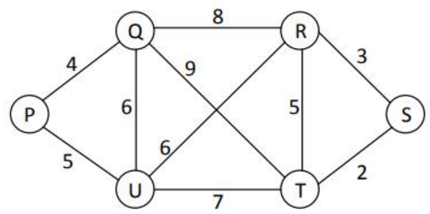
1. **Analyze** what problem arises if 'age' is set too low and what issue may arise if 'age' is set too high in the link state packets at the time of sequence number error. Draw a link state packet for router 'T' for the following subnet.



Ans- **If 'age' is set too low**

If the age of a link state packet (LSP) is set too low, the packet may expire and be removed from the routing database prematurely. This can cause routers to incorrectly assume that certain routes or network links are no longer available, leading to routing instability and potentially suboptimal routing paths. Additionally, frequent expiration and regeneration of LSPs can result in excessive flooding of these packets, consuming network bandwidth and processing resources on routers, which degrades overall network performance.

If 'age' is set too high

If the age of an LSP is set too high, the packet may remain in the routing database for too long. This can result in routers using outdated topology information for routing decisions, leading to inefficient or incorrect routing. Stale information in the routing database can cause delays in network convergence and potentially lead to routing loops or black holes, where packets are forwarded in circles or dropped because there is no valid path to the destination.

Link state packet for T subnet-

T	
Sequence#	
Age	
Q	9
U	7
R	5
S	2

2. Analyze how leaky bucket algorithm can help to control the flow which in turn helps network to remain congestion free. If the bursty input of leaky bucket algorithm is 496MB/sec for 950msec; what would be the duration of output for a leak considering network data rate is 24MB/sec?

Ans-

The Leaky-Bucket algorithm: It is like a bucket filled with water with a leak leaking water in small amounts. Pc devices create bursting nature data. Meaning it creates a huge amount of data at one time and sends it, then the other times creates small amounts of data. The network can not always handle this brusty data and will create congestion. So it needs to be shaped. So since a network interface card is the last point between a PC and network, here we apply The Leaky-Bucket algorithm.

So, in the algorithm there the bursty data will pool in a buffer and there will be a leak which has the size the network can handle the data. And from that leak in a steady fashion the packets will go to the network at a lower data rate. So the bursting nature data is regularized. The buffer has to have sufficient size to hold the huge pool of data.

Given that,

data rate = 496 MB/sec

time = 950 msec

duration = ?

network data rate = 24MB/sec

Total data generated.

$$\text{Data} = \frac{496}{1000} \times 950 \text{ MB}$$

$$= 471.2 \text{ MB}$$

$$\text{duration} = \frac{\text{Data}}{\text{Rate}} = \frac{471.2 \text{ MB}}{24 \text{MB/sec}}$$

$$= 19.635$$

$$= 19633.33 \text{ msec}$$

3. Following are the two scenarios where packet delays of 6 packets are shown in between source and destination. Find which of the following scenarios has low jitter and why. Briefly explain why low jitter is a necessity for multimedia communication and draw a graph to show a communication that has no jitter at all.

Scenario 1		
Source	Destination	Delay (msec)
P_1	P_1	89.5
P_2	P_2	91.6
P_3	P_3	89.4
P_4	P_4	90.7
P_5	P_5	90.8
P_6	P_6	89.9

Scenario 2		
Source	Destination	Delay (msec)
P_1	P_1	6
P_2	P_2	23
P_3	P_3	12
P_4	P_4	59
P_5	P_5	22
P_6	P_6	90

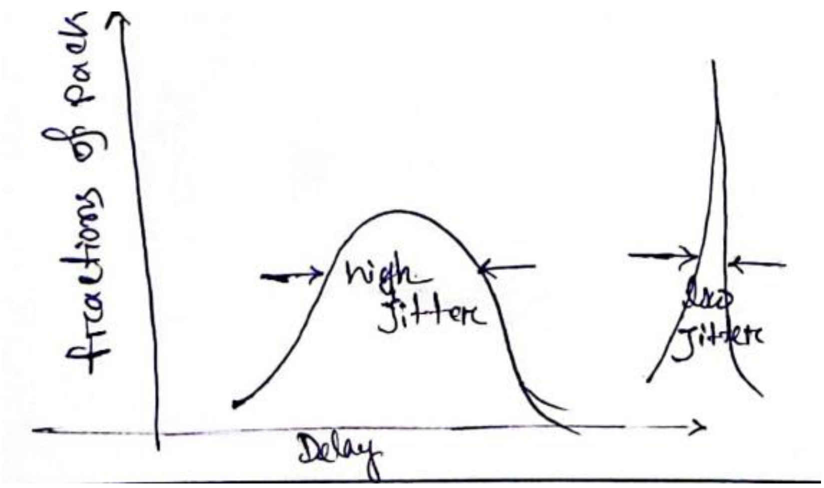
Ans-Here, in scenario 1 the jitter is low because the variation is low but delay is high . Besides, in scenario 2 the jitter is high because the variation is high and delay is low.

Audio and Video Synchronization: In multimedia communications like video calls, streaming, and online gaming, low jitter ensures that audio and video streams remain synchronized. High jitter can cause delays and desynchronization, leading to poor user experience.

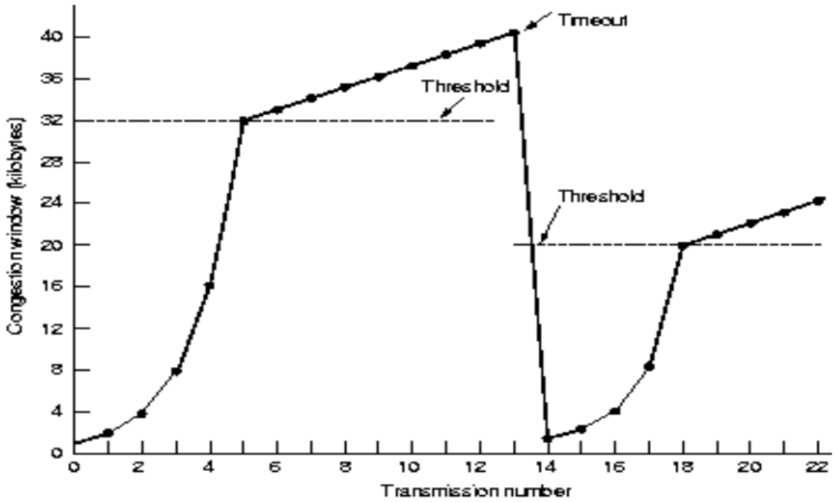
Continuous Streaming: Low jitter helps in maintaining a steady flow of data packets. This results in smooth and uninterrupted playback of multimedia content, reducing the chances of buffering and stuttering.

Consistent Quality: Low jitter contributes to a more consistent quality of service. In environments with high jitter, the quality can fluctuate, causing noticeable degradation in audio and video quality.

Reduced Latency: Jitter can introduce delays as systems need to buffer data to handle variability. Low jitter minimizes the need for large buffers, reducing overall latency and making interactions feel more immediate and responsive.



4. An example graph of congestion control in transport layer is given below. If the present threshold is set to be 612 KB and the present congestion window (i.e. transmission size) is 256 KB, **find** the sizes of the next 8 consecutive congestion windows considering 614KB is going to be the timeout point. (you don't need to draw the graph, only values will suffice)



Present threshold = 612KB
" congestion window = 256KB
timeout = 614KB

1st round
256×2 = 512KB

2nd round 512×2 = 1024KB

3rd round (1024 > 612)

3rd = 1024+1 = 1025KB

4th = 1025+1 = 1026KB

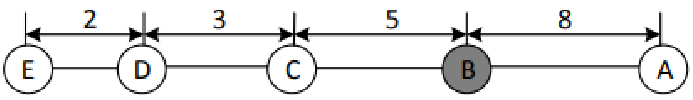
5th = 1027 KB

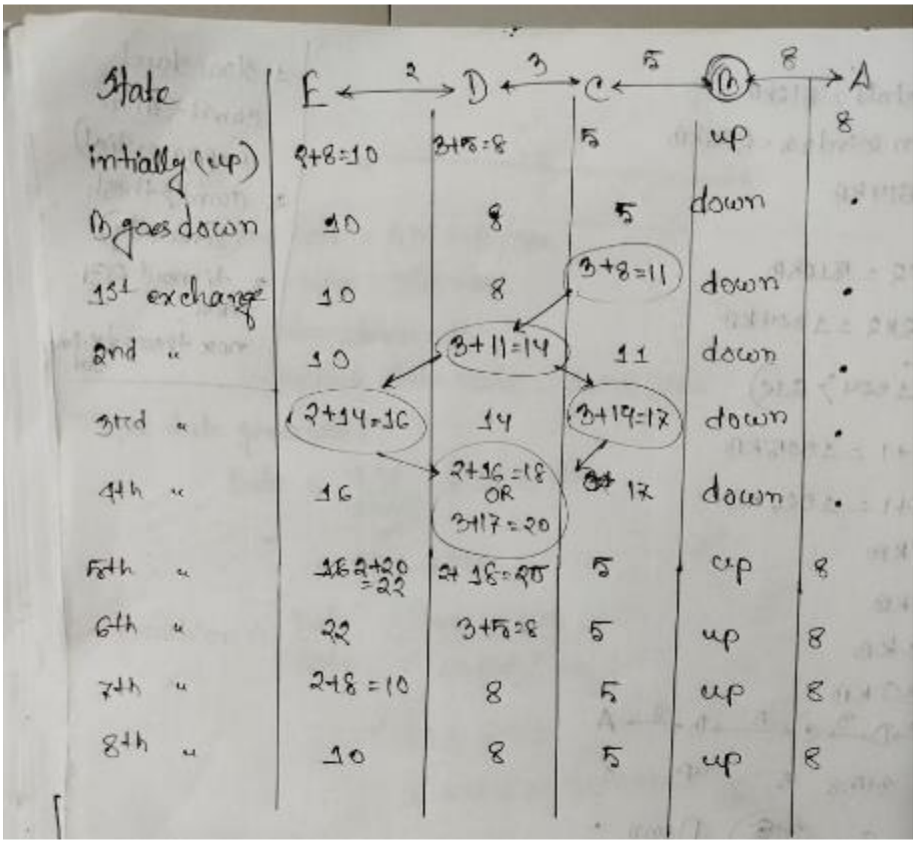
6th = 1028KB

7th = 1029 KB

8th = 1030 KB

1. Following is a linear subnet comprises of routers A, B, C, D and E; the internal distances between routers are shown in msec. **Show** the initial state considering router **B** is up. **Calculate** 4 more exchanges after router B gone down and additional 4 more exchanges after B gone up following the previous 4 exchanges for the following linear subnet.

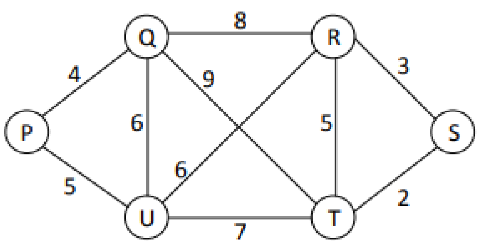




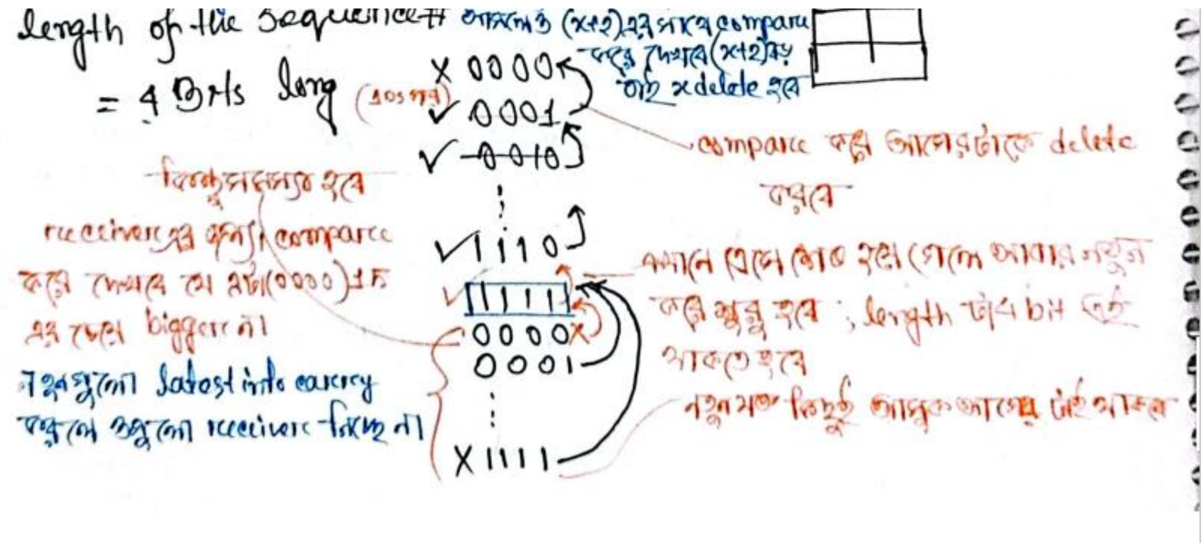
State	E ←2→ D	←3→ C	←5→ B	←8→ A	
initially (up)	2+8=10	3+5=8	5	up	8
B goes down	10	8	5	down	·
1st exchange	10	8	3+8=11	down	·
2nd "	10	3+11=14	11	down	·
3rd "	2+14=16	14	3+14=17	down	·
4th "	16	2+16=18 OR 3+17=20	~~3+~~ 17	down	·
5th "	~~16~~ 2+20 =22	~~2+18=20~~	5	up	8
6th "	22	3+5=8	5	up	8
7th "	2+8 =10	8	5	up	8
8th "	10	8	5	up	8

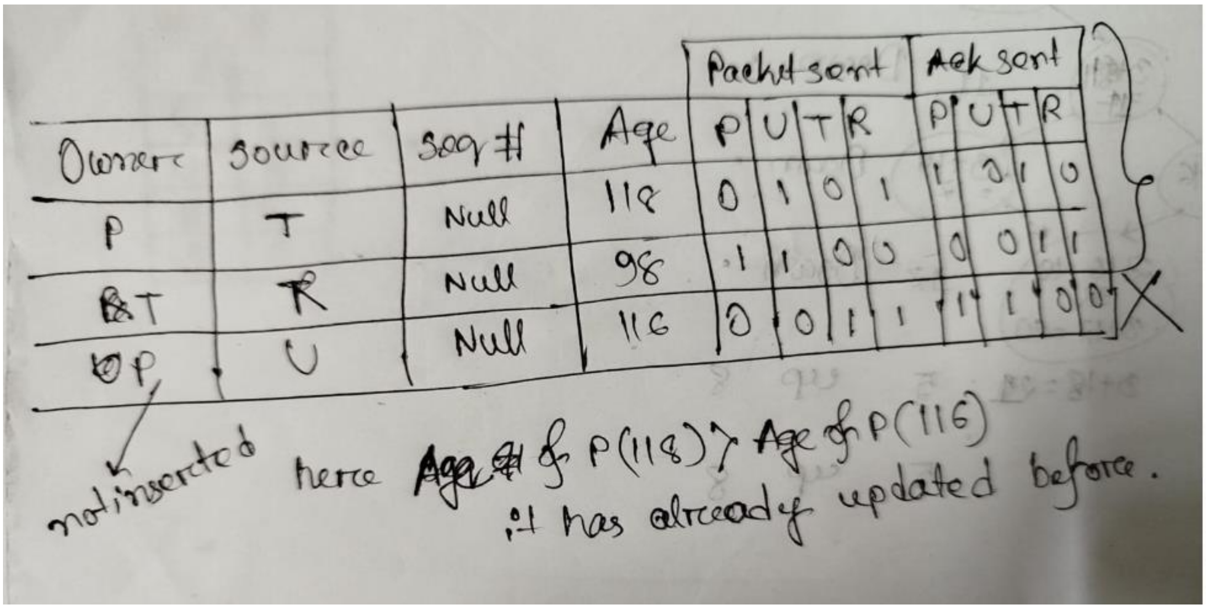
2. Briefly state how short length of a sequence number could be a trouble for link state packets. **Create** a distribution table for the router Q of following subnet for the following conditions:

   a. When Q receives P's link-state packet from T with seq. no 118
   b. When Q receives T's link-state packet from R with seq. no 98
   c. When Q receives P's link-state packet from U with seq. no 116



Ans-If the length of the sequence is short then, if a source generates a packet very quickly then the sequence number will quickly reach the maximum number. After that the source router will restart the sequence and send the updated packet will have a sequence number from the start. But since that new sequence number is lower than the max sequence number the receiver has, the receiver will not accept it even though the packet is new and updated.





Owner	Source	Seq #	Age	Packet sent P	U	T	R	Ack sent P	U	T	R
P	T	Null	118	0	1	0	1	1	0	1	0
~~B~~T	R	Null	98	1	1	0	0	0	0	1	1
P	U	Null	116	0	0	1	1	1	1	0	0

not inserted here Age of P(118) > Age of P(116)

it has already updated before.

3. **Define** the function of leaky bucket algorithm. Calculate the leak of the bucket if the system needs to be designed to transmit the received data from the PC with a rate of 512MB/sec for 950μsec in 25msec in a network whose data transmission capacity is 64MB/sec.

Ans-Function of Leaky Bucket Algorithm

The leaky bucket algorithm can be visualized as a bucket with a small hole at the bottom:

- **Bucket capacity**: The maximum amount of data the bucket can hold.

- **Incoming data**: Data packets arrive and are added to the bucket.
- **Outgoing data (leak rate)**: Data is sent out of the bucket at a constant rate, corresponding to the network's transmission capacity.
- **Overflow**: If the incoming data rate exceeds the leak rate and the bucket fills up, any additional incoming data is discarded

incoming data rate - 512MB/s

duration of incoming data burst = 950μs

total duration of transmission = 25ms

Network transmission capacity = 64MB/s

1st part / Total incoming data = incoming data rate × burst duration

$= 512 \text{ MB/s} \times 950 \mu s$

$= 512 \text{ MB/s} \times 950 \times 10^{-6} s$

$= 0.4864 \text{ MB}$

2nd part / Total transmission capacity = Network transmission rate × total transmission time

$= 64 \text{ MB/s} \times 25 ms$

$= 64 \times 25 \times 10^{-3} \text{ MB} = 1.6 \text{ MB}$

<u>Calculating overflow</u>

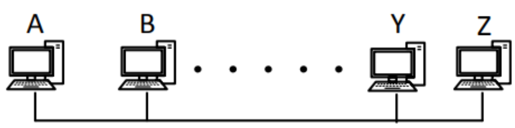
- The system can transmit 1.6MB in 25ms
- the burst data = 0. total incoming data = 0.4864MB

Here, total transmission capacity > burst size (total incoming data)

$\Rightarrow 1.6\text{MB} > 0.4864\text{MB}$

∴ the bucket won't overflow

3. A bit map protocol is followed by the following nodes. The propagation delay between the farthest two nodes is 9μsec, calculate the minimum duration of the polling phase if there are 26 nodes present in the following LAN segment. If the part of the bit map is 10011…..101011, how would node **E** determine its turn to transmit.



propagation delay between two farthest nodes = $9\mu s$

no of nodes = 26 = (প্রতিটি [illegible] $1\mu s$ হলে $26 \times 1\mu s = 26\mu s$)

total propagation delay = $2 \times$ propagation delay

$= 2 \times 9\mu s = 18\mu s$

$\therefore$ total polling phase duration = 2 propagation delay + processing time

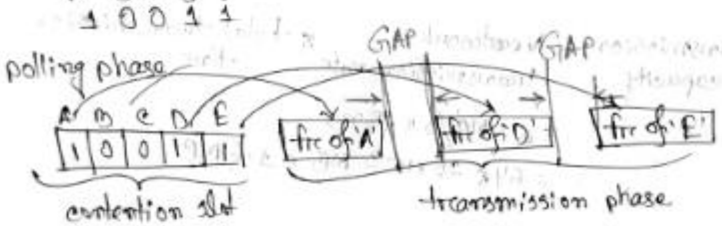
$= 18\mu s + 26\mu s$

$= 44\mu s$

Given bitmap, 10011 . . . 101011

A B C D E

1 0 0 1 1

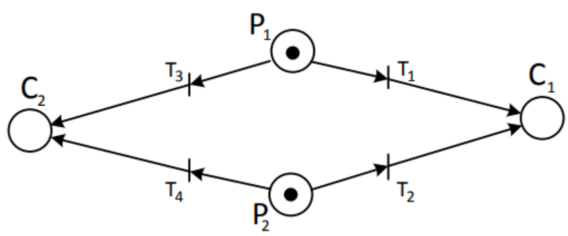


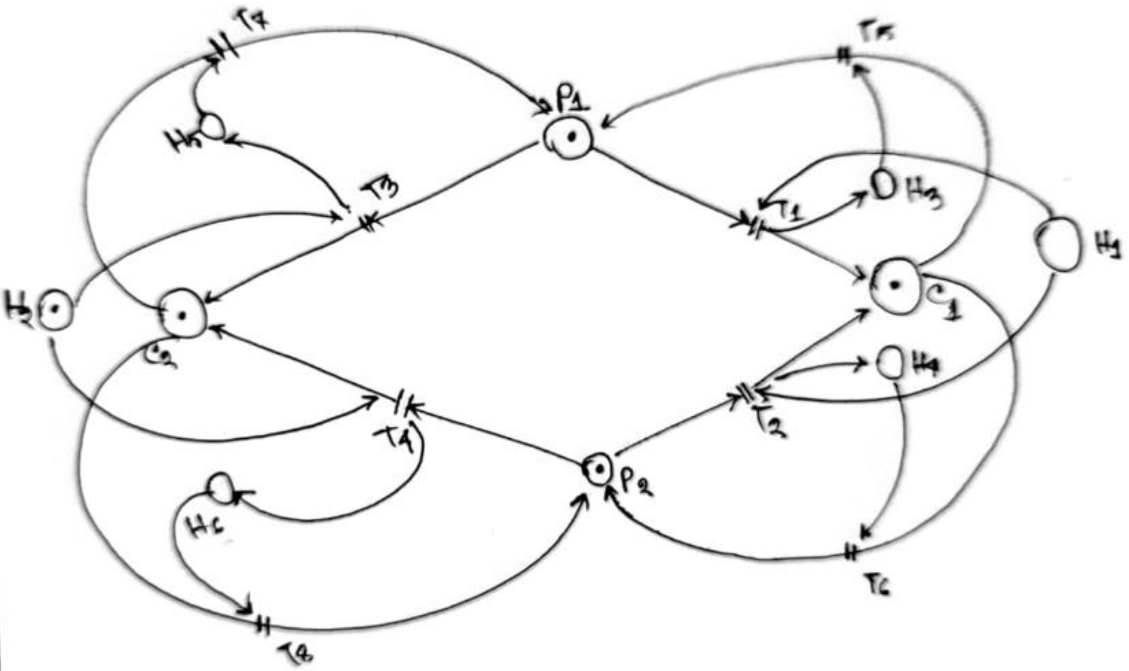
Here, 1 = if interested to transmit

0 = if not interested to transmit.

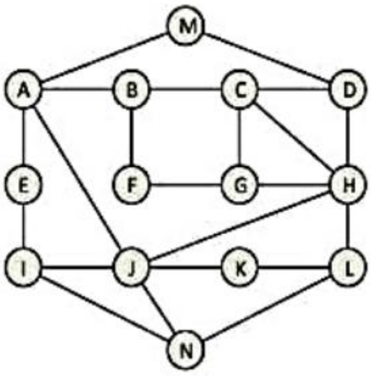
A will transmit first then D will transmit. Lastly E will transmit. Here E knows there are 2 device will transmit (A, D). So, D has to count 2 gaps then it will know that it's now 'D''s turn to transmit.

5. Determine the complete Petri net model for the following situation (mutual exclusion) where two processes P_1 and P_2 are going to use critical resources C_1 and C_2. Please note that critical resources can be used by one process at a time and having multiple tokens in any state is a violation.





4. Following is a subnet and the routing tables that router 'C' has from its neighbors B, D, G and H. The routers in the subnet follows distance vector routing algorithm. **Find** which path 'C' is going to take to reach 'J' if it computes the values to reach its neighbors (B, D, G and H) 10, 19, 16 and 18 msec respectively in that moment.



To	B	D	G	H
A	0	23	14	4
B	26	17	12	15
C	24	9	18	16
D	1	5	5	17
E	2	15	4	29
F	8	17	6	32
G	15	36	6	35
H	9	12	0	7
I	8	0	17	12
J	25	9	31	13
K	27	14	11	0
L	12	5	8	9
M	14	6	21	6
N	8	9	15	11

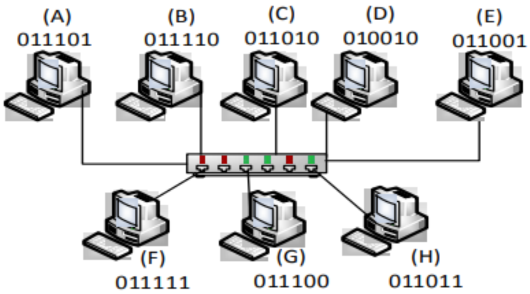
Source : C
Destination : J

C to J

CB + BJ = 10 + 25 = 35 ms
CD + DJ = 19 + 9 = 28 ms
CG + GJ = 16 + 31 = 47 ms
CH + HJ = 18 + 13 = 31 ms

from above calculation, I will choose to send the packet to D , since D will be the shortest time.

4. Show and **Solve** which of the following numbered stations will acquire the channel at first if B, C, D, G and H are interested to acquire channel by following "Binary countdown" protocol? In general, which stations has the possibility of getting into starvation in binary count down method.



Given that,
B = 011110
C = 011010
D = 010010
G = 011100
H = 011011

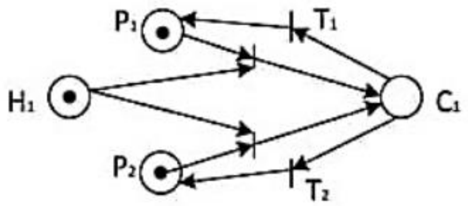
B (011110)	0	1	1	1	1	0
C (011010)	0	1	1	X	X	X
D (010010)	0	1	X	X	X	X
G (011100)	0	1	1	1	X	X
H (011011)	0	1	1	X	X	X
OR	0	1	1	1	1	0

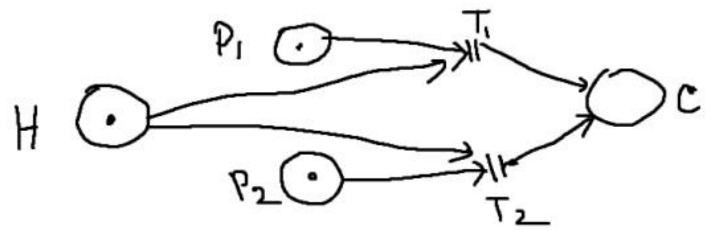
→ B winner

Here, B will acquire the channel at first.

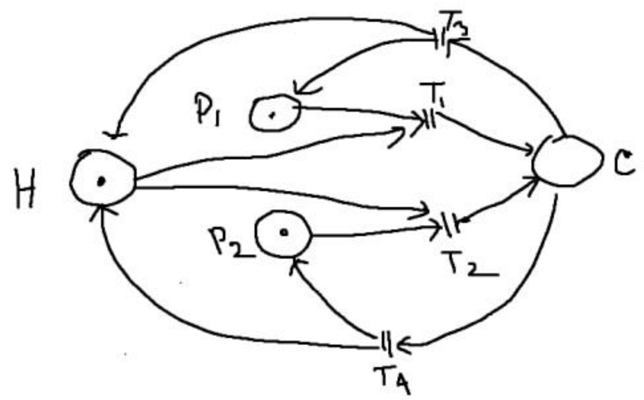
In general, when any device gets 000000 which is the lowest number and whose has the less chance of winning and only when it transmits data there is no other device all together. this situation is called starvation as like when CPU doesn't get the process time.

5. **Find and analyze** the problems exists in the following Petri net model for mutual exclusion between two processes $\mathbf{P_1}$ and $\mathbf{P_2}$, on the critical resource $\mathbf{C_1}$.

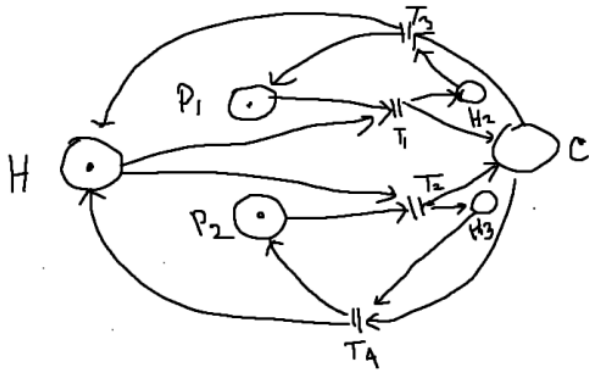




Here, to stop the malfunction we can use a helper state. Now we can see that to transit T1 and T2 needs 2 Tokens. When P1 need to use C it will get a helper Token from H to Transit T1 and go to C but at the same time if P2 wants to transit, It won't be able to transit T2 because T2 also need 2 Tokens and H already gave its token for T1s transition. So there the mutual Exclusion is intact.



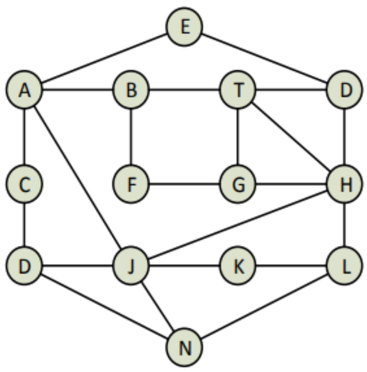
Now C can also return the tokens it received for a process to start from the said process and H through T3 and T4. But it still has a problem when returning the token.
When returning a Token to P1(suppose) C will send Tokens to both P1 & H through T3 and also to P2 and H through T4 since 2 arcs are going out of C to the T3,T4 transitions. At that time P1 will receive its 1 token but H will receive 2 Tokens and P2 will also have 2 Tokens because it will receive another Token including its original Token which was already in P2. This will cause another malfunction.



Here, To combat the malfunction we will use another H2, H3 helper states. H2, H3 will receive a Token from T1 and T2 when whichever transition is being used. Then when C returns the Token the Transition T3 and T4 will need 2 Tokens to transit. So suppose when P1 wants to use C it will use the T1 transit and use the help of H and transit from T1 to C. When transiting from T1, 1 Token will go to H2 and one will go to C. Now when C returns the Token back to P1 it will return 1 Token to both T3 and T4 but now these Transitions require 2 Tokens. So for the case of T3 it will receive 1 Token from C and 1 from H2 and then from T3 1 Token will go to P1 and H. But T4 will only receive 1 Token from C but no Token from H3 because H3 did not receive any Tokens. So T4 transition will not occur.

Now we have the perfect and final model of **Mutual Exclusion** for the **Petri Net Model**.

4. Following is a subnet and the routing tables that router '**T**' has from its neighbors B, D, G and H. The routers in the subnet follow distance vector routing algorithm. **Find** which path '**T**' is going to take to reach '**N**' if it computes the values to reach its neighbors (B, D, G and H) 10, 19, 16 and 18 msec respectively in that moment.



To	B	D	G	H
A	0	23	14	4
B	26	17	12	15
C	24	9	18	16
D	1	5	5	17
E	2	15	4	29
F	8	17	6	32
G	15	36	6	35
H	9	12	0	7
I	8	0	17	12
J	25	9	31	13
K	27	14	11	0
L	12	5	8	9
M	14	6	21	6
N	8	9	15	11

T to N; $TB + BN = 10 + 8 = 18$ ms

$TD + DN = 19 + 9 = 28$ ms

$TG + GN = 16 + 15 = 31$ ms

$TH + HN = 18 + 11 = 29$ ms

So, From the above calculation T will choose to send the packet to B, since B it will the shortest time.

3. What would be the duration of contention phase if contention slot for each host is τ for a 10 hosts LAN where a bit map protocol is followed for channel allocation. If the bit map is "1100101011" for the hosts A, B, ……,I,J respectively, how host G is going keep track of its turn.

- The LAN has 10 hosts (A, B, C, D, E, F, G, H, I, J).
- The bit map representing the hosts contending for the channel is "1100101011".
- The contention slot duration for each host is τ.

The duration of the contention phase is calculated as follows:

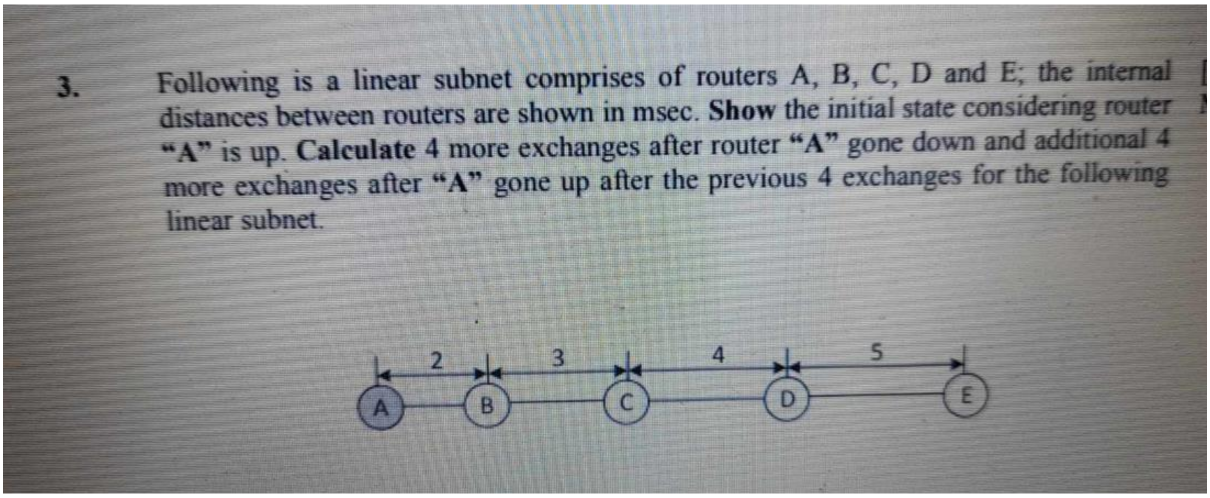
Number of contending hosts = Number of "1" bits in the bit map Number of contending hosts = 6 (hosts A, B, C, F, G, J)

Duration of the contention phase = Number of contending hosts × τ Duration of the contention phase = 6 × τ

A	B	C	D	E	F	G	H	I	J
1	1	0	0	1	0	1	0	1	1

Frame of 'A'	Frame of 'B'	____GAP________	Frame of 'E'	_GAP_	Frame of 'G'

G detects 3 gaps before it's turn because there 3 ones are there which are interested to transmit. After their transmission G will transmit.



3. Following is a linear subnet comprises of routers A, B, C, D and E; the internal distances between routers are shown in msec. **Show** the initial state considering router **"A"** is up. **Calculate** 4 more exchanges after router **"A"** gone down and additional 4 more exchanges after **"A"** gone up after the previous 4 exchanges for the following linear subnet.