CSE2029: Data Communication & Computer Networks

Lecture-10: Transport Layer

Faculty: Dr. Sandeep Kumar

Outline

- *Transport layer: Congestion Control and Avoidance*
  - *Detection and Control*
    - *Warning bit*
    - *Choke packets*
    - *Load shedding*
  - *Avoidance*
    - *Random Early Discard*
    - *Traffic Shaping*

Congestion

- informally: “too many sources sending too much data too fast for *network* to handle”

- different from flow control which is an end-to-end issue!

- Consequences:
  - lost packets (buffer overflow at routers)
  - long delays (queue-ing in router buffers)

Transport Layer: Congestion Control and Avoidance

- 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

Detection and Control

Avoidance

- **The first 3 deal with congestion detection and Control. The last 2 deal with congestion avoidance.**

Congestion Detection and Control

The following 3 Methods are used to Detect & Control the Congestions :

1. **Warning bit**
2. **Choke packets**
3. **Load shedding**

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.

6

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.

Internet control mess. protocol.

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.
- In 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.
- Moreover, marked packets can be marked with discard priority by using some application/process.

Congestion Avoidance

The following 2 Methods are used to Avoid the Congestions :

1. **Random Early Discard**
2. **Traffic Shaping**

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.

Random Early Discard (RED) Cont'd…

- 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 and subsequently the queuing or discarding of the packet is decided.

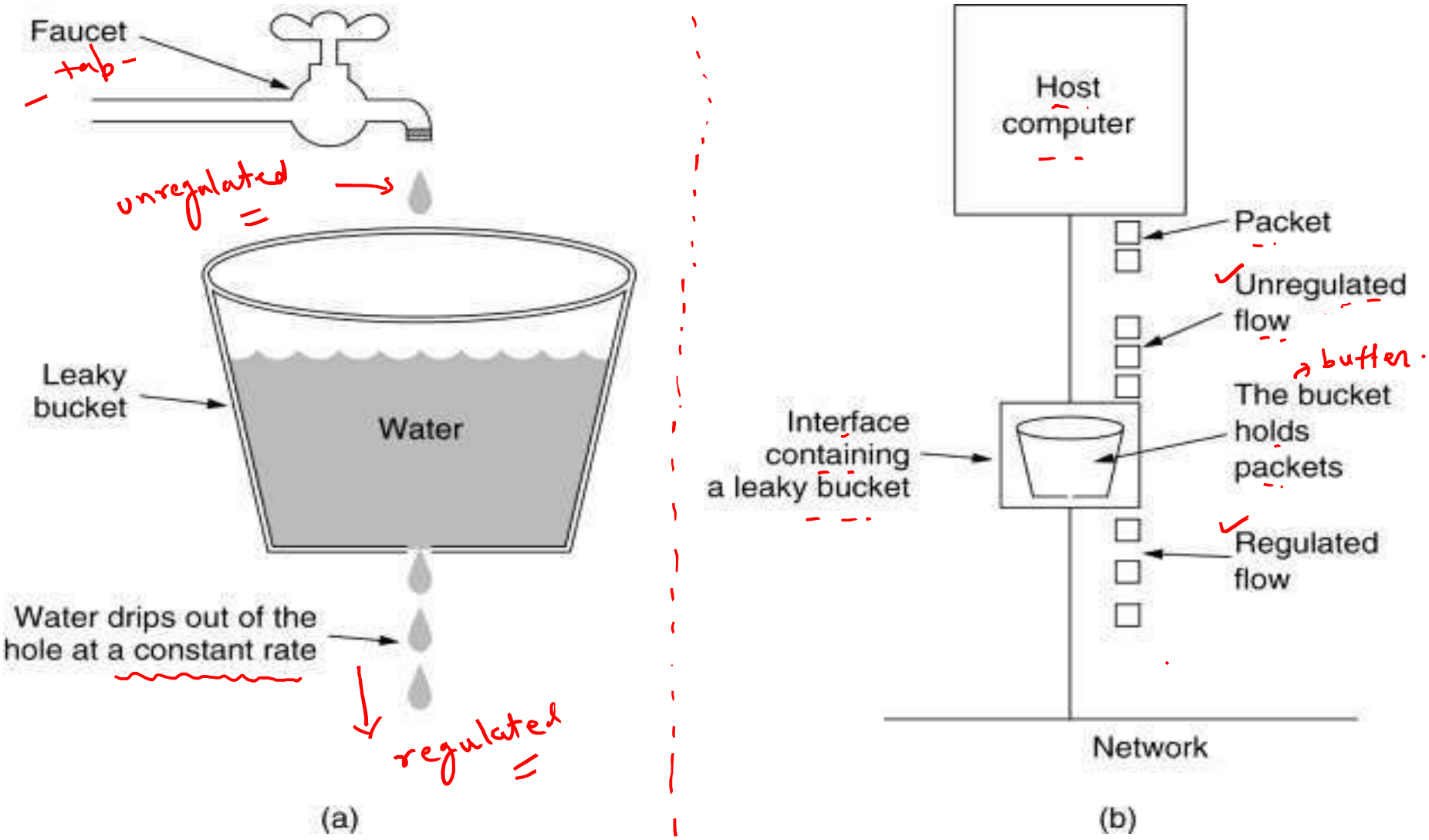
Traffic Shaping

- Another method of congestion Avoidance 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 (asynchronous transfer mode) 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** is used to control the data rate in a network.
- **Each host** is connected to the network by an interface containing a leaky bucket (buffer), that is, a final internal queue of the packets.
- If the bucket (buffer) overflows then packets are discarded.
- In fact it is nothing other than a single-server queuing system with constant service time.

The Leaky Bucket Algorithm

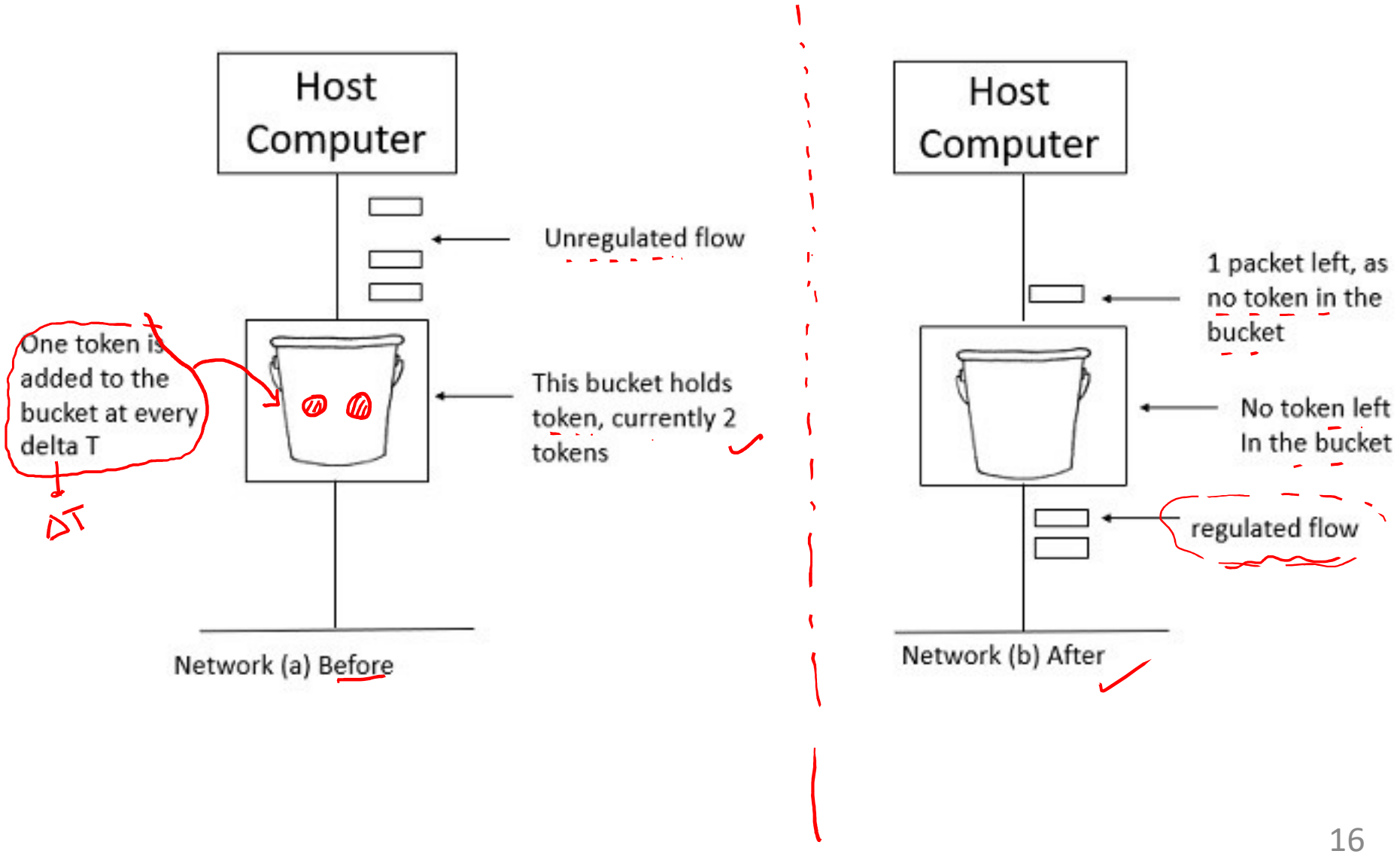


(a)A leaky bucket with water. (b) a leaky bucket with packets.

Token Bucket Algorithm

- To deal with the more traffic, we need a flexible algorithm so that the data is not lost. One such approach is the token bucket algorithm.
- This algorithm step-wise is as follows:
- Step 1: Each host computer is provided with a bucket and the tokens are thrown into that bucket at regular intervals (Δt).
- Step 2: The bucket has a maximum capacity f.
- Step 3: If the packet is ready, then a token is removed from the bucket, and the packet is sent.
- Step 4: Suppose, if there is no token in the bucket, the packet cannot be sent.

Token Bucket Algorithm



To be continued in next lecture.
Thank you.