# Congestion Control

Abusayeed Saifullah

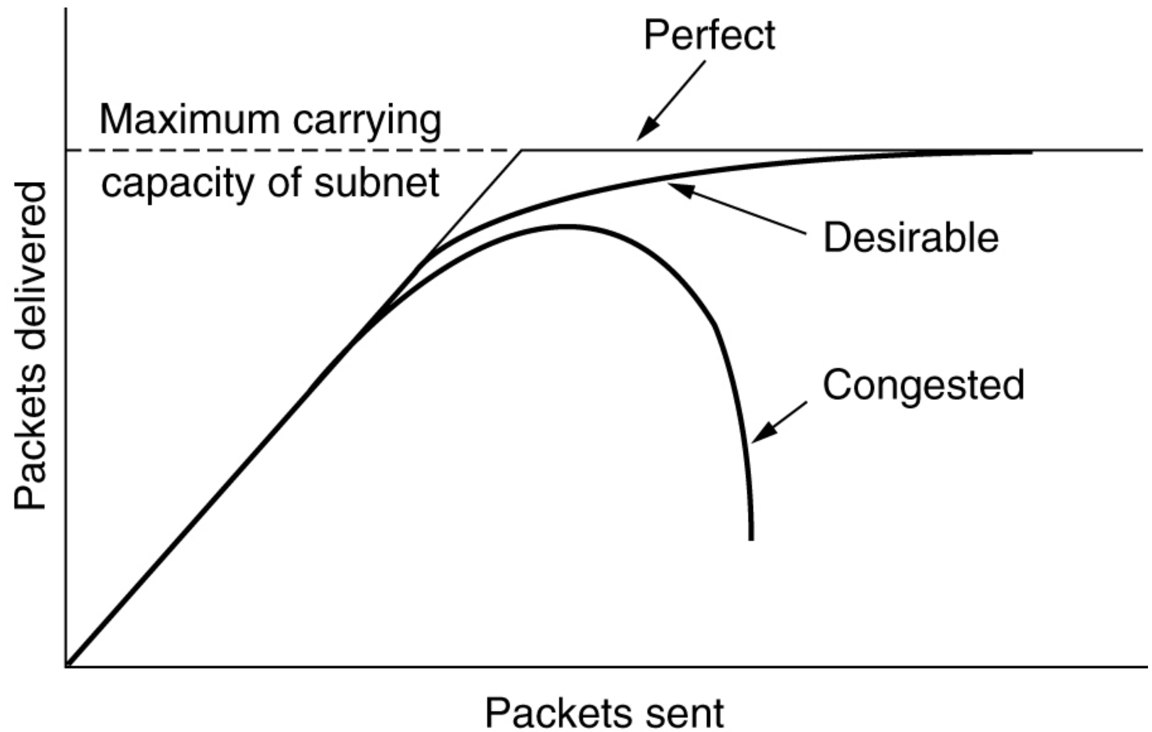
CS 5600 Computer Networks

# Network Congestion

- Congestion: When one part of the subnet (e.g. one or more routers in an area) is overloaded.
- The network and transport layers share the responsibility for handling congestion.
- It is the network layer that directly experiences congestion → network layer must take action.

Network congestion has similarity with congested road traffic.

# Congestion Effects




- Packet delay
- Packet loss
- Degraded performance

# Congestion Control

- Most effective way to control congestion is to reduce the load that the transport layer is placing on the network.
- This requires the network and transport layers to work together.
- In this lecture, we will look at the network aspects of congestion.
- We will complete the topic by covering the transport aspects of congestion later.

# Factors that Cause Congestion

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets
- Bursty traffic
- Slow processor

# Congestion Control vs Flow Control

- Congestion control is a global issue – involves every router and host within the subnet
- Flow control – scope is point-to-point; involves just sender and receiver.

# General Principles of Congestion Control

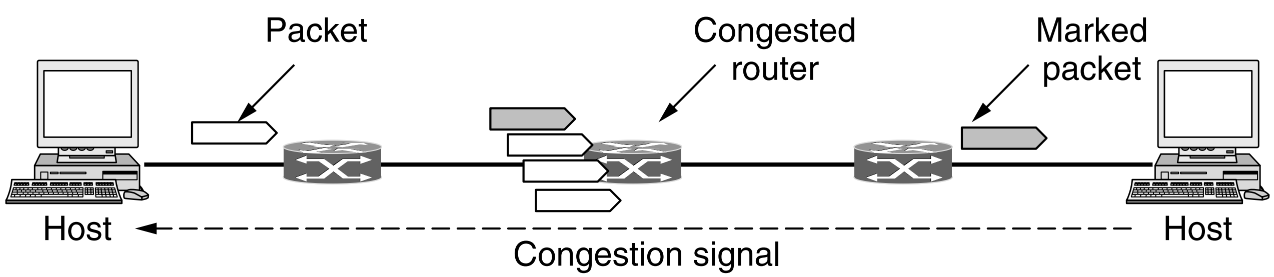
1. Monitor the system .
   – detect when and where congestion occurs.
2. Pass information to where action can be taken.
3. Adjust system operation to correct the problem.

# Congestion Control Techniques

- Congestion Control is concerned with efficiently using a network at high load.
- Several techniques can be employed. These include:
  - Warning bit
  - Choke packets
  - Load shedding
  - Random early discard
  - Traffic shaping
- The first 3 deal with congestion detection and recovery. The last 2 deal with congestion avoidance.

# Warning Bit

- A special bit in the packet header is set by the router to warn the source when congestion is detected.
- The bit is copied and piggy-backed on the ACK and sent to the sender.




- The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

# Choke Packets

- A more direct way of telling the source to slow down.
- A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
- The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
- An example of a choke packet is the ICMP Source Quench Packet.

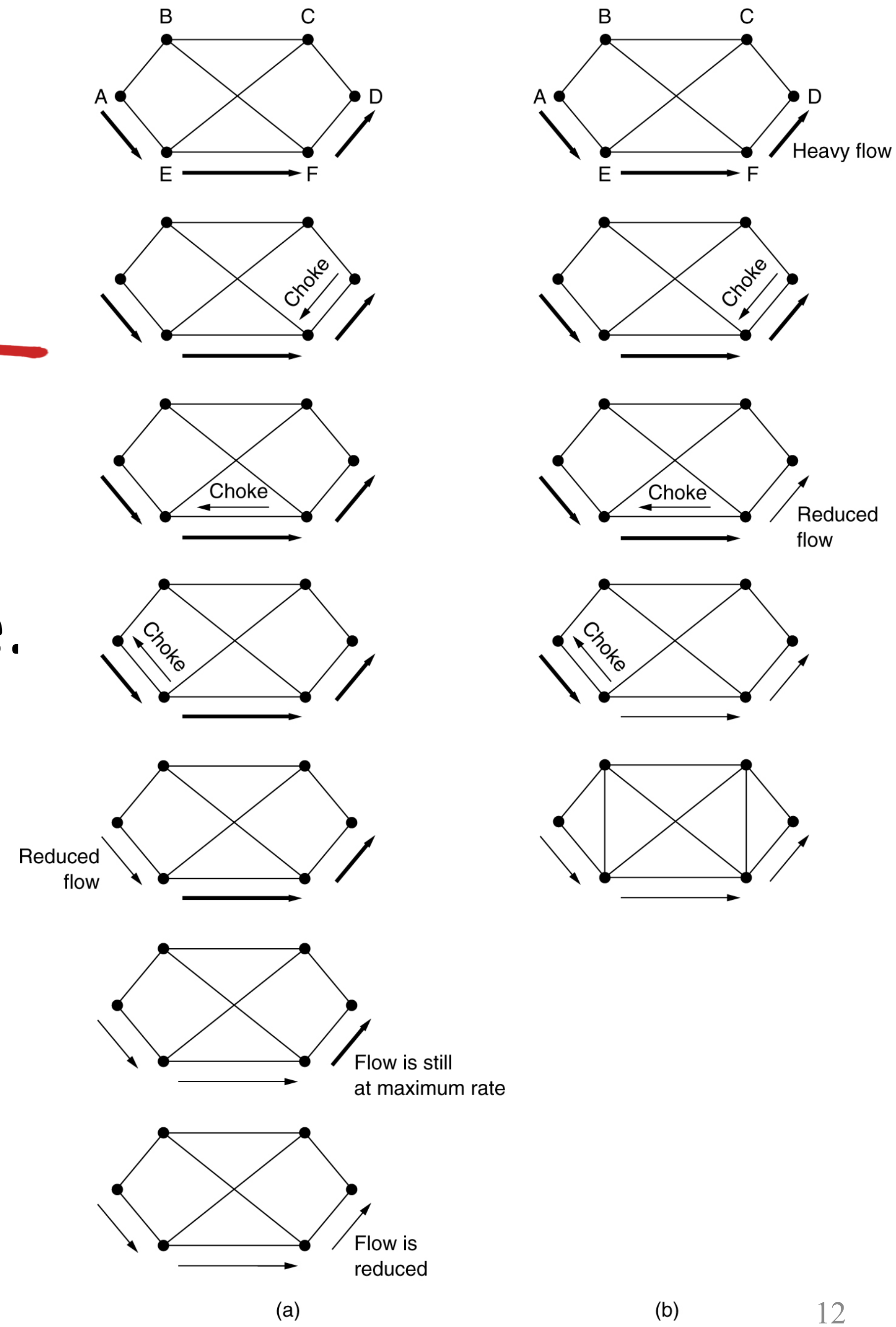
# Hop-by-Hop Choke Packets

- Over long distances or at high speeds choke packets are not very effective.
- A more efficient method: hop-by-hop backpressure.
- This requires each hop to reduce its transmission even before the choke packet arrives at the source.

# Hop-by-Hop Choke Packets

(a) A choke packet that affects only the source.

(b) A choke packet that affects each hop it passes through.



# Load Shedding

- When buffers become full, routers simply discard packets.
- Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
- Wine (old is better than new): e.g. a file transfer cannot discard older packets since this will cause a gap in the received data.
- Milk (new is better than old): e.g. For real-time voice or video, it is probably better to throw away old data and keep new packets.
- Get the application to mark packets with discard priority.

# Random Early Detection (RED)

- This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
- Each time a packet arrives, the RED algorithm computes the average queue length, ***avg***.
- If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.

# Random Early Detection (RED)

- If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
- If *avg* is between the two thresholds, this might indicate the onset of congestion. The probability of congestion is then calculated.

# Traffic Shaping

- Another method of congestion control is to "shape" the traffic before it enters the network.
- Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).
- The agreement is often called SLA (service level agreement):

  "*My traffic pattern will look like this; can you handle it?*"

# Traffic Shaping

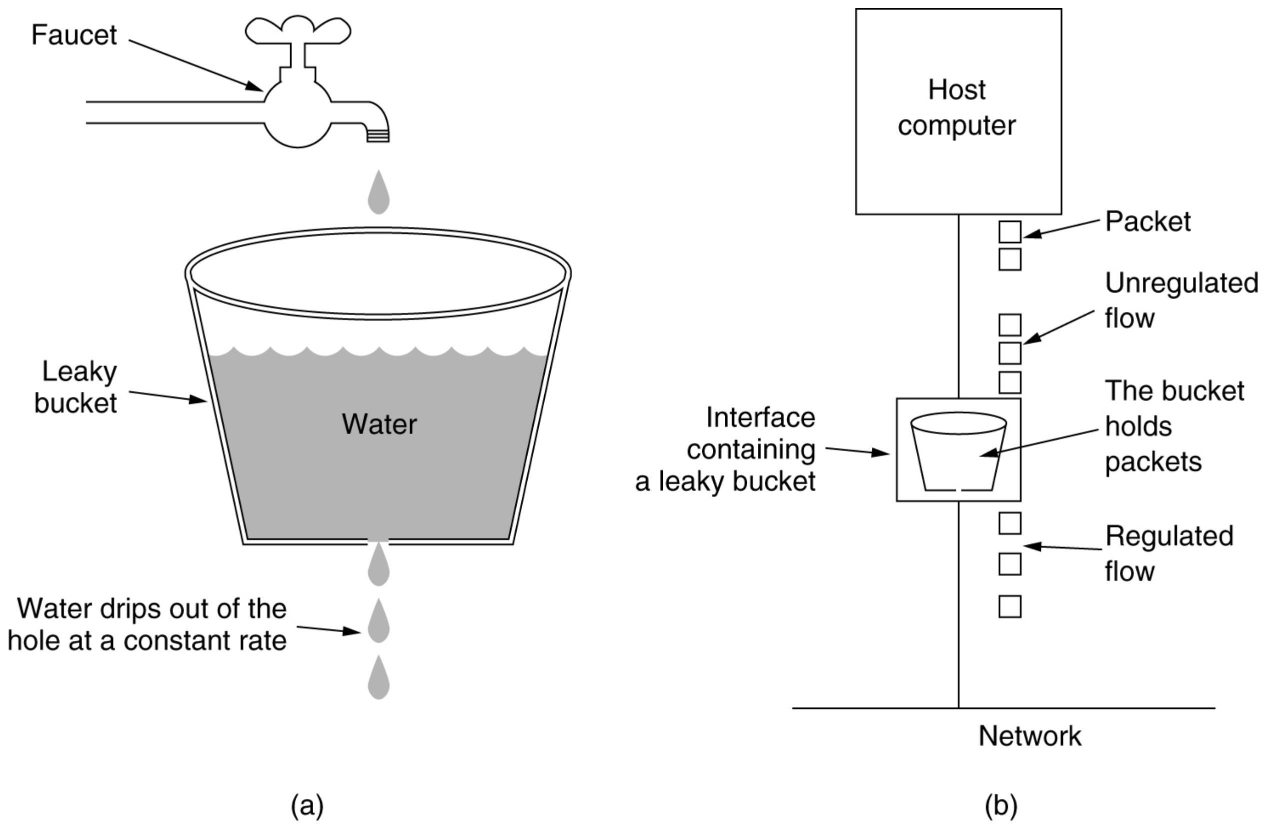
- What to do if SLA is not followed by sender
  - Packets in access of the agreed pattern might be dropped or less prioritized

- Two traffic shaping algorithms are:
  - Leaky Bucket
  - Token Bucket

# The Leaky Bucket Algorithm

- The **Leaky Bucket Algorithm** used to control rate in a network.
- It is implemented as a single-server queue with constant service time.
- If the bucket (buffer) overflows then packets are discarded.

# The Leaky Bucket Algorithm

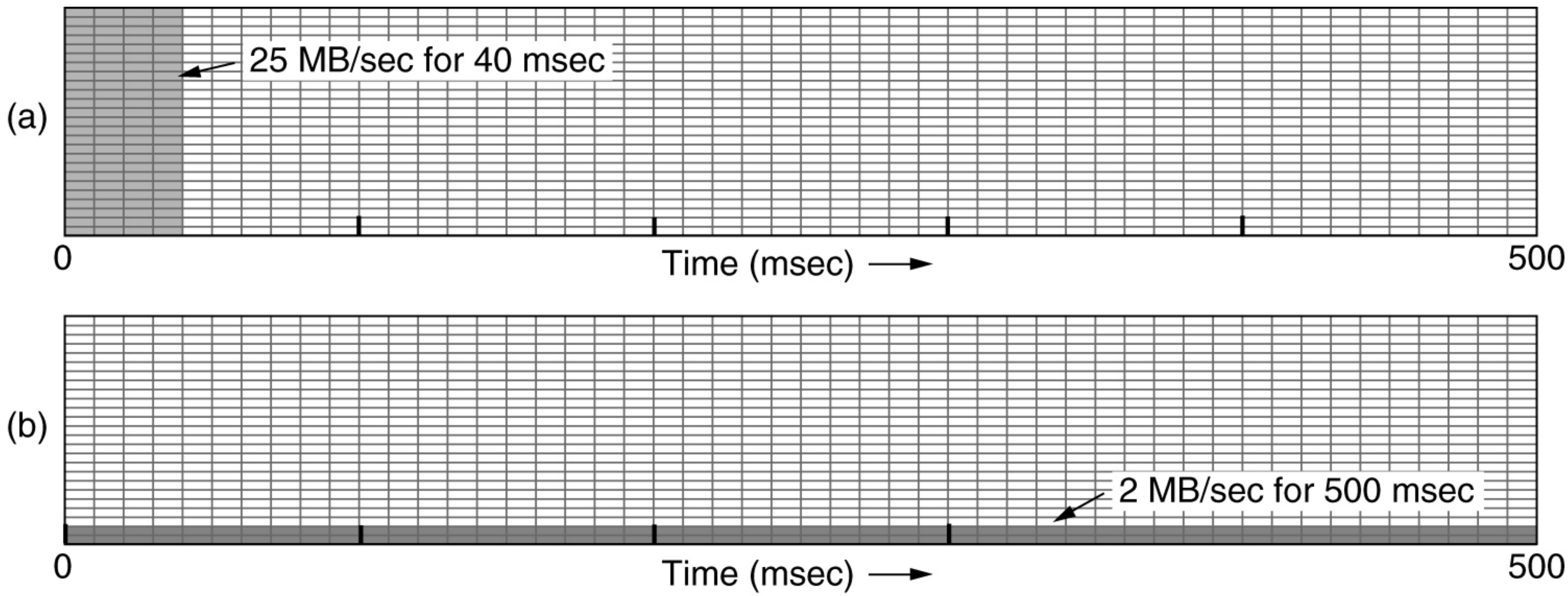



(a) A leaky bucket with water. (b) a leaky bucket with packets.

## Leaky Bucket (LB) Algorithm, cont.

- The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
- The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
- When packets have the same size (as in ATM cells), one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick. E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick.

# The Leaky Bucket Example




(a) Input to a leaky bucket. (b) Output from a leaky bucket

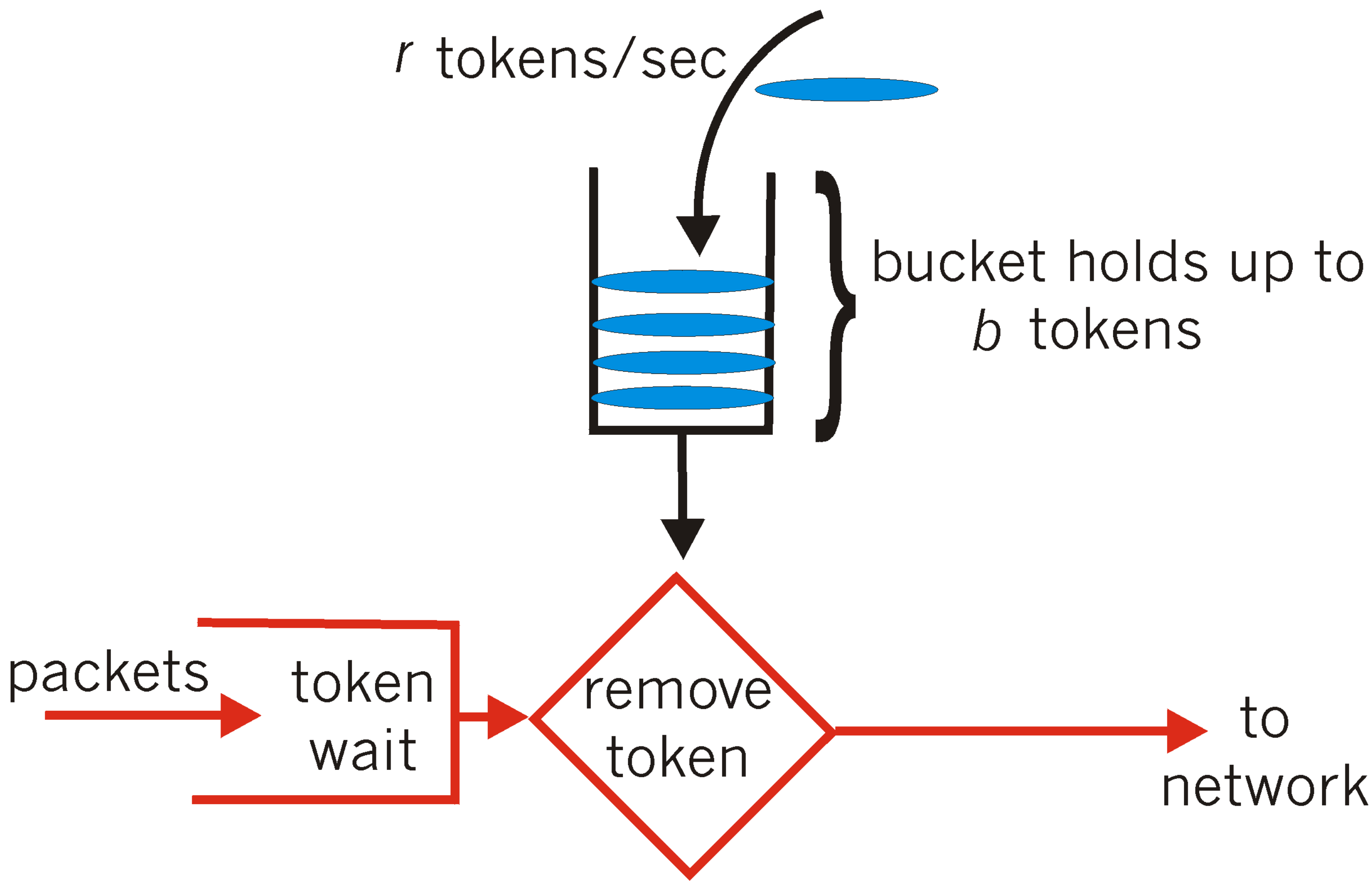
# Token Bucket (TB) Algorithm

- In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
- Token: a unit or fixed number of bytes or single packet of fixed size
  - We assume a token=a unit byte
- In the TB algorithm, the bucket holds tokens.
- To transmit a packet of n bytes, the host must capture and destroy n tokens.
- Number of tokens in the bucket → sending burst size

# Token Bucket (TB) Algorithm

- Tokens are generated by a clock at the rate r tokens per second (i.e. one token every 1/r sec).
- The bucket can hold at the most *b* tokens. If a token arrives when the bucket is full, it is discarded.
- If the number of tokens in the bucket is less than the number of bytes in a packet, no tokens are removed from the bucket, and no packet is injected
- (Initially or) Idle hosts can capture and save up b tokens in order to send larger bursts later.

# Token Bucket Regulator (Shaper)



# Remarks

- When tokens are not available, TB has 2 policies
  - Store packet (shaper)
  - Discard packet (policer)
- A flow as 2-tuple (input rate, output rate)
- The sum of two regulated flows $(b_1,r_1)$& $(b_2,r_2)$ behave like a regulated flow $(b_1+b_2,r_1+r_2)$
- Cascading TB after a LB can shape the burst rate