CONGESTION CONTROL TECHNIQUES

- Refers to the technique used to control or prevent congestion.
- Broadly classified into Two categories

(1) OPEN LOOP CONGESTION CONTROL

- Applied to prevent congestion before it happens.

Policies adapted by open loop congestion control

1. Retransmission Policy – If the sender feels that a sent packet is lost or corrupted, the Packet needs to be retransmitted.
   To prevent congestion, retransmission timers must be designed to prevent it.

2. Windows Policy – The type of window at the sender side may also effect the congestion.
   - We must use selective repeat window as it sends the specific Packet that may have been lost rather than Go-Back N.

3. Discarding Policy – must be adapted by the routers to prevent congestion by partially discarding the corrupted or less sensitive Package to maintain the Quality.
   Eg – In audio file transmission, routers can discard less sensitive Packets to prevent congestion & maintain quality of audio file

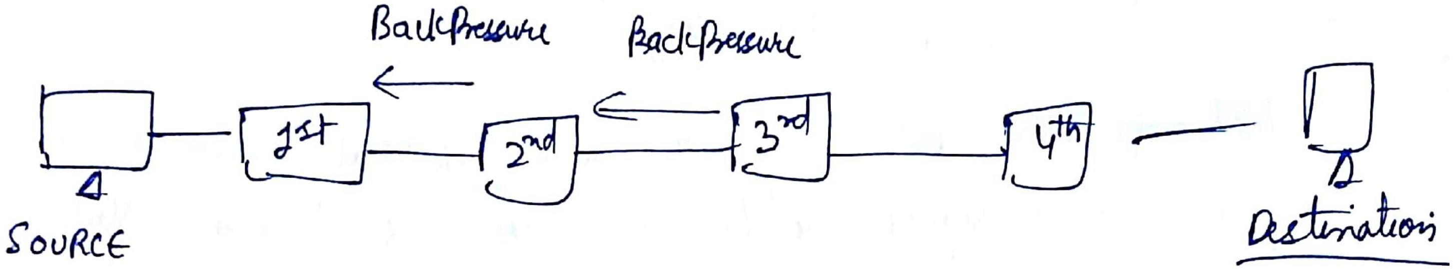
(4) Acknowledgement Policy – Since acknowledgement is also part of load in the n/w. The receiver should send acknowledge for N Packets rather than sending for a single Packet.

CLOSED LOOP CONGESTION CONTROL

used to treat or alleviate congestion after it happens

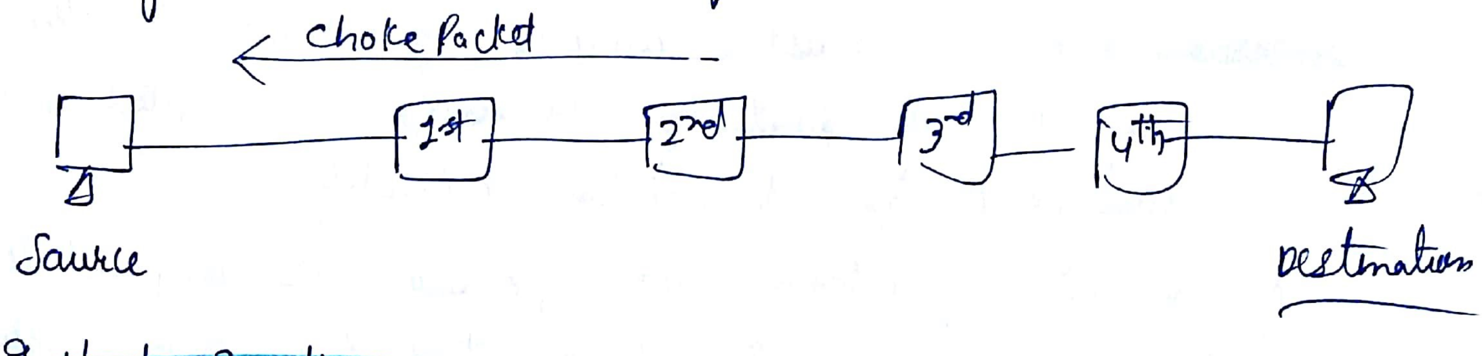
① **Back Pressure** – Technique in which congested node stops receiving Packet from upstream node.

- It is a node-to-node congestion control technique that Propagates in the opposite direction of data flow



2) **Choke Packet technique** – A choke Packet is a Packet send by a node to the Source to inform it of congestion :

- Each router monitors its resources & utilization, when resource utilization exceeds the threshhold value set by the administrator, the router directly sends a choke Packet to the Source giving it a feedback to reduce the traffic

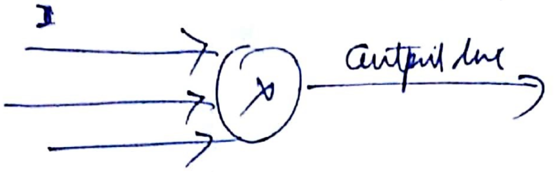


③ **Implicit Signalling**

The Source guesses That there is Congestion in a n/w because it does not receive any acknowledgement.

Causes of Congestion:

1. Input traffic rate exceeds the capacity of the output lines



2. Routers are too slow to perform bookkeeping tasks (Queuing buffers updating tables etc)

3. Router's buffer is too limited

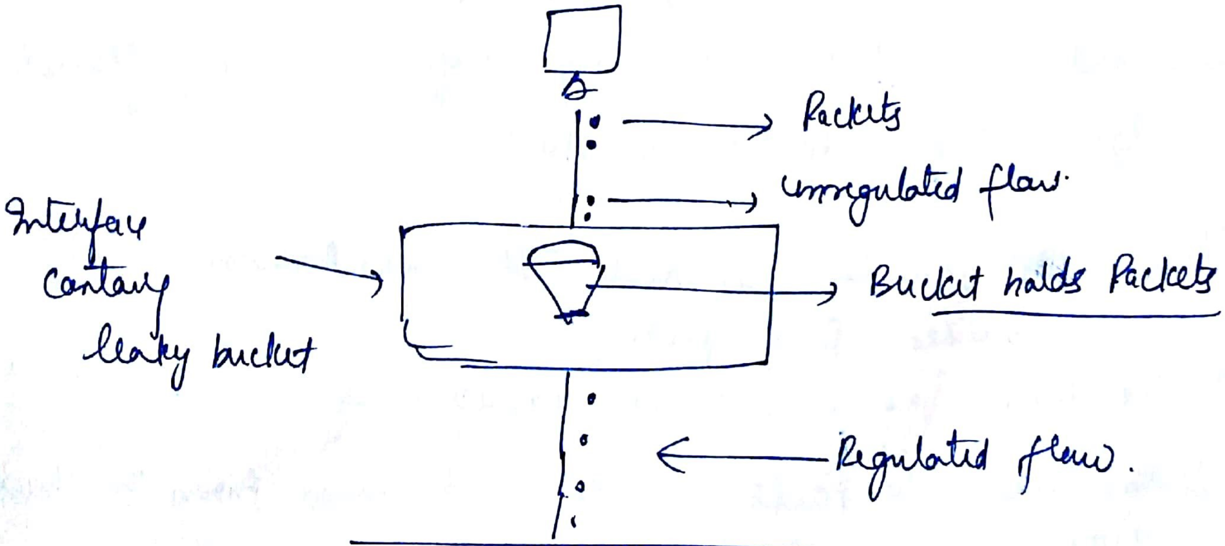
4. Congestion ~~a a~~ in a n/w can occur if the Processors are slow

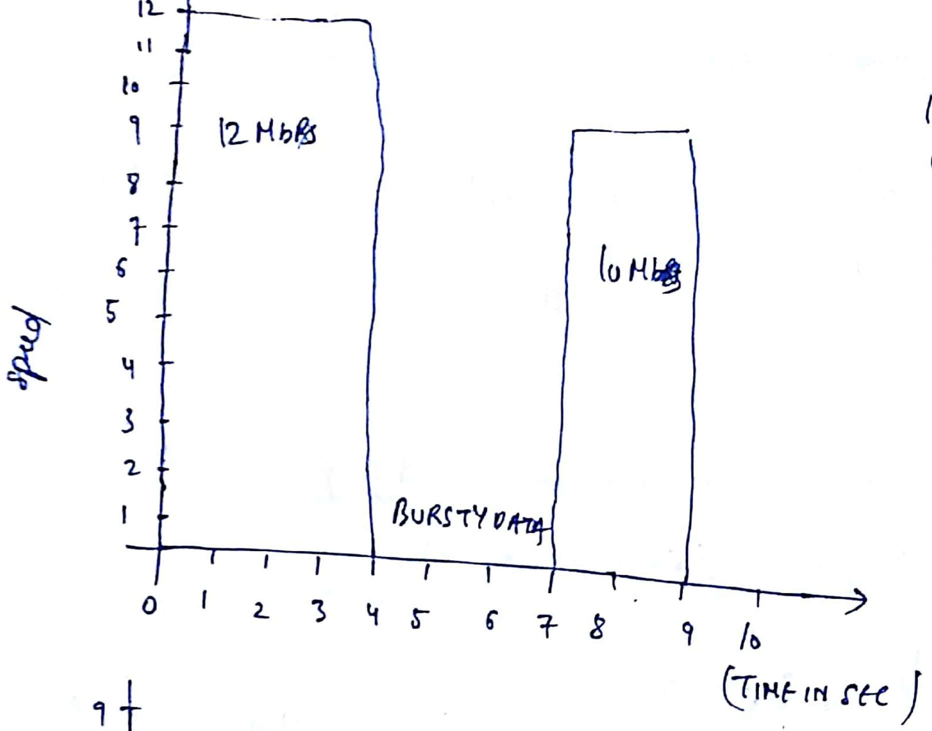
Congestion control algorithms

- Leaky Bucket algorithm
- Token Bucket algorithm

Leaky Bucket algorithm

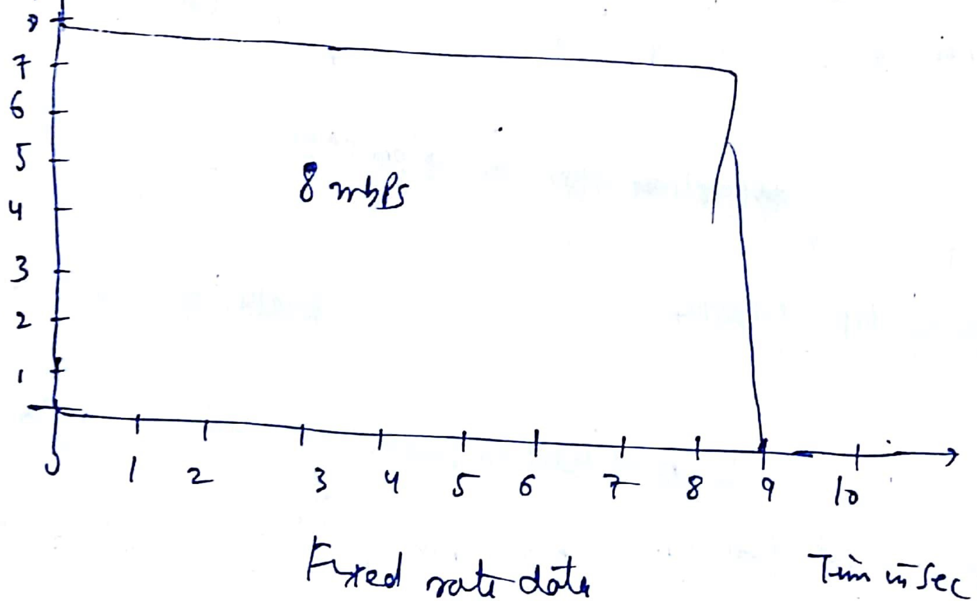
- A leaky bucket algorithm shapes bursty traffic into fixed rate traffic
- Rate at which the water is poured into the bucket is not fixed & can vary but it leaks from the bucket at a constant rate





1 sec - 12mbps
4 sec = 48

2sec = 10 ×2
= 20
= 68 mb.



② Token Bucket algorithm

In leaky bucket algorithm, it cannot deal with bursty data. So in order to deal with the bursty traffic, we need a flexible algo so that data is not lost

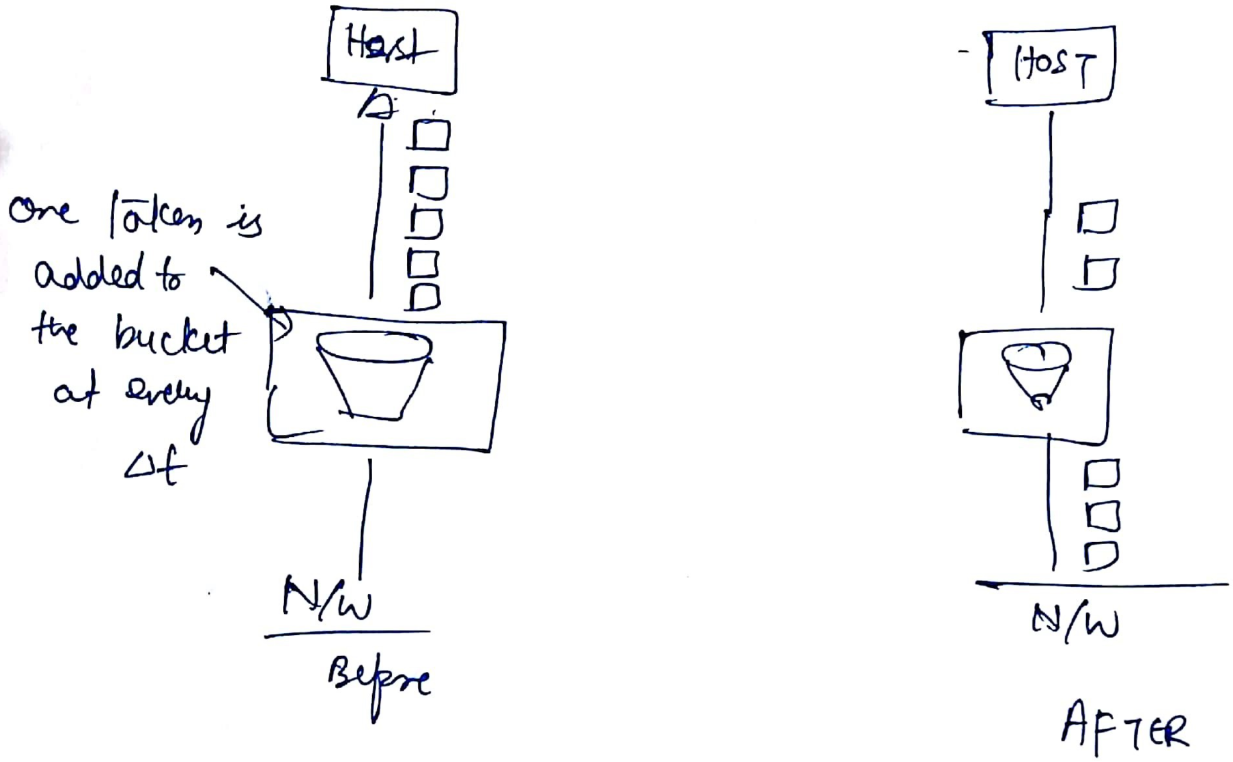
Steps - ① In regular intervals, tokens are thrown into the bucket ~~f~~ of capacity f

② The bucket has a maximum capacity f

③ If there is a ready packet, a token is removed from the bucket & the packet

(3)

(8) If there is no tokens into the bucket, the Packet cannot be send.



Host
One token is added to the bucket at every Δt
N/W
Before
HOST
N/W
AFTER