CONGESTION CONTROL

Congestion Control

- When one part of the subnet (e.g. one or more routers in an area) becomes overloaded, congestion results.
- Because routers are receiving packets faster than they can forward them, one of two things must happen:
  - The subnet must prevent additional packets from entering the congested region until those already present can be processed.
  - The congested routers can discard queued packets to make room for those that are arriving.

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.

Congestion Control, cont.

- 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 is to send to choke packets hop-by-hop.
- This requires each hop to reduce its transmission even before the choke packet arrive at the source.

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.
- For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.
- 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 Discard (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.

RED, cont.

- 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.

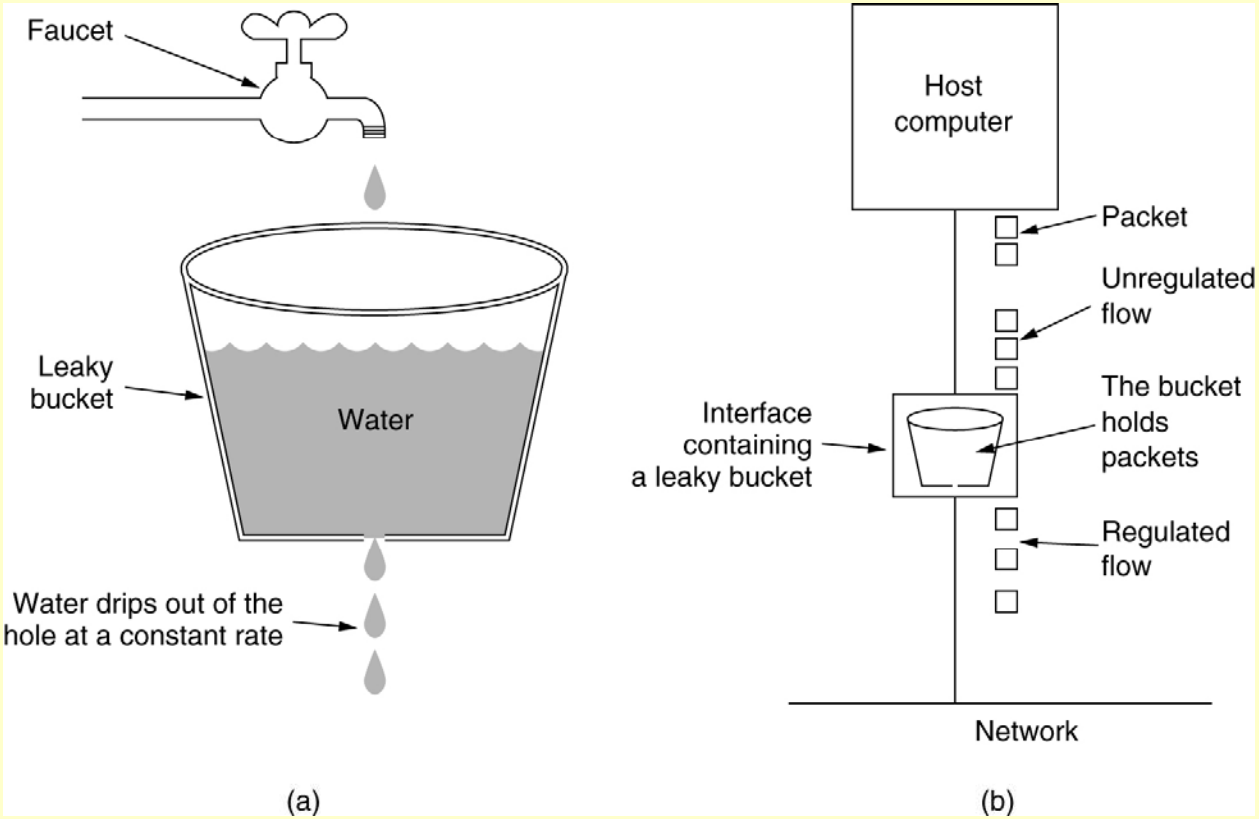
Traffic Shaping

- 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).
- 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.

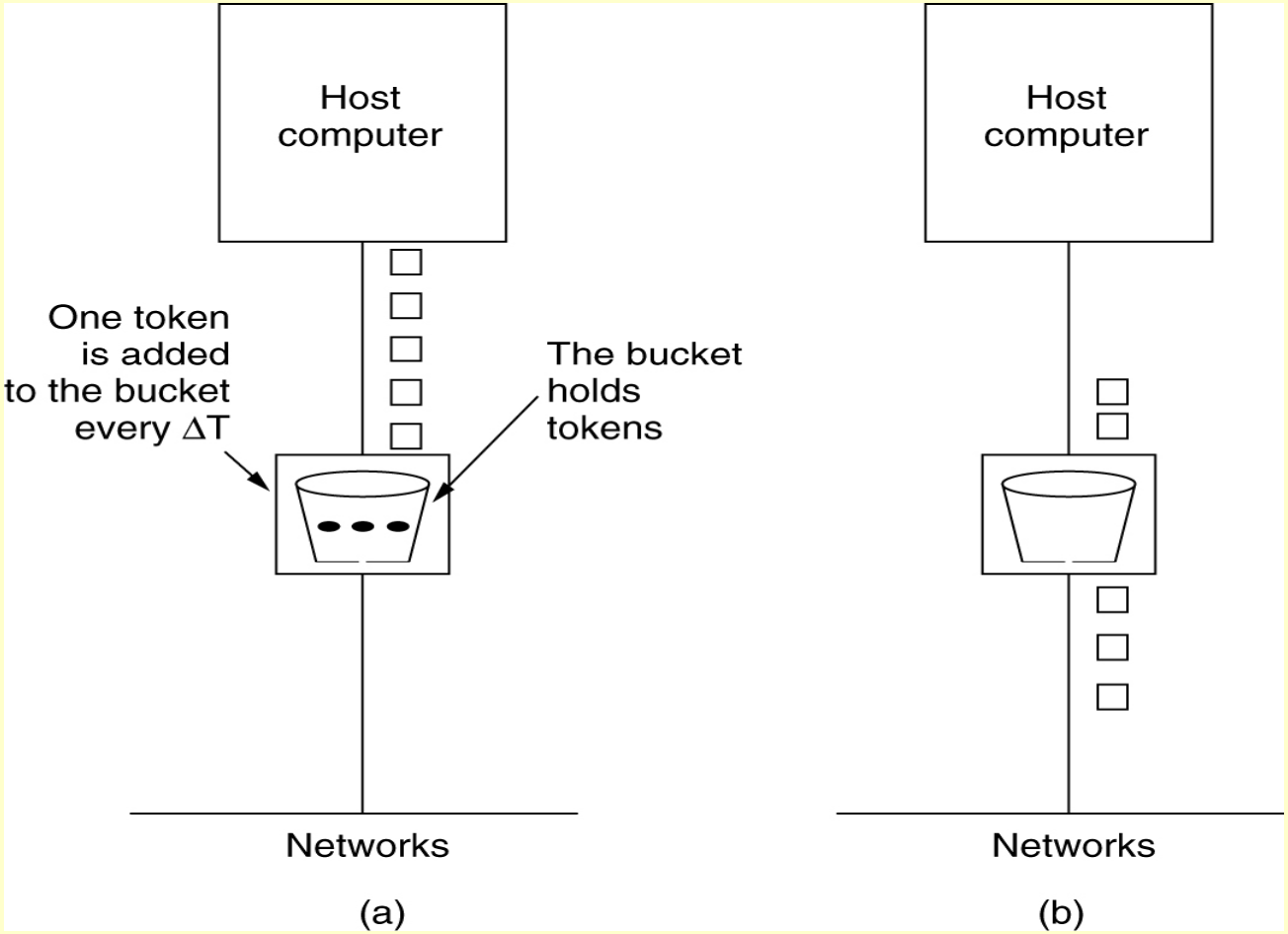
Leaky Bucket Algorithm, cont.

- 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 are the same size (as in ATM cells), the 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.

Token Bucket Algorithm

- In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
- In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.

The Token Bucket Algorithm



(a) Before. (b) After.

Leaky Bucket vs Token Bucket

- LB discards packets; TB does not. TB discards tokens.
- With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
- LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
- TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.