To get 6 objects, we can establish TCP connection twice, in which 5 objects are delivered once and the 6^{th} object in 2^{nd} connection. Only 2 handshakes are enough

So, time to fetch 6 objects $= 2 \times 2 \times RTT_{HTTP}$
$= 2 \times 2 \times 94 = \underline{376 ms}$

Total time $= 35 + 188 + 376$ ms $= \underline{599 ms}$

$\therefore$ It takes 599 ms ~~to get~~ from click on link to getting all objects through ~~persistent HTTP~~ non-persistent parallel HTTP.

d) Time taken to fetch IP address $= RTT_0 + RTT_1$

$= \underline{35 ms}$

Time taken to fetch base object $= 2\times RTT_{HTTP}$

$= 2\times 94 ms = \underline{188 ms}$

For receiving 6 objects through 5 parallel connections, we could receive 5 objects in a shift and the 6^{th} object in the next shift.

Since, the connection is persistent, no more handshakes are required after receiving base object.

∴ Time to fetch 5 of the 6 objects $= RTT_{HTTP}$

$= 94 ms$

Time to fetch the remaining objects $= RTT_{HTTP}$

$= 94 ms$

∴ Total time $= 35 + 188 + 94 + 94 = \underline{411 ms}$

Hence, it takes 411ms from click on link to fetching all objects through persistent parallel HTTP e

Fastest method we've explored is Persistent Parallel.
This is evident from the delay calculations we did above.

In persisent HTTP connections, the connection is opened once and objects are received in a single GET response, unlike non-persistent HTTP connections, where we have to open the connection again and again to receive all objects.
Also, parallel connection allows transfer of multiple objects at same time. Hence, ~~the~~ we can explain why Persistent Parallel connections are faster.

Ans3 Given,

. Bandwidth is 64 Mbps (Megabits per second)

propagation speed = $\frac{2}{3} \times$ speed of light

$= \frac{2}{3} \times 3 \times 10^8 \, ms^{-1} = 2 \times 10^8 \, ms^{-1}$

propagation delay (Pd) $= \frac{d}{S}$ → d → distance b/n routers, S → speed of propagation

$= \frac{4000 \times 10^3}{2 \times 10^8} = \underline{\underline{20 \text{ ms}}}$

transmission delay (Td $= \frac{L}{R}$ → L → length of packet, R → transmission rate

$= \frac{8KB}{64Mbps} = \frac{\cancel{8} \times 8 \times 10^3}{\cancel{64} \times 10^6} = \underline{\underline{1 \text{ms}}}$

Assume sender window size is k
⇒ receiver window size = k

} ∵ Protocol is Selective Repeat.

$$\text{Efficiency} = \frac{n \times T_d}{T_d + 2 \times P_d} = \frac{n \times 1}{1 + 2 \times 20} = \frac{n}{41}$$

For full capacity of network (given), efficiency = 1

$$\Rightarrow \frac{n}{41} = 1 \Rightarrow \boxed{n = 41}$$

Hence sender & receiver window size = 41

Let sequence number has 't' no. of bits

$$\Rightarrow \text{sender window size} = 2^{t-1} = n$$

$$\Rightarrow 2^{t-1} = 41 \Rightarrow 2^t = 82$$

$$\Rightarrow t = \lceil \log_2 82 \rceil \Rightarrow \boxed{t = 7}$$

Hence, the minimum size of sequence number field is 7 bits

Ans 4.

Given, probability of packet error = 0.6

$\therefore$ probability of no-error packet $= 1 - 0.6$
$= \underline{\underline{0.4}}$

$\Rightarrow$ Out of every 'x' packets sent, $\underline{0.4x}$ are transmitted without error (on average)

Given, Avg. no. of transmission attempts required to transfer 'x' packets is 500

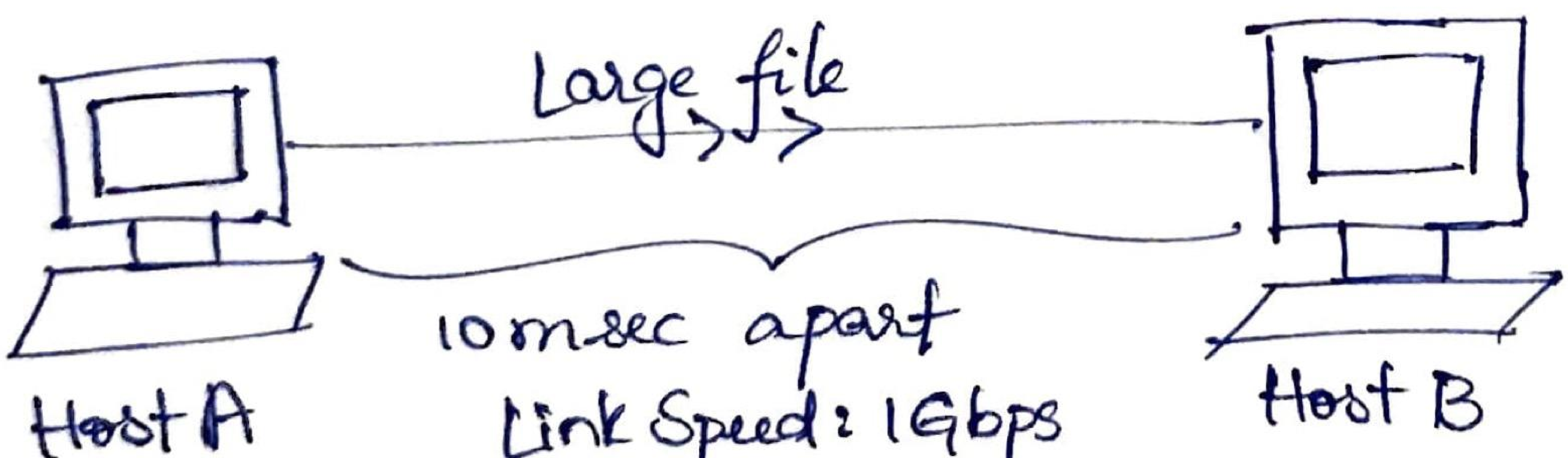
$$\Rightarrow 0.4x = 500$$

$$\Rightarrow \boxed{x = \frac{500}{0.4} = 1250}$$

$\therefore$ $\boxed{\text{Value of } x \text{ is } 1250}$

Ans 5.

Given, host A is sending host B a large file through TCP.



(a)

Bandwidth = 1Gbps

Total ~~numb~~ bits transferred in an RTT = 1Gbps × 20ms
$= 2 \times 10^7$ bits

Since utilization = 80%.

Total bits transfered in an RTT $= 0.8 \times 2 \times 10^7$

$= \underline{1.6 \times 10^7 \text{ bits}}$

Size of one packet = 1000 bytes

= 8000 bits

Window size is no. of packets in a round trip. ACK's not considered as their size is considered to be negligible.

$$\Rightarrow \boxed{\text{Window Size} = \frac{1.6 \times 10^7}{8000} = 2000 \text{ packets}}$$

Hence size of window is 2000 packets

(b) Since, there are no losses & no competing traffic,

If count of RTTs required to transfer a window is n,

then $2^{n-1} = 2000$

$\Rightarrow n - 1 = \lceil \log_2 (2000) \rceil \Rightarrow n = 1 + \lceil \log_2 (2000) \rceil$

$\Rightarrow \boxed{n = 12.}$

Hence, total time required = $n \times RTT = 12 \times 20\text{ms}$

$\Rightarrow$ total time required is 240ms

Ans 6.

HLEN represents the IP header length. It is the number of 32 bit words in the header.

Given that HLEN = 7 in an IPv4 datagram.

⇒ There are 7 32-bit words in header.

⇒ Size of header = 7 × 32 bits
= 7 × 4 bytes
= 28 bytes

Base header itself has 20 bytes

⇒ No. of option bytes = $\left\{\begin{array}{l}\text{Size of header}\\\text{(in bytes)}\end{array}\right\} - \left\{\begin{array}{l}\text{Size of base}\\\text{header}\end{array}\right\}$

= 28 − 20

= 8

Hence there are 8 option bytes

Ans 7 **Important Formulae**

$$\text{estimatedRTT} = (1-\alpha)^* \underset{\text{(pervious)}}{\text{estimatedRTT}} + \alpha^* \text{sample RTT}$$

$$\text{devRTT} = (1-\beta)^* \underset{\text{(previous)}}{\text{devRTT}} + \beta^* \left| \underset{\text{(previous)}}{\text{estimatedRTT}} - \text{sampleRTT} \right|$$

$$\text{TCP timeout} = \text{estimated RTT} + 4^* \text{devRTT}$$

Given

Current estimated RTT = 330ms
current dev RTT = 11ms

Next 3 RTT measured, values = 320 msec (first), 360ms (second), 300ms (third)

$\alpha = 0.125$

$\beta = 0.25$

Substituting above values in formulae

(a) Estimated RTT after first RTT

$= (1-0.125)^* 330 + 0.125^* 320$

$= \boxed{328.75 \text{ ms}}$

(b) RTT deviation ~~after~~ for first RTT

$= (1-0.25)^* 11 + 0.25^* |330-320|$

$= \boxed{10.75 \text{ ms}}$

(c) TCP timeout for the first RTT

= ~~estR~~ estimated RTT + 4* dev RTT

$= 328.75 + 4^* 10.75 = \boxed{371.75 \text{ ms}}$

(d) estimated RTT after second RTT

$$= (1-0.125)*\begin{Bmatrix}\text{estimated RTT}\\ \text{after } 1^{st}\text{ RTT}\end{Bmatrix} + 0.125*\text{second RTT}$$

$$= 0.875*328.75 + 0.125*360 \text{ ms}$$

$$= \boxed{332.65625 \text{ ms}}$$

(e) RTT Deviation after second RTT

$$= (1-0.25)*\begin{Bmatrix}\text{RTT Deviation}\\ \text{for first RTT}\end{Bmatrix} + 0.25*\left|\begin{matrix}\text{estimated RTT after}\\ 1^{st}\text{ RTT}\\ -\text{ sample RTT}\end{matrix}\right|$$

$$= (1-0.25)*10.75 + 0.25*|328.75-360|$$

$$= 0.75*10.75 + 7.8125$$

$$= \boxed{15.875 \text{ ms}}$$

(f) TCP timeout for second RTT

$$= \begin{Bmatrix}\text{estimated RTT}\\ \text{after } 2^{nd}\text{ RTT}\end{Bmatrix} + 4*\begin{Bmatrix}\text{devRTT}\\ \text{after } 2^{nd}\text{ RTT}\end{Bmatrix}$$

$$= 332.65625 \text{ ms} + 4*15.875 \text{ ms}$$

$$= \boxed{396.15625 \text{ ms}}$$

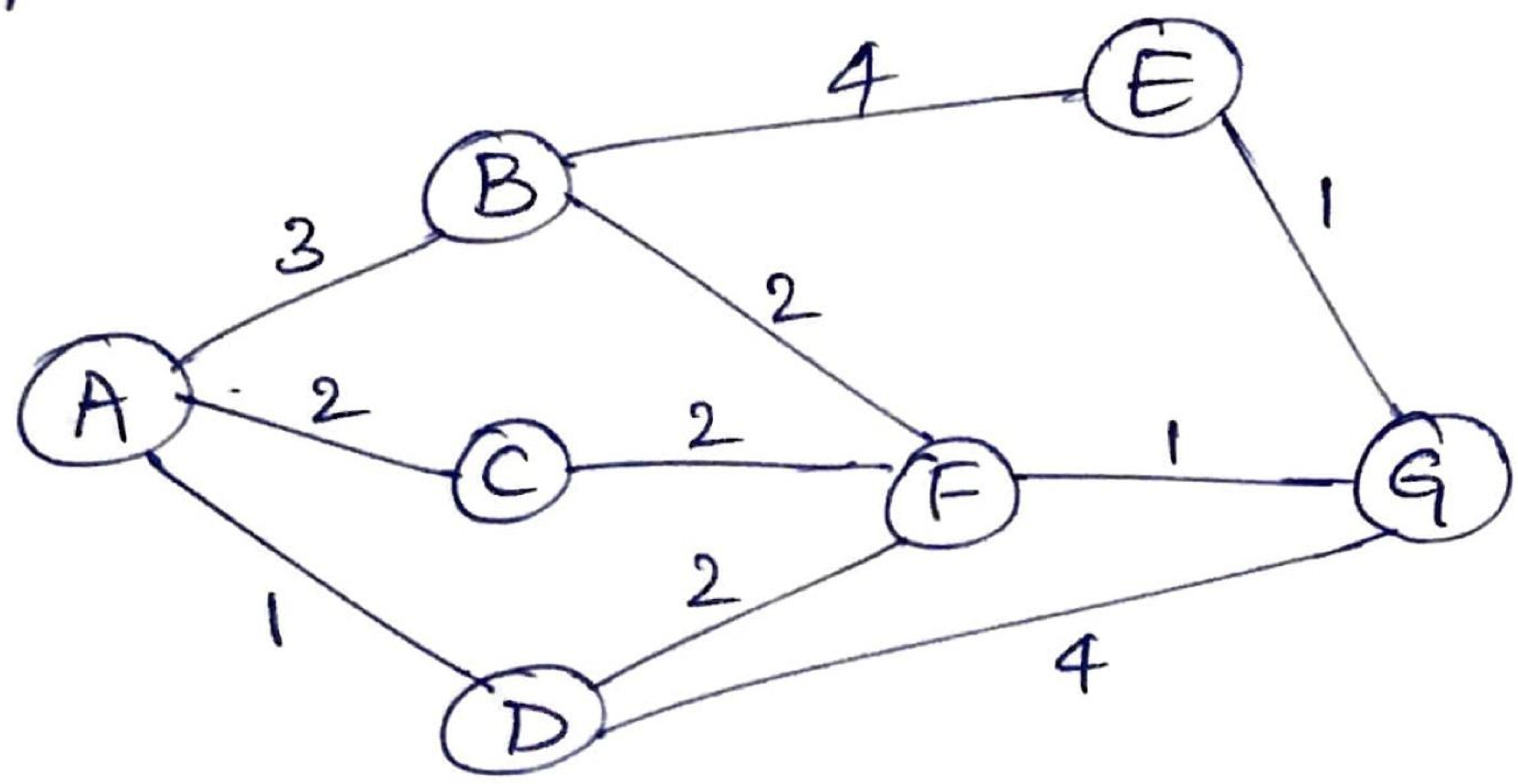
(g) estimated RTT after third RTT

$$= (1-0.125)*\begin{Bmatrix}\text{estimated RTT after}\\ 3^{rd}\text{ RTT}\end{Bmatrix} + 0.125*\text{third RTT}$$

$$= 0.875*332.65625 + 0.125*300 \text{ ms}$$

$$= \boxed{328.5742188 \text{ ms}}$$

Ans8 Given graph:



We are constructing a ~~graph~~ shortest path tree for node A using Dijkstra's algorithm.

First, we create set of nodes N'. This set is gradually filled with the nodes whose least-cost path is definitely known.

Since, the ~~short~~ least-cost-path for node A itself is known

$$\Rightarrow \quad N' = \{A\}$$

We maintain a table of current estimate of least-cost-path from A to every other vertex and the predecessor node along path from A to that vertex, as follows

Step	N'	B D(B), p(B)	C D(C), p(C)	D D(D), p(D)	E D(E), p(E)	F D(F), p(F)	G D(G), p(G)
0	A						

D(v), p(v) for a vertex v represent the current least-cost estimate and predecessor node to v along that path respectively.

For every vertex v that is adjacent to A in the graph $D(v)$ is initialized to $C_{A,v}$ or cost of edge b/n A, V & $p(v)$ is initialized to A

So, the table looks like this

Step	N'	B	C	D	E	F	G
0	A	3, A	2, A	1, A	∞	∞	∞

If a vertex v is not adjacent to A, $D(v) = \infty$ and $p(v)$ is not initialized.

① Now, D is a vertex for which $D \notin N'$ and $D(D)$ is minimum

② Hence add D to N'. ⇒ $N' = \{A, D\}$

③ update $D(v)$ for all v adjacent to D and $v \notin N'$ by

$$D(v) = \min(D(v), D(D) + C_{v,D})$$

also $p(v) = D$ if $D(D) + C_{v,D}$ is minimum

⇒ Table is updated as shown

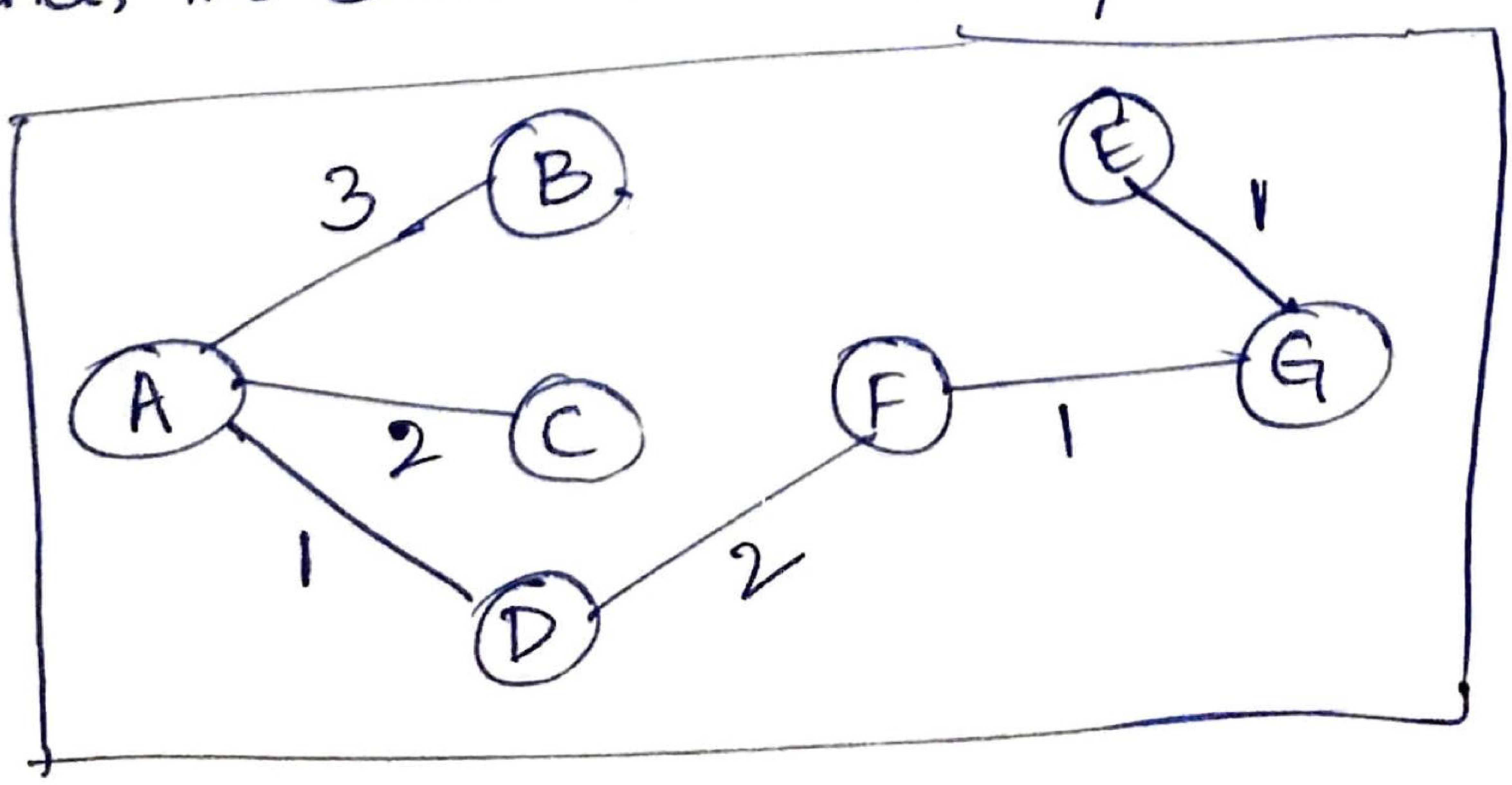
Step	N'	B	C	D	E	F	G
0	A	3, A	2, A	1, A	∞	∞	∞
1	AD	3, A	2, A		∞	3, D	5, D

Since, F and G are closer to A through D than they were directly through A, we updated the table as above.

On repeating steps 1, 2, 3 for rest of the vertices we get the table as follows

Step	N'	B	C	D	E	F	G
0	A	3,A	2,A	1,A	∞	∞	∞
1	AD	3,A	2,A		∞	3,D	5,D
2	ADC	3,A			∞	3,D	5,D
3	ADCB				7,B	3,D	5,D
4	ADCBF				7,B		4,F
5	ADCBFG				5,G		
6	ADCBFGE						

Hence, the Shortest Path tree for node A is



Ans 9

(a) Slow Start is at intervals 1-6 and 23-26 sec.
Cwnd starts from 1 and increases exponentially.

(b) 24^{th} segment is sent in 5^{th} round

In 1^{st} round → packet 1 is sent

2^{nd}	→	2-3
3^{rd}	→	4-7
4^{th}	→	8-15
5^{th}	→	16-31
⋮		
n^{th}	→	$2^{n-1} - (2^n - 1)$

(c) Triple duplicate happens in 26^{th} round

$$\left.\begin{array}{l}\text{Now, cwnd} = 8\,MMS \\ \text{ssthread} = 13\,MMS\end{array}\right\}\text{ initially}$$

$$\boxed{\text{new cwnd} = \text{new ssthread} = \frac{\text{cwnd}}{2} = 4\,MMS}$$

f Ans 10 ⓐ Statement $\boxed{\text{True}}$

TCP uses sliding window protocol. The sender sends data equivalent to size of window and the receiver stores that in buffer. Now, Buffer Size > Window Size

$$\text{Number of ACK}_{\text{received}} \leq \text{Window Size}$$

$$\text{No. of ACK} + \text{No. of unACK} = \text{Window Size}$$

$$\Rightarrow \boxed{\begin{array}{l}\text{No. of unacknowledged bytes} \\ \leq \text{window size} < \text{buffer size}\end{array}}$$

ⓑ $\boxed{\text{False}}$

If segment has n bytes, if current segment number is m, next segment number is m+n, and n need not be 1 always.

ⓒ $\boxed{\text{False}}$

$$RTT_{new} = \alpha^{*} \text{estimated}_{\cancel{prev}} RTT + (1-\alpha) RTT_{\cancel{prev}}$$

$$\Rightarrow \text{If } RTT = 1, \text{ then } RTT_{new} = \alpha^{*} \text{estimated} RTT + 1 - \alpha$$

Hence, ~~cost~~ RTT depends on α, estimated RTT

ⓓ $\boxed{\text{False}}$

Well, if one of the packets in 38-41 are lost. That could be sent again, so next need not be 42 packet

Ans 11. CIDR notation is used to represent an IP address and its associated ~~mask~~. network mask.

$\boxed{\text{IP address / suffix}}$

suffix → Represents the number of network bits in the subnet mask

Eg:
An IP address of 131.10.55.70 with a subnet mask of 255.0.0.0 (which has 8 network bits) is represented as 131.10.55.70/8

Given, IP address = 222.1.1.20
Mask = 255.255.255.192

Binary equivalent of Mask is

$\underbrace{11111111}_{8}\ \underbrace{11111111}_{8}\ \underbrace{11111111}_{8}\ \underbrace{11}_{2}000000$

The mask has 8 + 8 + 8 + 2 = 26 network bits

Hence, CIDR notation is 222.1.1.20/26 for the IP address mask

Ans 12 Given network address = 211.1.1.0
Subnet mask = 255.255.255.0

Binary notation of mask → $\underbrace{11111111}_{8}$ $\underbrace{11111111}_{8}$ $\underbrace{11111111}_{8}$ 00000000

⇒ Network Bits count = 24

Hence, 32-24=8 bits can be used to address the host

Hence, 2^8 = 256 address ~~are~~ can possibly be used
but 2 address are reserved as the
subnet ID (255.255.255.0) and ~~the last~~
broadcast address (255.255.255.255)

So, $\boxed{\text{total hosts possible is } 256-2 = 254}$

Ans 13 Given IP address is 195.1.1.0

Default subnet mask for class C IP is 255.255.255.0

One bit can be used to distinguish b/n both subnets.

For representing 48 hosts, we need 6 bits
[∵ $32 = 2^5 < 48 < 2^6 = 64$]

∴ 7 bits are need to represent the 2 subnets ~~and~~
[with each 48 hosts] ~~atleast~~

So, the masks could be 255. 255. 255. x

where x is 128 = 10000000
or
192 = 11000000

Hence possible masks are { 255 . 255.255.128
255 . 255. 255. 192

Ans14

IP address = 197.1.2.67
Mask = 255. 255.255.192

Binary forms of

IP → 11000101 00000001 00000010 01 000011
Mask → 11111111 11111111 11111111 11 000000

So, these bits represent subnet

⇒ subnet = 01 = 1

So, these bits represent host

⇒ host = 000011
= 3

Ans15 (a) Destination address = 00100000

Now, the prefixes matching are 00 and 001

But the longest-prefix that is matching is 001

Hence, this datagram ~~address~~ is forwarded to interface 4

(b) Destination address = 11001100

Now, prefix that matches is 110 only.

Hence, datagram is forwarded to interface 5

(c) Destination address is 01101110

Now, prefix that matches is 011 only.

Hence, datagram is forwarded to interface 3

Ans 16.

172.23.88.0/23 is a private address space, with 23 network bits and 9 bits for host addresses.

~~To~~ Conditions specified to accomodate A,B,C subnets

i) Amount of address space is minimized
[No wastage]

ii) Largest possible contiguous space.

We can see subnet C requires 159 host addresses

∴ It requires $\lceil \log_2 159 \rceil = 8$ bits

Similary Subnet B needs $\lceil \log_2 20 \rceil = 5$ bits

Subnet A needs $\lceil \log_2 18 \rceil = 5$ bits

Hence, I have allocated the subnets as,

A → 172.23.89.32 to 172.23.89.63

B → 172.23.89.0 to 172.23.89.31

C → 172.23.88.0 to 172.23.88.255

So, answers in next page →

a) Private address space

b) No. of bits for host address = 9

No. of hosts = $2^9 - 2$ = 510

[2 are reserved addresses]

c) CIDR for Subnet A is → 172.23.89.32/27

d) Broadcast address for Subnet A is 172.23.89.63

e) 172.23.89.33 is the starting address for Subnet A

f) 172.23.89.63 is the ending address for Subnet A

g) Subnet B's CIDR is 172.23.89.0/27