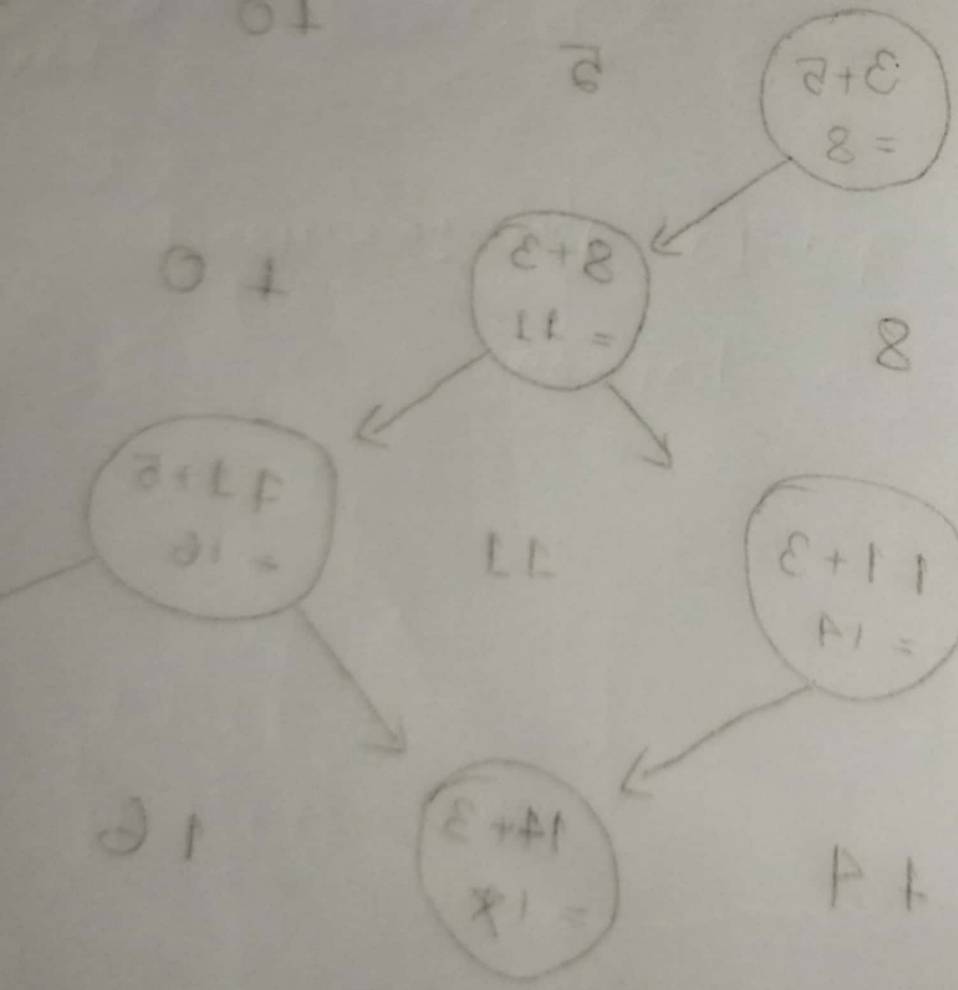
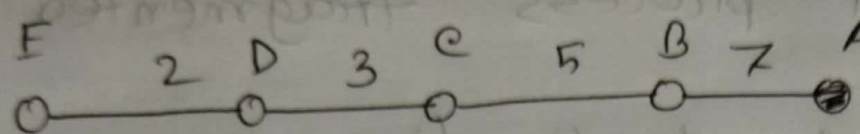
Name: Ryan Mohammad Bin Shahjahan

ID : 2019-1-68-031

Exam: Final



Ans: to the qu: NO: 1



UP

17 15 12 7 initially

down

17 15 12 5+12 = 17 ← 1st ex

17 15 17+5 = 22 17 ← 2nd ex

17 22+3 = 25 22 22+5 = 27 ← 3rd ex

25+2 = 27 25 27+5 = 32 27 ← 4th ex

UP

27 25 32 7 ← 1st ex

27 25 7+5 = 12 7 ← 2nd ex

27 12+3 = 15 12 7 ← 3rd ex

15+2 = 17 15 12 7 ← 4th ex

Ans: to the qu: NO: 02

In choke packet algorithm we can't get remedy. Packets are suffere to generate. for the same data rate, packets time consumption takes long. To solve this problem we use hop by hop choke packet algorithm. Because hop by hop algorithm reduce the data ret of the packet. so, every routers in their way reduce it. That is why controling packet can generate. But in the choke packet routers directly sends packet to the source.

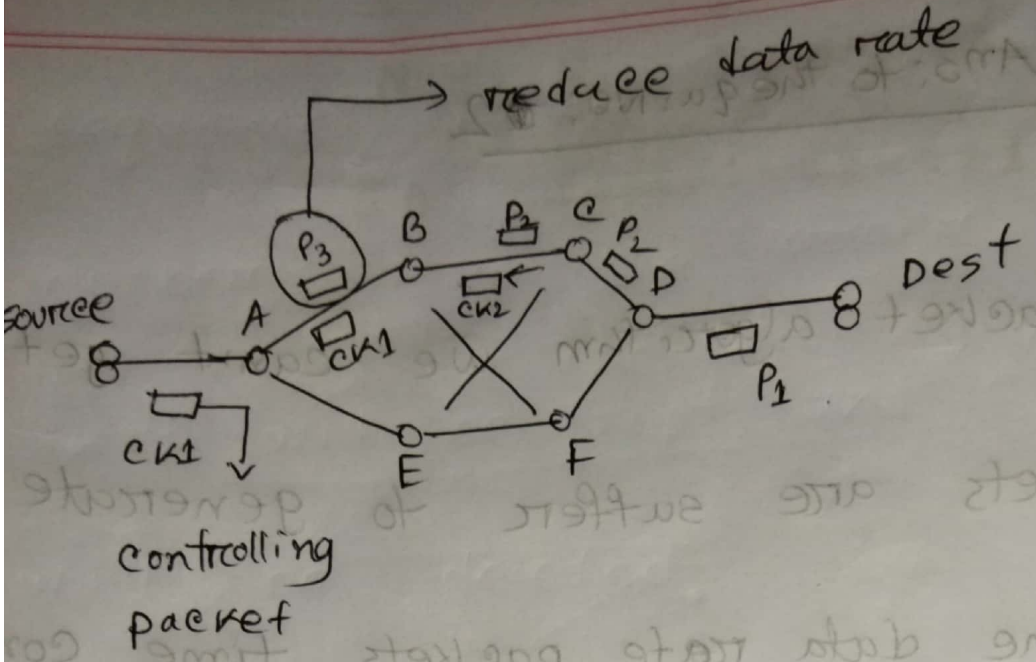


Diagram: hop-by-hop congestion control algorithm.

Ans: to the Qu: NO: 3

B	
sequence	
A	2
E	3
C	4

assuming, the data is sent from B.

If the neighbour router receive the packet, the sequence will be:

1

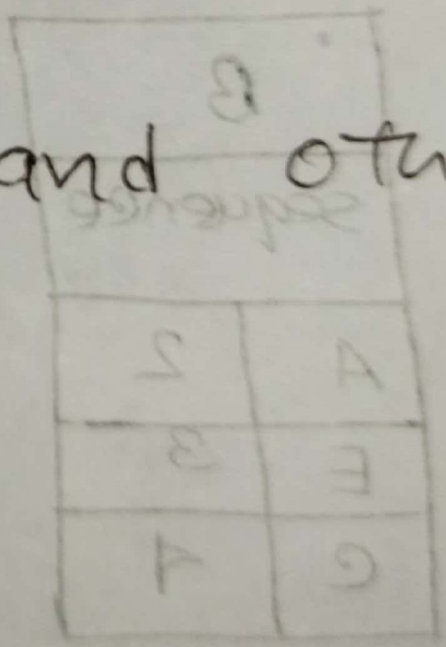
2

8 ← error

1
5
6
7
8 — not going to receive new information

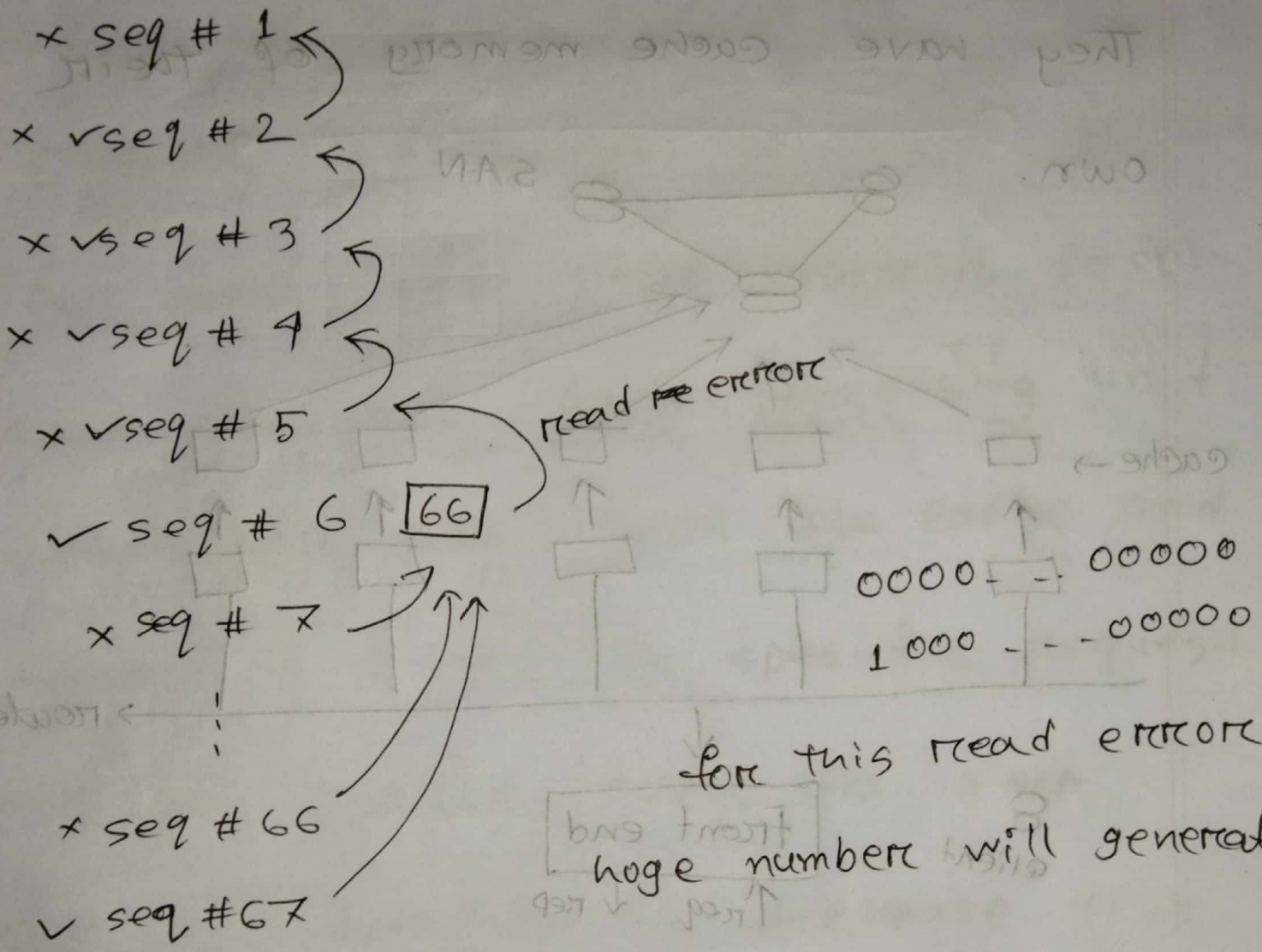
9 ← start taking new info

sequence number detect the latest number where the larger number that will be kept and other one will be deleted.



assuming,

assuming,

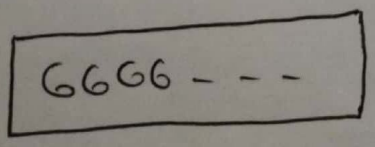


for this read error
huge number will generate

Example: 666666---

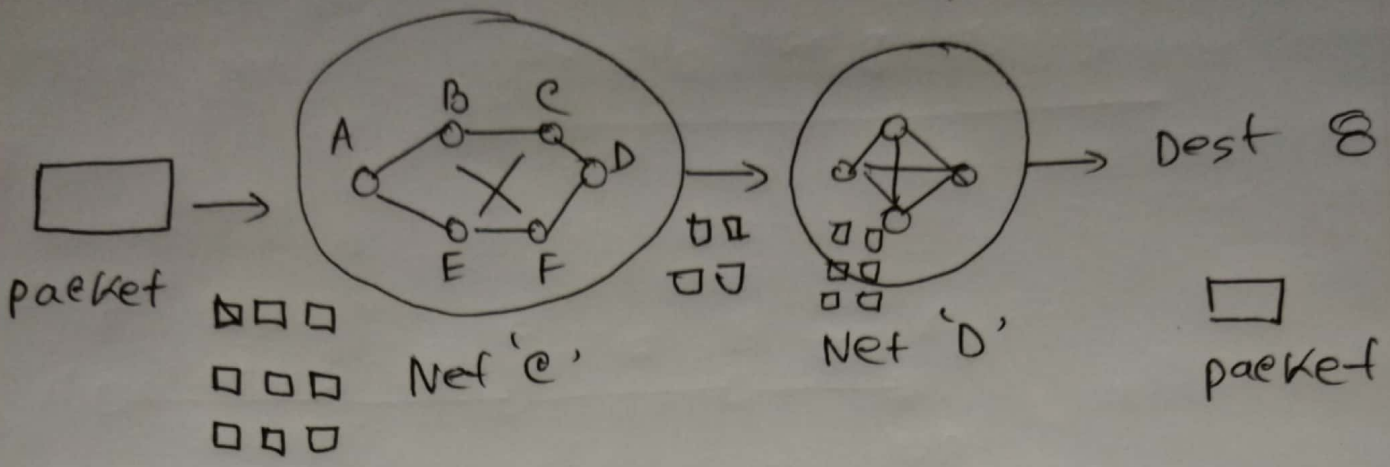
To solve this problem,
age has been added

Age
60sec



Ans: to the qu: NO: 4

Fragmentation process fragmented the data to send destination. Here packet's size greater than its transporting capacity, ~~packets~~ and the intermediate connection is oriented. so, we should use, non-~~trans~~ transparent fragmentation.



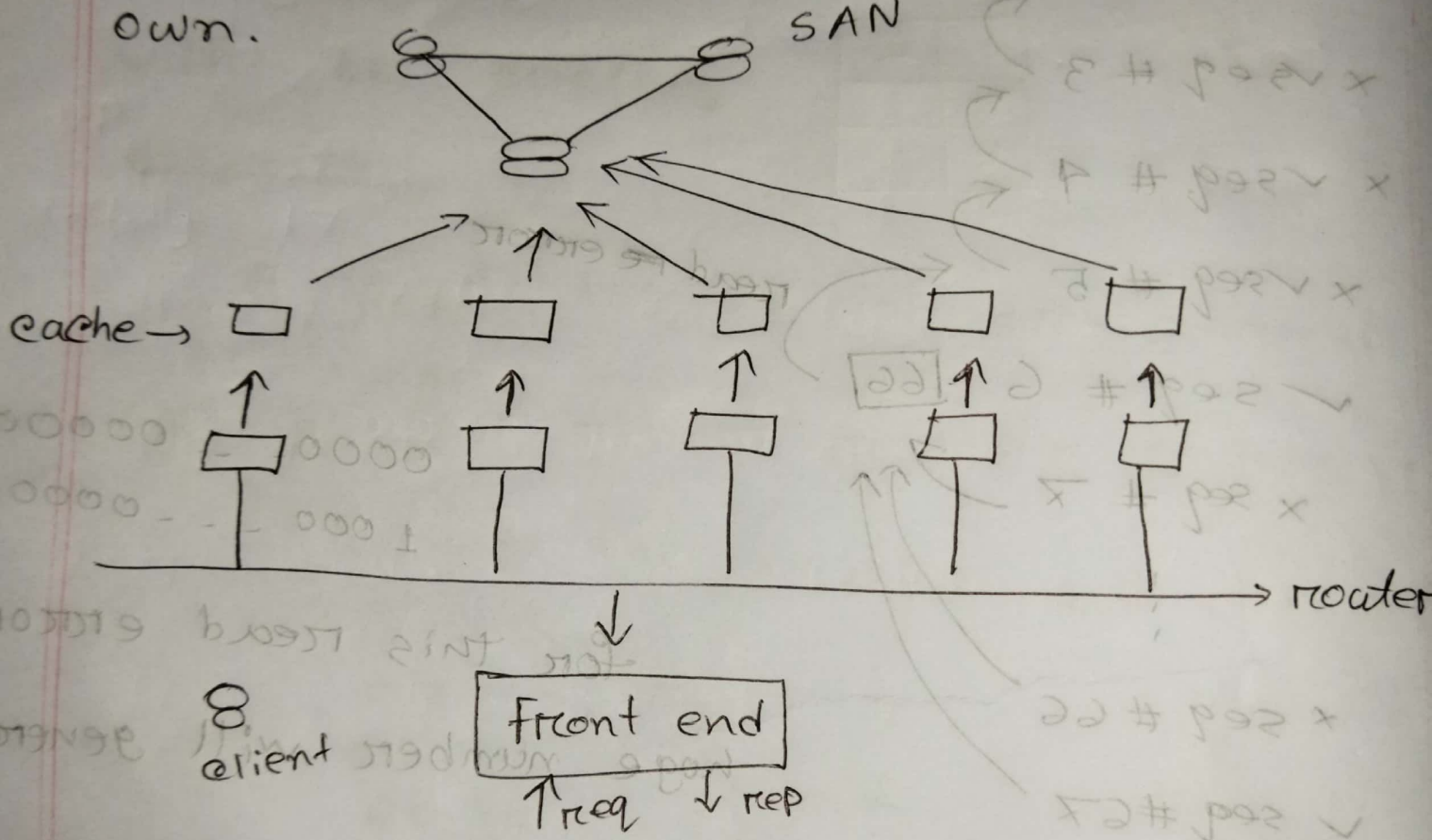
Non-transparent fragment

Its no need to reassemble, because it has flexibility. The largerr data can fragmented by its way and send data flexibly. No time goes for orientation and reassemble. If direct reassembled in destination.

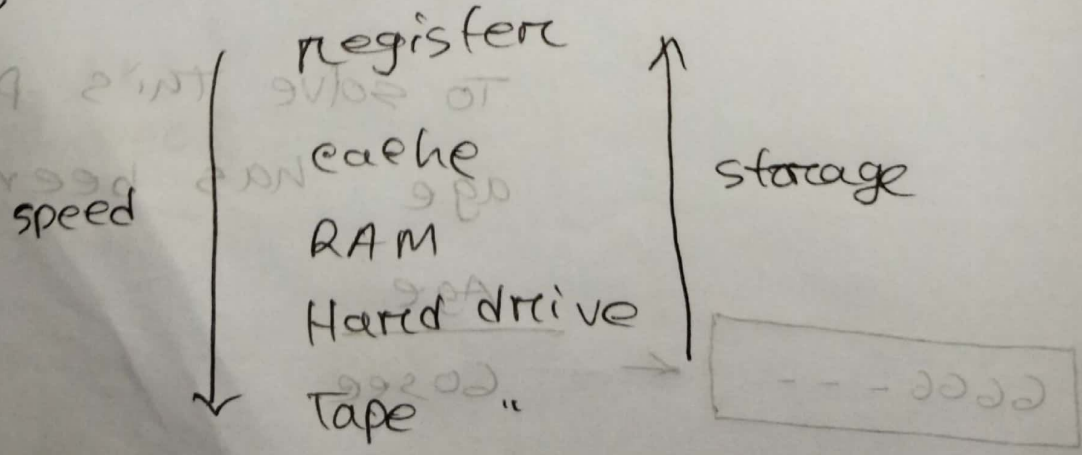
Thus, non-transparent fragmentation needed here.

Ans: to the qu: NO: 5

They have cache memory of their own.



Here,



The processing node can deliver request fastly and request will go to the client directly. If first unit ~~data~~ remain in cache x and client requests for that data the first processing unit will fetch data ~~from~~ ~~the~~ cache and serve to client. So, specific request will be handed over to a ~~sep~~ specific machine. To decrease time consumption. It's called TCP handoff