Lecture #

Introduction

This section deals with Quality of Service & algorithms Leaky Bucket algo & Token Bucket algo r discussed.

Concept 1

Quality of Service (QoS)

→ QoS r seriously considered ∝ & maintained thro n/w

→ to maintain QoS protocols r designed.

Requirements.

- a stream of pkts frm S - D → flow.
- In connection oriented s/w - pkts follow the same route.
- In connectionless s/w - diff pkts follow diff. routes.
- Need of each flow is characterized by primary parameters reliability, jitter, delay, bandwidth. — all these together determine the QoS.

Common Appln's	R	D	Jitter	Bandwidth
email	high	L	L	L
file transfer	H	L	L	M
Web access	H	M	L	M
Telephony	L	H	H	L

audio/video
video conf

Techniques 4 achieving good QoS

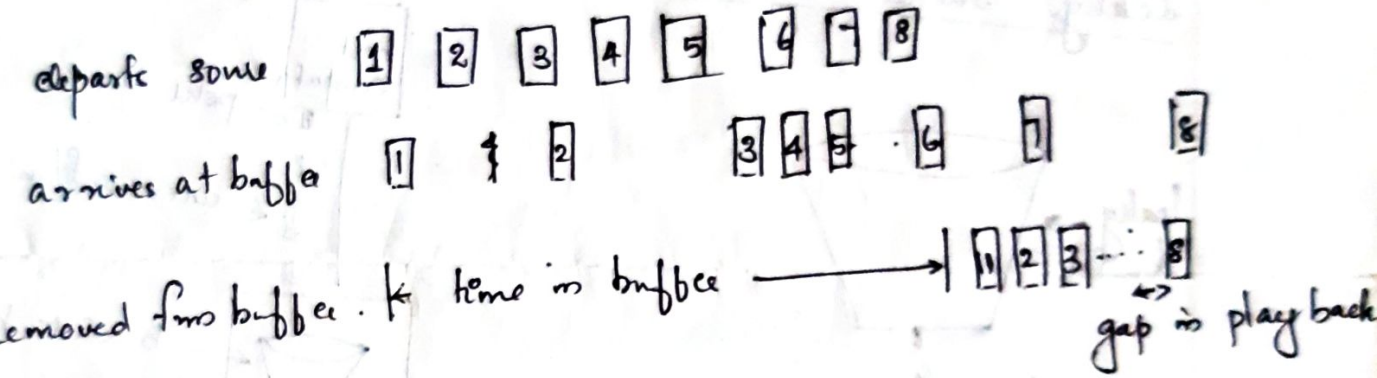
Overprovisioning

- provide so much router capacity, buffer space & bandwidth so that pkts can fly thru it easily.
- very expensive.

Buffering.

- Flows can be buffered on the receiving side b4 delivering.
- doesn't affect reliability / bandwidth
- ↑ delay, but smoothens jitter.

Audio & video - jitter is a serious issue so this method can be used.



eg:- Smoothing the o/p stream by buffering packets.

Traffic Shaping

- main reason 4 congestion is traffic.
- open loop method is used to manage congestion is forcing the pckts to be transmitted at a more predictable rate -
- TS - regulating the avg. rate of data transmission.
- Sliding window protocol - reduce the amount of data in transit at once. but not rate.

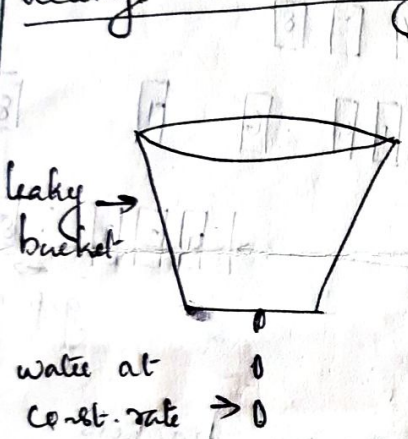
- When a connection is set up, the user and the subnet agree on a certain traffic pattern (shape)

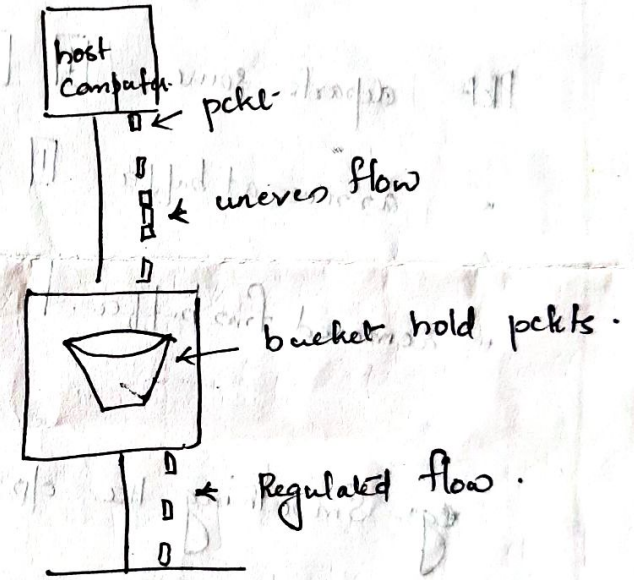
↓

service level agreement.

- Monitoring a traffic flow - traffic policing.
- agree to traffic shape & policing - easy in virtual ckt than datagram.

Leaky Bucket Algorithm.





- each host is connected to the n/w by an interface - leaky containing a leaky bucket.
- proposed by Turner in 1986.
- I/F → LBA consists of a finite internal queue.
- when pckt comes, it is appended to the queue if there is enough space.

- else discarded.
- host is allowed to put one pckt / clock time into the n/w.
- LBA turns an uneven flow of pckts from the user processes inside the host to an even flow of pckts on to the n/w.
- When the pckts is of same size the above algo is used.
- for variable sized pckts, allow fixed no: of bytes is sent / clock tick.
  eg:- 1024 B (2 x 512 / 4 x 256 B).
- Here for each tick, a counter is initialized to n.
- If the 1st pckt on queue has fewer bytes than the current value of counter, pckt is transmitted and counter is decremented by that no: of bytes.
- additional pckts may also b sent as long as counter is high enough.
- When the counter drops below the length of the next pckt on the queue, transmission stops until the next tick, the residual byte count is reset & flow can continue.

Token Bucket Algorithm.

- leaky bucket donot holds a token; but token bucket algo holds a token.
- token generated by clock at every ΔT sec.
- for a pkt to be transmitted must capture token & destroy.

one token.

Diff.

LB

- LB enforces a rigid o/p pattern at avg. rate
- don't consider the traffic.
- doesn't allow the idle hosts to save up permission to send large bursts later.

discards pkt.

TB

+ suitable if large bursts occur.

allow saving upto the max. size of bucket (n) ie burst of upto n pkts can be sent at once.

TB throws away tokens when bucket is full but never discards pkt.

TBA → pkts can be sent only if tokens r available.
- ↳ count var → incremented by one who new token generated
- ↳ decremented by 1 when a pkt is sent