CMPN415 – CMP405B Computer Networks

Part FOUR

Internetworking – Quality of Service

Most Important Slides

Quality of Service

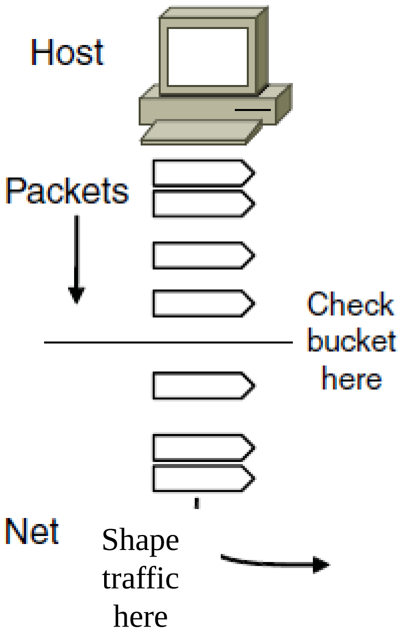
Approaches to Achieve “*Good*” QoS

1. Over-provisioning
2. Buffering
3. Traffic Shaping
4. Packet Scheduling
5. Admission Control
6. Resource Reservation

Usually more than one technique is used at the same time to satisfy the QoS requirements

3. Traffic Shaping

Traffic shaping regulates the average rate and burstiness of data entering the network



Objective of traffic shaping:

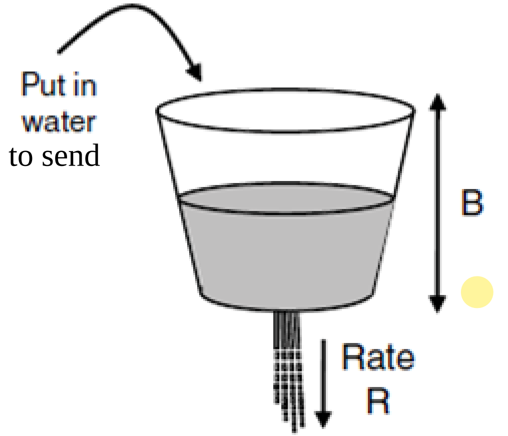
- Smoothing irregular traffic
- Regulating the average rate
- Limit burstiness
- Sometimes called ***traffic regulation***

Traffic regulation reduces congestion

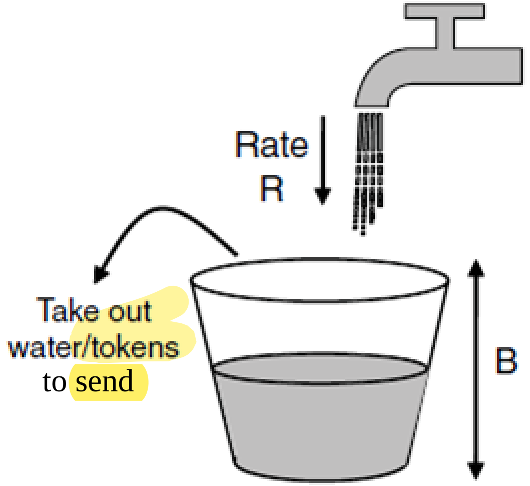
- Remember congestion is sometimes caused by temporary increase in generated traffic

Traffic Shaping

- Token/Leaky bucket limits both the average rate (R) and short-term burst (B) of traffic
- For token, bucket size is B, water enters at rate R and is removed to send; opposite for leaky.

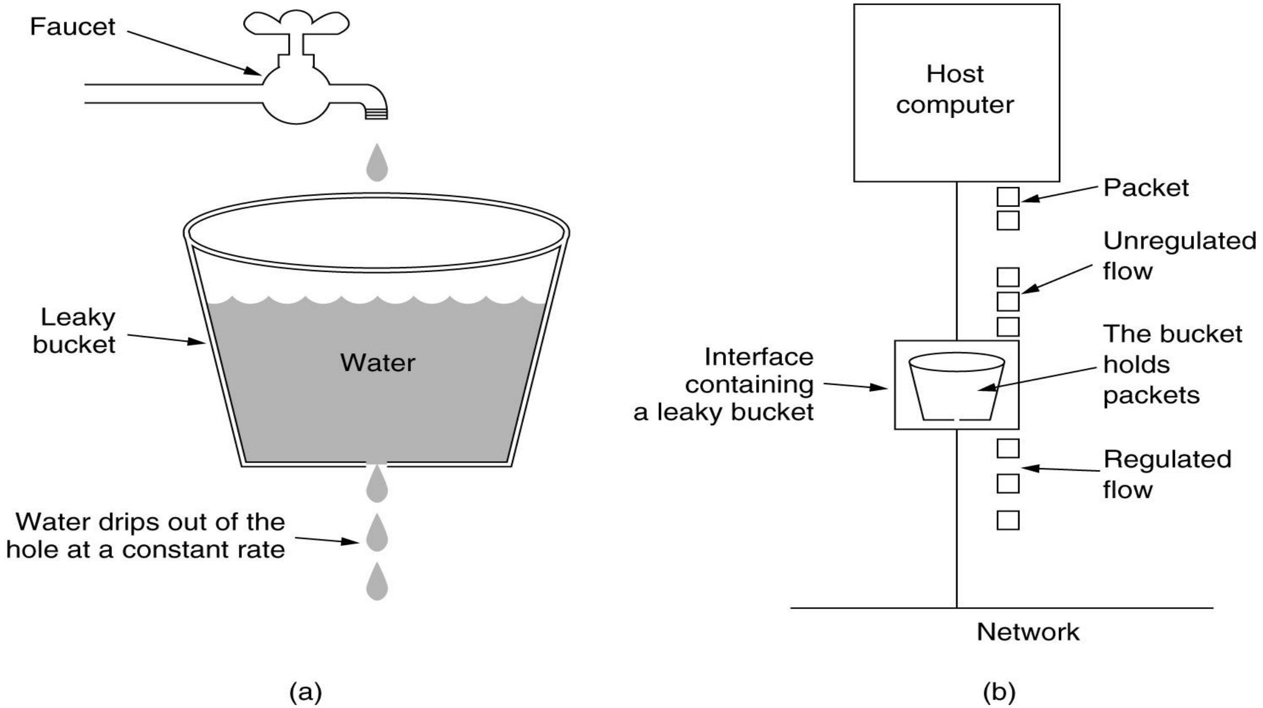


Leaky bucket
(need not full to send)



Token bucket
(need some water to send)

3.1 The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.

- Packet leaky bucket can be implemented using a queue
- Output rate is fixed at ***R*** bits/bytes/packets per second
- Packet **exceeding** the queue size are **discarded**
- Similar to a **single server** queuing system
- Enforces a **rigid** average rate with **no** burstiness

Leaky Bucket

Consider a source with a bucket capacity B of 1Mb and data input rate is ρ 25mbps. Calculate:

1. Time needed to fill the bucket
2. If the output rate R is 2mbps, time needed to empty the bucket

Tfill= B / ρ = 1/25= 40 msec **IN**

Tout= B / R **=** ½ = 500 msec **OUT**

3.2 The Token Bucket Algorithm

- Sometimes we
  - want allow *some* burstiness: e.g. *I* frames in an MPEG video stream
  - do not want to drop traffic

- Tokens are added to the bucket at a constant rate ρ **tokens/sec**

- **The bucket capacity is B tokens**

- If the bucket is full, **Tokens are thrown away**

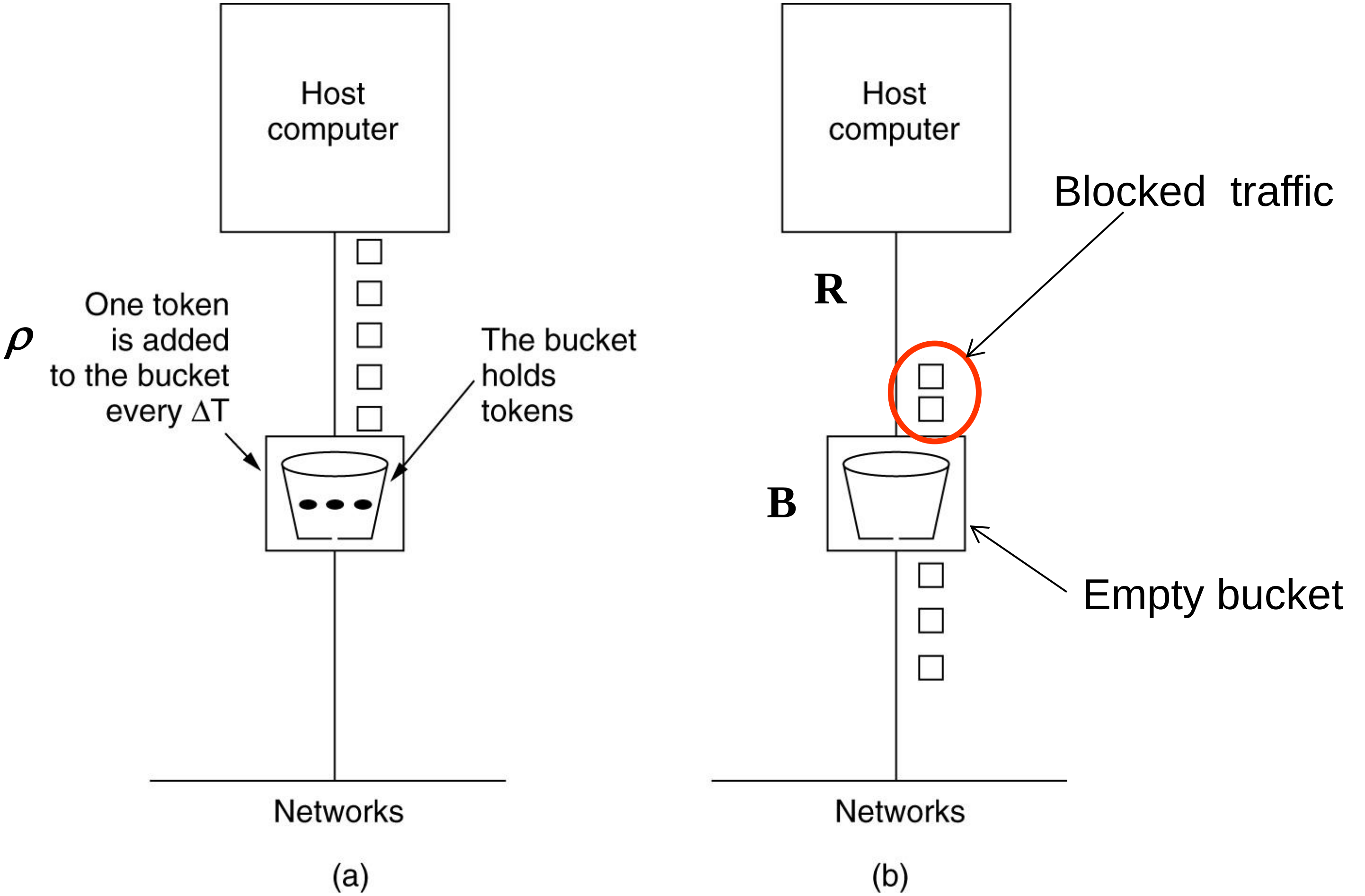
- A packet must consume a token to get transmitted

- If the bucket is empty
  - packets are NOT dropped, instead the application is blocked
  - Packets may be drop in case of a router because a router cannot block a remote host
  - Dropping packets by a router is sometimes called ***policing***

- As long as a packet has a token, it gets transmitted at the **maximum** possible rate **M**.

- Allows a burst **period** of up to ***S*** seconds
- During the burst period, traffic is transmitted at the *maximum* possible rate

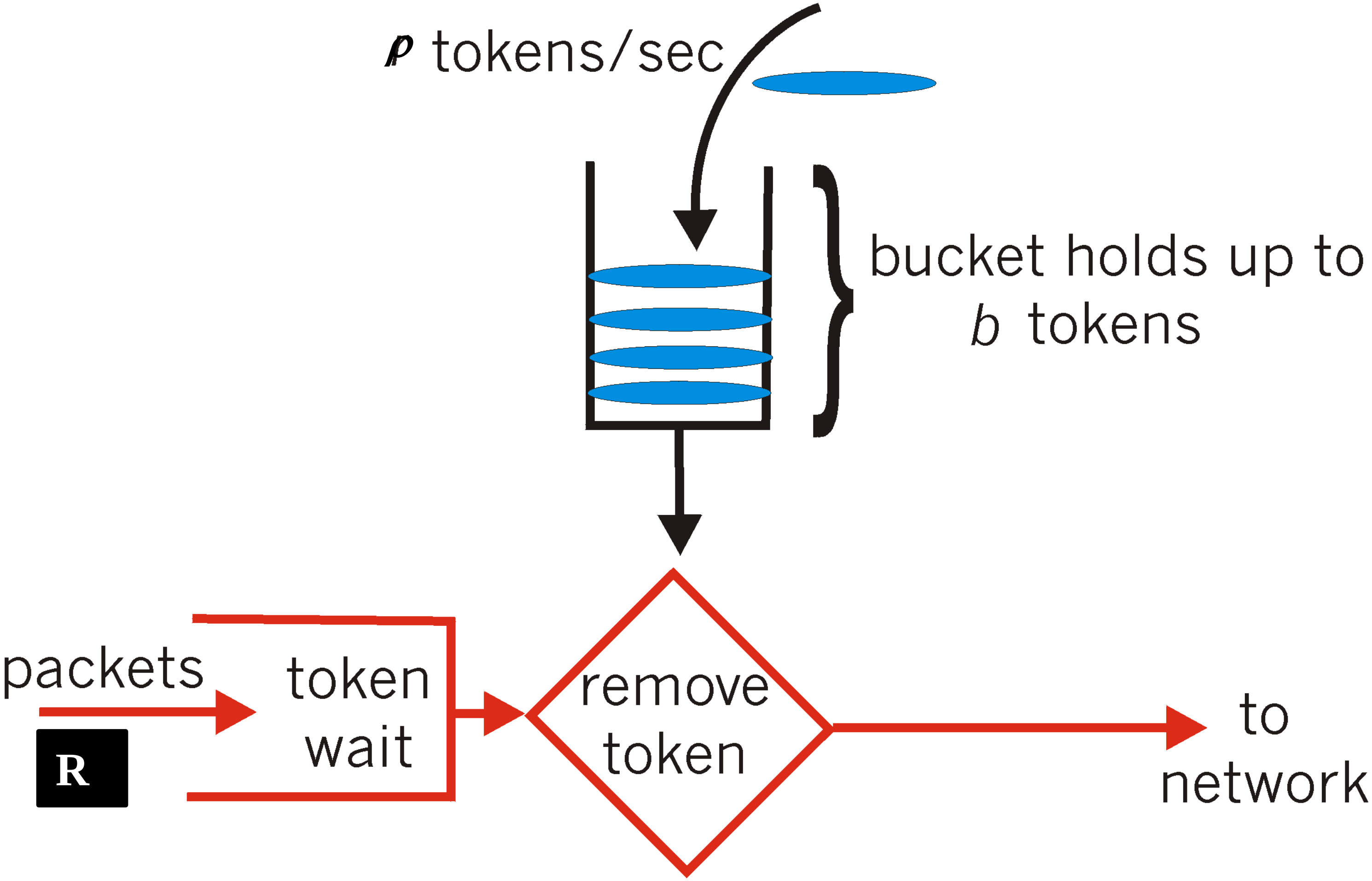
The Token Bucket Algorithm



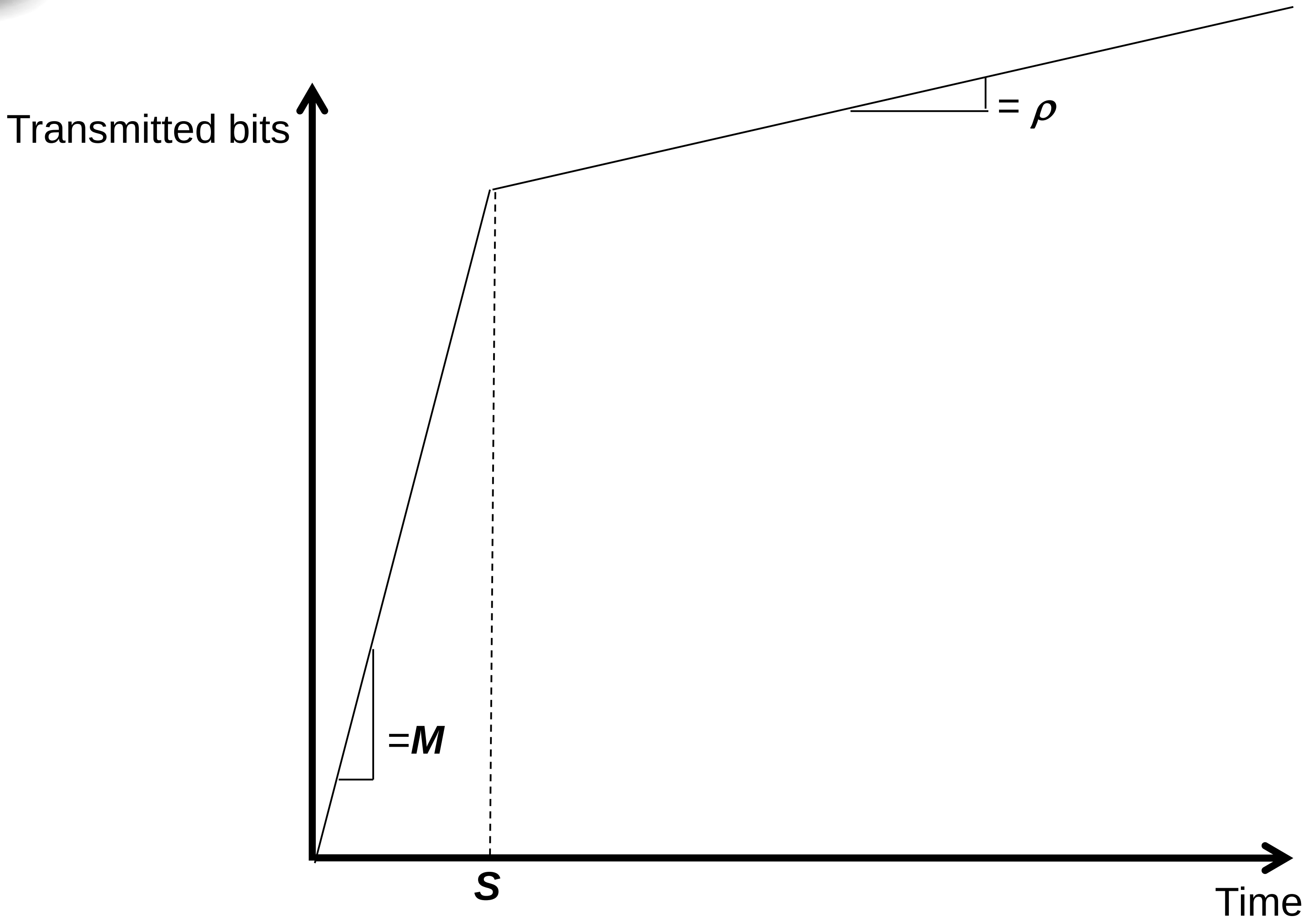
(a) Before starting to transmit

(b) After emptying the bucket.

Token Bucket Regulator (Shaper)



Token Bucket: Traffic Profile



During the *burst* period: $b + \rho S = MS$

***M* Link Capacity, *b* bucket size, ρ token rate, *S* burst**

Token Bucket Example

Consider a network with a capacity C of 1Mb (1000kb) and data is arriving at rate of R 25mbps for 40 msec. Token arrival rate ρ is 2Mbps, capacity of the bucket B is 500 kb, maximum output rate is 25mbps. Calculate:

1. Burst length
2. Total transmission time

B=capacity of bucket=500kb, M=25mbps, ρ ***= 2Mbps***

$$\boldsymbol{B + \rho S = MS}$$

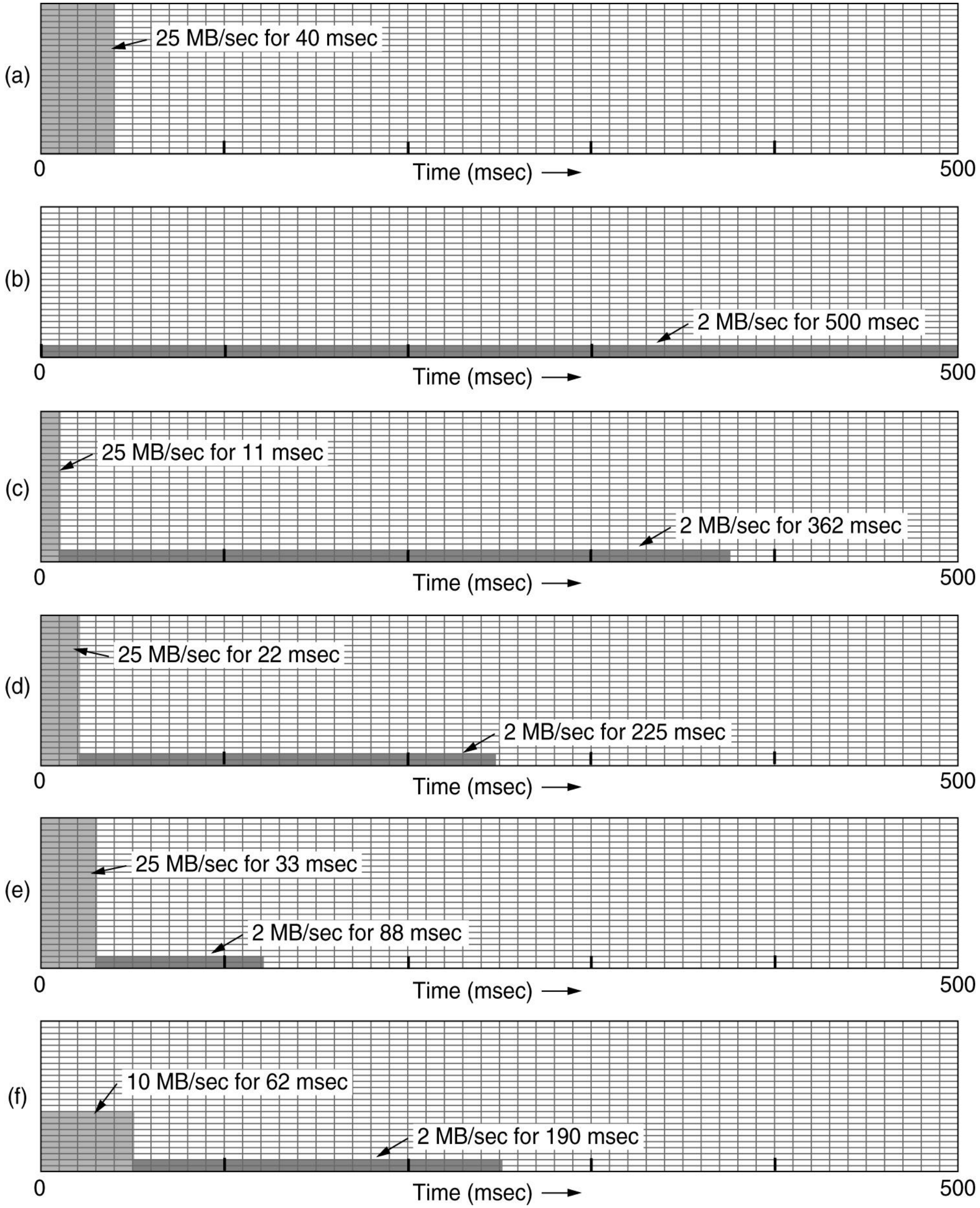
- Total data to transfer = 25*1000*(40/1000)= 1000 kb
- S burst time = B / (M- ρ***) = 500/ ((25 - 2)*1000)= 22 msec***
- Data transferred during S= 22*25=550Kb
- Time after burst for C = (1000-550)*1000/(2*1000)=225 msec
- Total transmission time is 247 msec

Comparing Leaky and Token Bucket Algorithm

(a) Input to a leaky bucket.

(b) Output from a leaky bucket.

Output from a token bucket with capacities of (c) 250 KB, (d) 500 KB, (e) 750 KB, (f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket.



Leaky Bucket vs. Token Bucket

Leaky Bucket	Token Bucket
When the host has to send a packet, packet is thrown in bucket.	Bucket holds tokens generated at regular intervals of time.
Bucket leaks at constant rate	Bucket has maximum capacity.
Bursty traffic is converted into uniform traffic by leaky bucket.	If there is a ready packet , a token is removed from Bucket and packet is send.
In practice bucket is a finite queue outputs at finite rate	If there is a no token in bucket, packet can not be send.
Token independent	Token Dependent