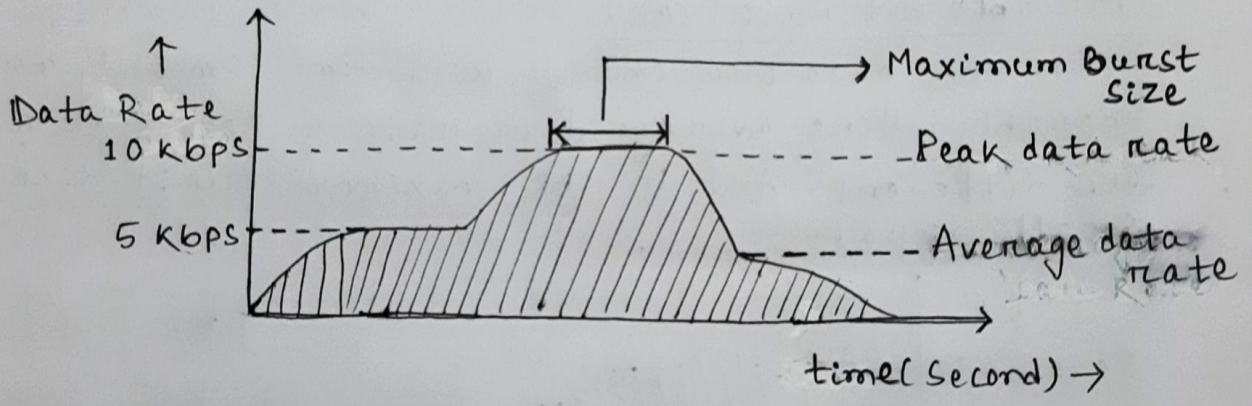
Congestion Control & Quality of Service

→ To maintain or balance data traffic so that congestion will not arise & at the same time we also preserve the quality of service.

→ **Traffic Descripton** –

Traffic descripton provide qualitative values that represent data flow. Different types of traffic descriptons are,

(a) Average data rate
(b) Peak data rate
(c) Maximum Burst size
(d) Effective Bandwidth



a) **Average Data Rate** – It is defined as no. of bits send during a period of time divided by no. of seconds in that period. i.e:

$$\text{Avg. Data Rate} = \frac{\text{Amount of Data}}{\text{time}}$$

b) **Peak Data Rate** – It defines maximum data rate of data traffic or peak bandwidth of network.

c) **Maximum Burst Size** – It refers to the maximum length of time the traffic is generated at peak rate.

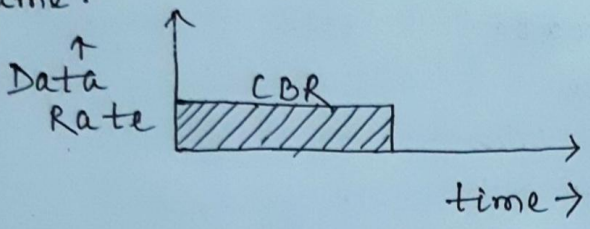
d) **Effective Bandwidth** – It is the bandwidth that the network need to allocate for the flow of traffic. Effective BW is a function of average data rate, peak data rate & maximum burst size.

Effective BW = f(avg. data rate, Peak data rate, Maximum Burst size)

→ Traffic Profile → It provides different types of data flow. They are
- Constant bit rate (CBR)
- Variable bit rate (VBR)
- Bursty Data (BD)

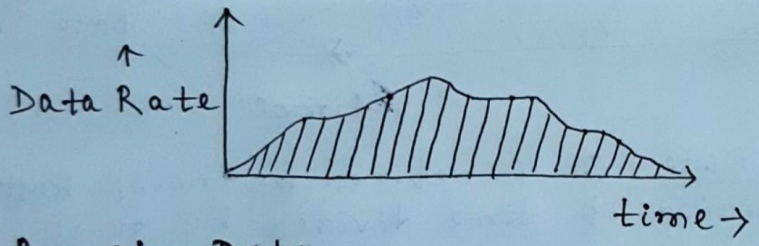
Constant Bit Rate (CBR)

It is fixed data rate that doesnot change. Here in CBR average data rate & peak data rate are same.



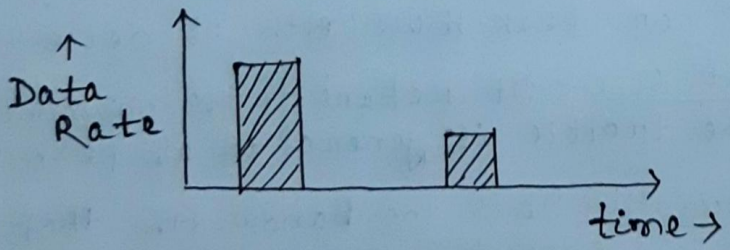
Variable Bit Rate (VBR)

In VBR data flow changes with time smoothly not sharphly. Here average data rate & peak data rate are different. And in CBR maximum burst size is small duration.



Bursty Data

In Bursty Data, data rate changes sharphly or suddenly in a very short time. Here average bit rate & peak bit rate are different.



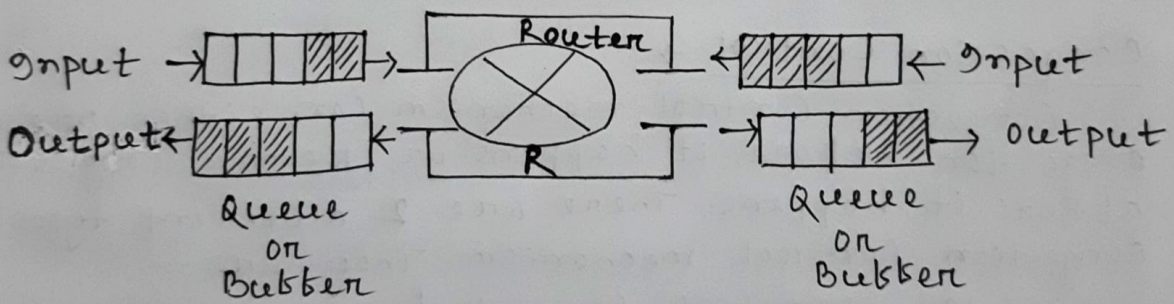
Congestion

→ Congestion in a network occur because load in a network is greater than capacity of the network. Load means no. of packets send to the network. Capacity means no. of packets a network can handle.

→ Congestion in a network occur because router or switches have queues or buffers that hold the packets before and after processing.

→ Different steps to process in a queue -

1. Packets is placed at the end of queue while waiting to be checked.
2. Processing module of router removes packets from input queue once it reaches the front of queue & uses its routing table & the destination address to find the route.
3. Packet is put in the appropriate queue & wait its turn to be send.

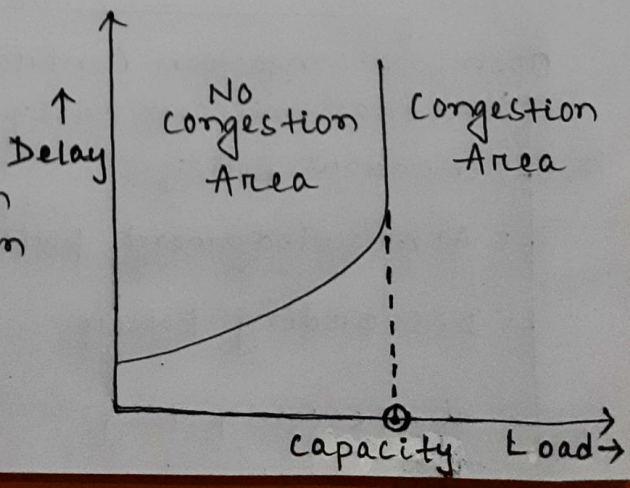


Network Performance →

Network performance is described by describing comparison between Delay, Load and throughput, Load.

a) Delay vs Load -

When Load is much less than capacity then delay is minimum. And the minimum delay is due to propagation delay & processing delay, which is negligible.

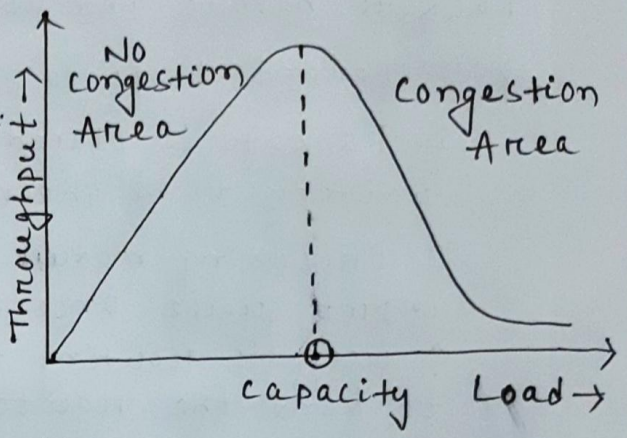


when Load reaches the network capacity, delay increases sharply because now delay is due to the propagation, processing & waiting time in queue.

when Load is greater than the capacity of network then delay is ∞.

(b) Throughput Vs Load -

Throughput means no. of packets passing through the network per unit time.



when Load is below the capacity then throughput increases with Load.

when Load reaches the capacity of network then throughput decreases sharply because the queue becomes full & the router discard packets.

Congestion Control →

Congestion control mechanism can either prevent congestion before it happens or remove congestion after it happens. There are 2 different types of congestion control mechanism. They are -

(a) open loop congestion control.

(b) close loop congestion control.

Congestion Control

- Open Loop congestion Control
  - → Retransmission policy.
  - → window policy.
  - → Acknowledgement policy.
  - → Discarding policy.
  - → Admission policy.
- Close loop Congestion Control
  - → Back Pressure
  - → Choke packet
  - → Implicit Signaling
  - → Explicit Signaling

Open Loop Congestion Control –

It is applied for preventing congestion. Therefore open loop congestion control is applied before congestion has happens. Different types of congestion control for open loop are –

1. **Retransmission policy →**

If a send packet is lost or corrupted then it need to be retransmission which leads to congestion in network. This is because resending several packets again due to lost or damaged increase data traffic that- consequently increase the chance of congestion. Hence a good retransmission policy & retransmission timer can prevent congestion.

2. **Window policy →**

window size can also create congestion. In Go-Back-N ARQ the window size is more at sender side than Selective Repeat ARQ. So chances of congestion is more in Go-Back-N ARQ. This is because in Go-Back-N ARQ when timer of a packet times out then several no. of packets may be resend than selective repeat- ARQ. And these duplicate packets due to resending process Leads to congestion.

3. **ACKnowledgement policy →**

ACKnowledgement process slow down transmission or data sending process. So it prevents congestion. Also we send only one cumulative ACK for 'N' no. of packets. Hence imposing less ACK packets on network will leads to less congestion.

4. **Discarding policy →**

Discarding policy choosen by routers to discard overwhelmed packets to prevent congestion but the integrity or quality of message should be maintain -ed.

5. **Admission policy →**

Admission policy can prevent congestion by checking resource requirement before establishing connection

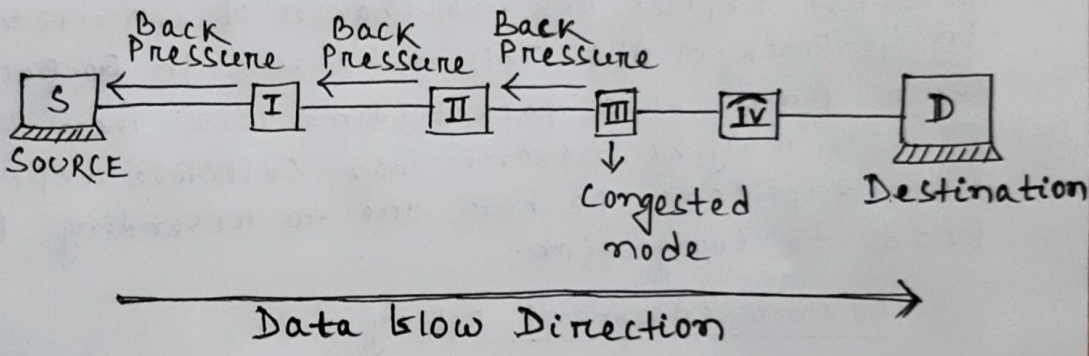
A connection can not be established if there is congestion in network.

Closed Loop Congestion Control →

Closed loop congestion control is applied if congestion has already happened in network. Therefore closed loop congestion control is applied for removal of congestion. Different mechanisms of closed loop congestion control are –

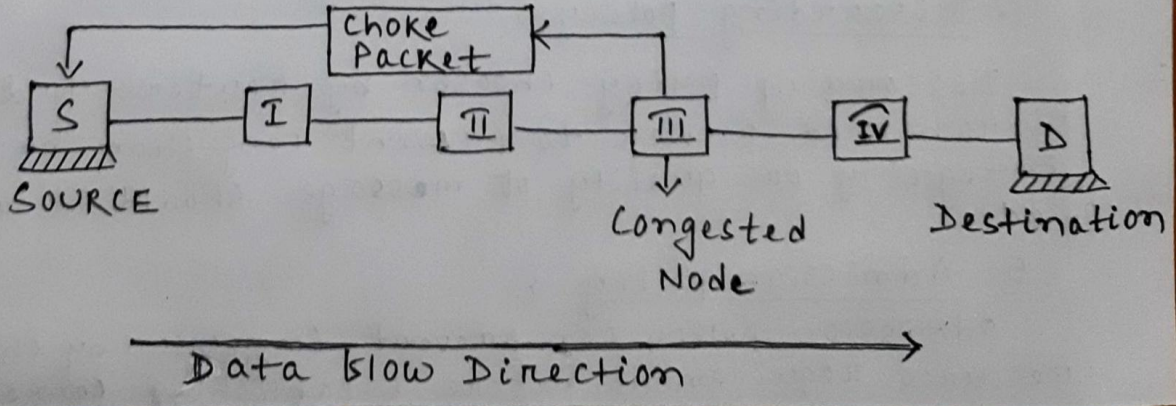
1. Back pressure →

On this mechanism a congested node stops receiving data packets from immediate upstream nodes. (And a upstream node is one from which a flow of data is coming). Back pressure mechanism cause congestion in upstream one. But this mechanism can cause to control congestion that start with congested node & propagate in opposite direction of data flow to the source.



2. Choke packet →

It is a packet sent by a congested node to the source directly to inform the source of congestion. The intermediate nodes through which packets has travelled are not warned.



3. Implicit signaling →

In this mechanism, the congested node doesnot inform to the source rather the source guesses that there is a congestion somewhere in the network from other symptoms. e.g - when congestion occur packets are discarded & no ACK come to the source.

4. Explicit signaling →

In this mechanism the congested node can explicitly send a signal to the source or the destination and the explicit signal is included in the data packet. Explicit signaling is of 2 types -

(a) Backward signaling

(b) Forward signaling

Backward signaling

→ If the explicit signal bit set in the data packet move in the opposite direction of the congestion to inform or warn the source about congestion, then it is called as backward signaling.

→ It slow down data packet sending.

Forward signaling

→ If the explicit signal bit set in the data packet move in same direction of congestion to warn the destination about congestion then it is known as forward signaling.

→ It slow down ACK sending process.

Quality of Service (QOS)

Quality or integrity of data flow is described by the flow characteristics. Different flow characteristis are -

1) Reliability
2) Delay
3) Jitter
4) Bandwidth

1. Reliability –

Reliable data transmission provides error free & successful data transmission. But lack of reliability means loosing data packets & ACK.

2. Delay –

Different application program can tolerate different degrees of delays e.g – audio conference & video conference can tolerate very low degrees of delay or minimum delays. But FTP or e-mail can tolerate maximum delays.

3. Jitter –

Jitter is variation in arrival time of data packet or Jitter is variation in delay for packets belonging to same message. For some message it is acceptable & for some message it is not acceptable. e.g – for real time application like LIVE telecast, jitter can never be acceptable.

4. Bandwidth –

Different application need different bandwidth For e.g – video conferencing need more bandwidth & e-mail need less bandwidth for transmission.

Techniques to improve QOS :–

(a) Scheduling
(b) Traffic shaping
(c) Admission control
(d) Resource reservation

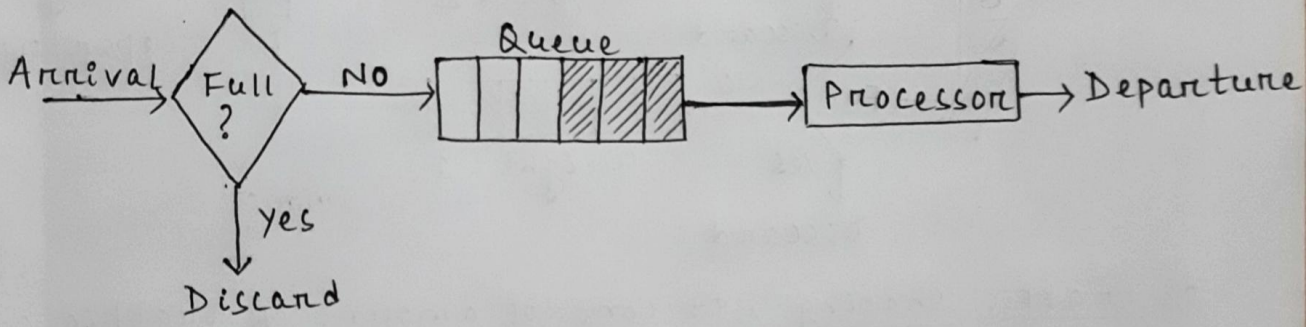
a) Resource Reservation → QOS improved if the resources are reserved prior to the data communication. e.g – of resources are – buffer or queue, bandwidth.

b) Admission control → Admission control refers to the mechanism used by a router or a switch to accept or reject a flow based on predefined parameter called flow characteristics.

c) Scheduling → Packets from different flow arrive at a switch or a router for processing. To treat different flows in a fair manner there are different scheduling mechanisms in QoS. They are -

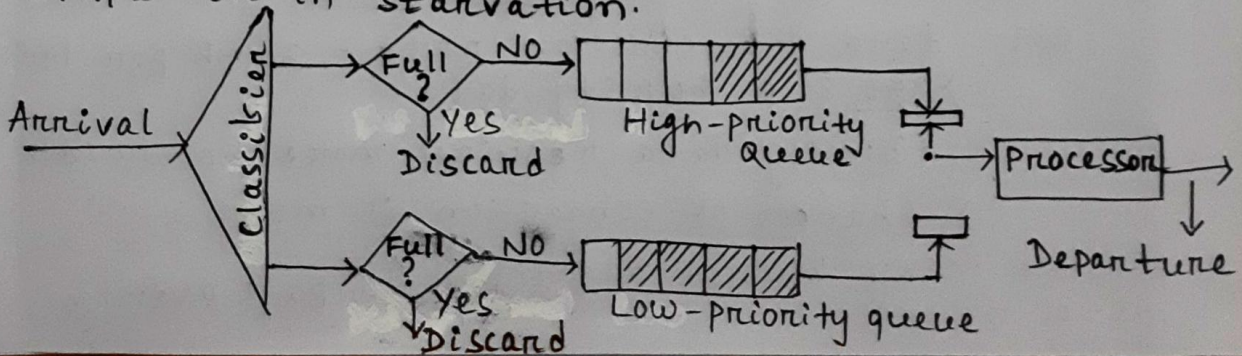
1. FIFO Queuing
2. Priority Queuing
3. Weighted - Fair Queuing

1. FIFO Queuing - Packets wait in a queue or buffer until they are processed by node. If the average arrival rate is higher than the average processing rate then the queue will filled up & new packets will be discarded.



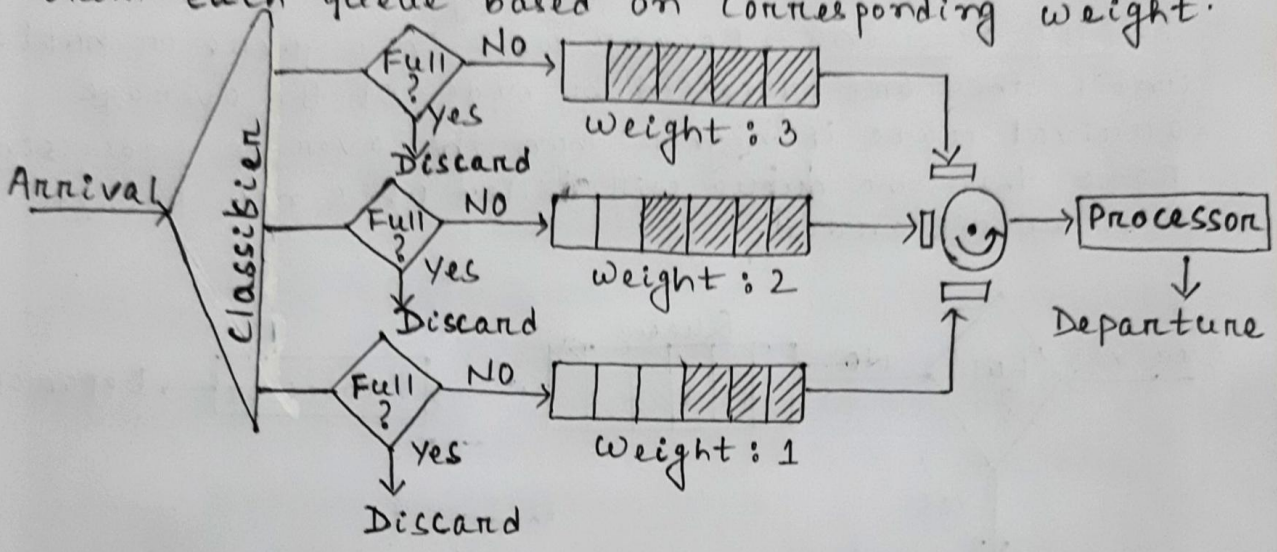
2. Priority Queuing - In this mechanism packets are first assigned to a priority class. Each priority class has its own queue. So packets in the high-priority queue are processed first than lowest-priority queue. Therefore, the system doesnot stop serving queue until it is empty. Priority Queuing Provides better quality of service than FIFO Queuing.

Drawback - If there is a continuous flow of data Packets from high priority queue then the data Packets from low priority queue always is in wait state or in starvation.



3. Weighted-fair Queuing - In this technique, packets are assigned to different class & assigned to them different queues. The queues are weighted based on priority of queues. [Highest priority has highest weight & lowest priority has lowest weight.]

System process packets from each queue in a round robin fashion with no. of packets selected from each queue based on corresponding weight.



d) Traffic Shaping → It control amount of traffic or rate of data traffic on the network. There are two different types of traffic shaping. They are -

1. Leaky Bucket algorithm.
2. Token Bucket algorithm.

1. Leaky Bucket algorithm -

It shapes bursty data into fixed data rate by averaging the data rate. It may drop packets if the bucket is full.

It smooth bursty data traffic i.e. the input data rate may vary but output data rate remain constant. Bursty data stored in a bucket (buffer) & sent out at an avg. data rate

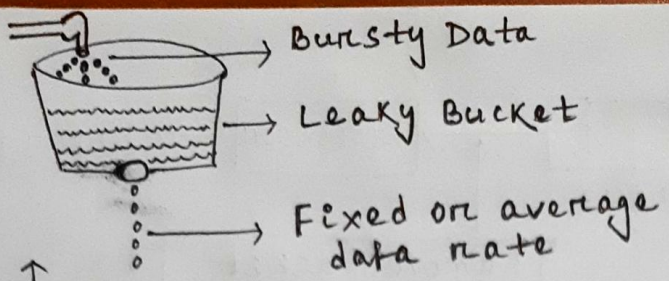
e.g. - data transmission rate = 30 Mb per 10 sec

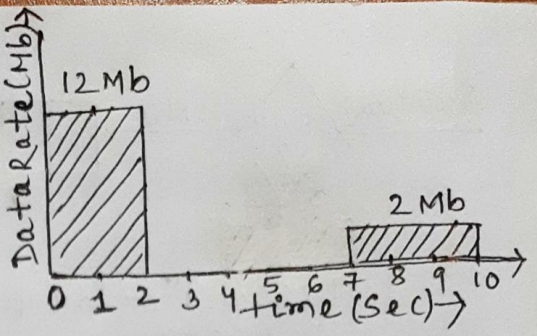
⇒ It is a bursty data.

⇒ Bursty data requires more bandwidth

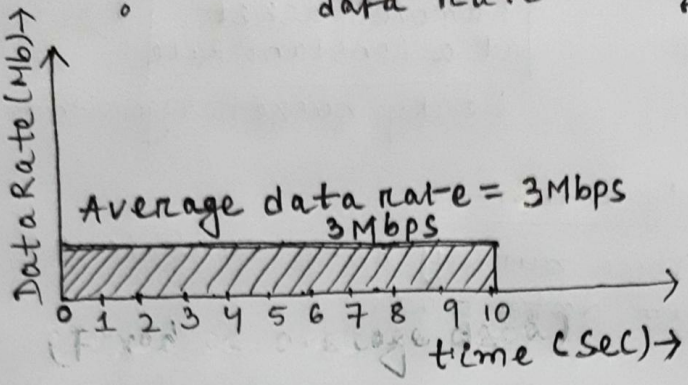
⇒ Chances of congestion is more

So, use leaky bucket algorithm to pass average data rate.





Total Data flow =
12 Mb x 2 sec = 24 Mbps
2 Mb x 3 sec = 6 Mbps
30 Mbps



(Bursty Data)

Total data flow = 3 Mb x 10 sec
= 30 Mbps.

(Fixed data rate or average data rate)

Leaky Bucket implementation →

Queue present for leaky bucket is FIFO.

→ If data traffic consist of fixed-size packet then the leaky bucket algorithm removes fixed no. of packet from the queue at each clock tick.

→ If data traffic consist of variable-size packets then there is fixed output data rate based on no. of bytes or bits.

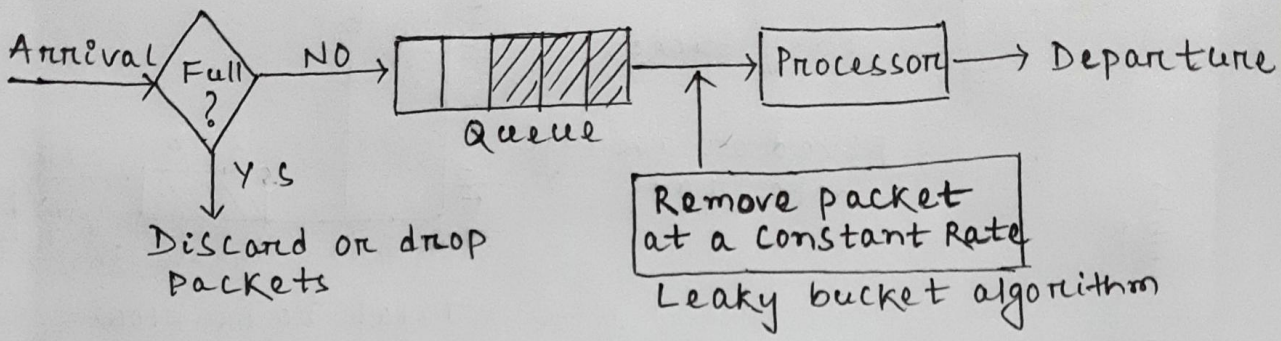
→ Algorithm for variable length packet –

Step 1. Initialize counter to 'n' at each clock tick

Step 2. If n ≥ packet size, then send a data packet & decrement the counter by packet size. Repeat step 2 until 'n' is smaller than the packet size.

Step 3. Reset the counter & go to step 1.

→ Drawback – when there is bursty data then there is average data flow but when there is no data or for idle host, this algorithm is not taken into account.



Leaky bucket algorithm

2. Token - Bucket algorithm -

Token bucket allows bursty data traffic at a regulated maximum data rate.

Algorithm 1 -

1. For each tick of clock, the system send 'n' no. of tokens to the bucket. (System send one token for every byte of data send. Here system mean algorithm of token bucket)

So, bursty data can send as long as token of bucket is not empty.

e.g. $n = 100$ (no. of token)

no. of ticks = 100

$\Rightarrow$ Bucket collect tokens = $100 \times 100 = 10,000$

Algorithm 2 -

Step 1. Initialize token to '0'. Each time a token is added the counter is incremented by 1. (Initialize counter to '0').

Step 2. Each time a unit of data is sent then the counter is decremented by 1. When the value of counter is '0' then host or system can not send data further.