Announcements

- Reading
  - Today: Chapter 5 (5.4-5.5)
- Project #2
  - Due on Friday Sept 28th (10 AM)

Multicast Routing

- Specify a (relatively) small list of hosts to receive traffic
  - may need to exchange traffic as a group
  - must create/destroy group
- Using spanning trees
  - prune links that are have no members of mulicast group
  - for distance-vector use a variation on reverse path forwarding
    - when a router gets a message it doesn’t need it send a prune message back
    - recursively prunes back un-needed subnets
- core-based trees
  - one tree for group not one per group member
  - hosts send to “core” and it multicasts it out

Congestion

- Too much traffic can destroy performance
  - goal is to permit the network to operate near link capacity
  - can reach a knee in the packets sent vs. delivered curve
- Sources
  - all traffic is destined for a single out link
    - backup in traffic consumes buffers
    - other (cross traffic) will not get through due to lack of buffers
  - slow router CPU
    - can’t service all requests at link speed
      - links still backup
- Often feeds on itself
  - queuing delays can cause packets to timeout
    - introduces more traffic due to re-transmissions

Congestion Control

- Two possible approaches
  - open loop: prevent congestion from every happening
    - tends to be conservative and result in under utilizaion
  - closed loop: detect and correct
    - some congestion will still occur until it is corrected
- Open loop
  - request resources before using them
  - global (or regional) resource allocation
    - responds yes or no to each request for service
- Closed loop
  - monitor network to detect congestion
  - pass information back to location where action can be taken
  - adjust system operation to correct the problem

Responding to Congestion

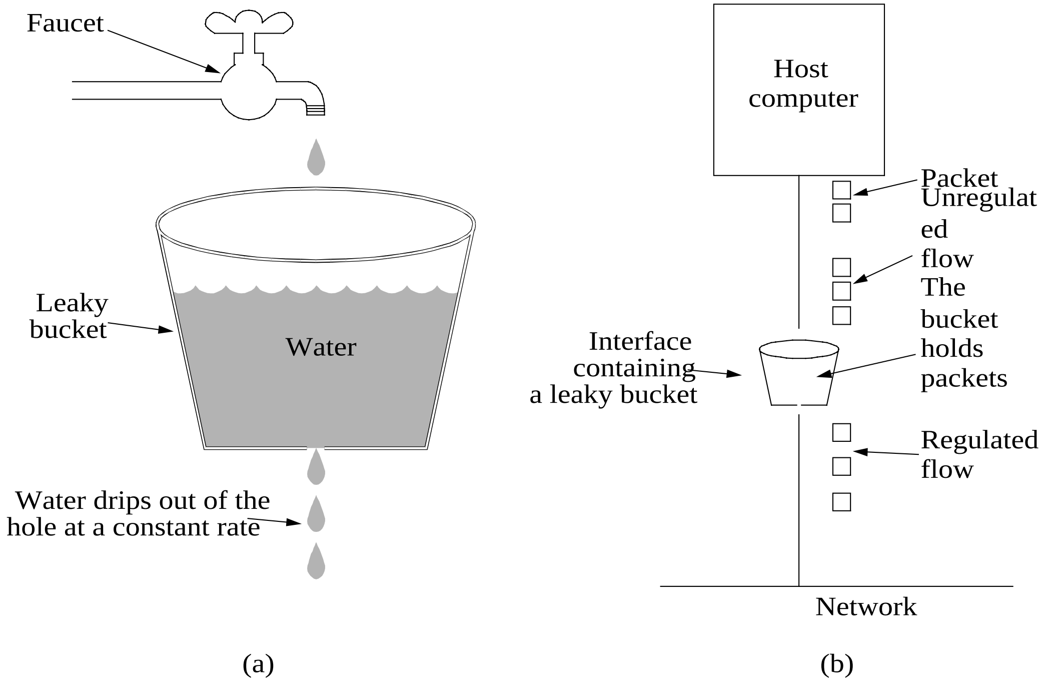
- Add more resources
  - dialup network: start making additional connections
  - SMDS: request additional bandwidth from provider
  - split traffic: use all routes not just optimal
- Decrease load
  - deny service to some users: based on priorities
  - degrade service to some or all users
  - require users to schedule their traffic

Traffic Shaping

- Traffic tends to be bursty
  - great variation between min and max bandwidth used
  - this uncertainty leads to inefficient use of the network
- Flow Specification
  - user proposes a specific probability distribution
    - maximum packet size
    - transmission rate (min, max, or mean)
    - maximum delay
    - maximum delay variation (jitter)
    - quality guarantee (how strong is this agreement)
  - network can
    - agree to request
    - refuse it
    - counter offer

Leaky Bucket

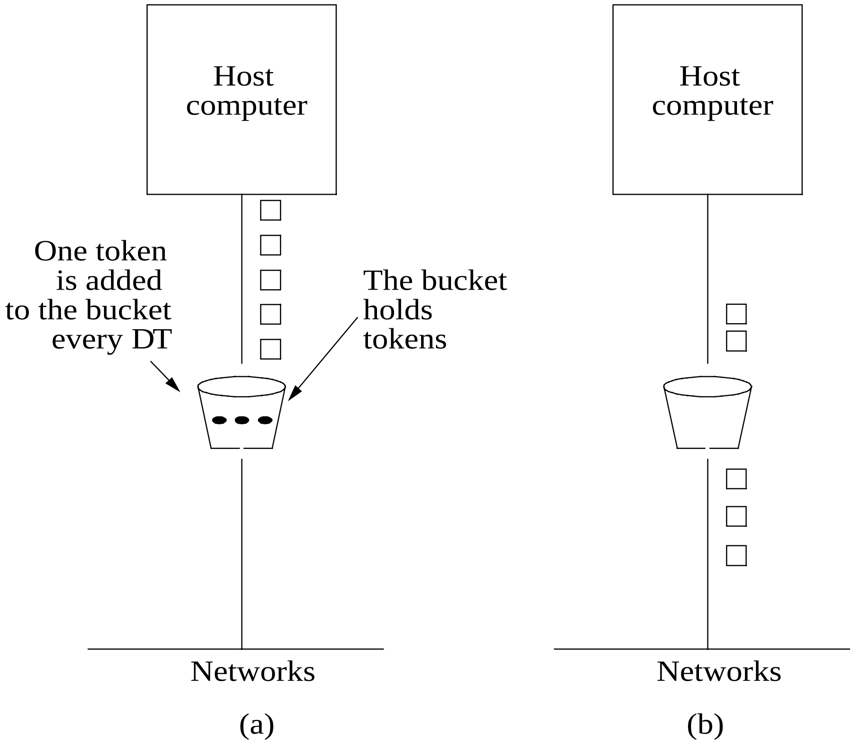
- buffer accepts traffic at link rate
  - buffer has a bounded size (limits burst size that is accepted)
- output is limited to a lower rate
  - traffic is constrained to this rate



From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

Token Bucket

- Bucket hold tokens (generated one every T seconds)
- Can save up to a fixed limit of n tokens
- When traffic arrives, it must a have token to be sent



- Max burst rate

  C - capacity of bucket

  S - burst length in seconds

  M - max output rate

  p - token credit rate

  C + pS = MS

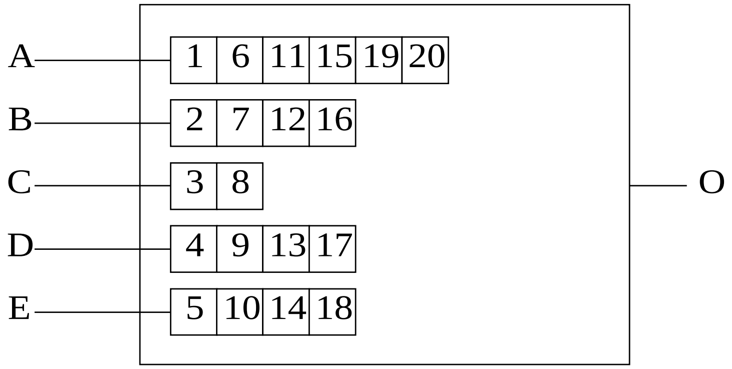
From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

Congestion Control with Virtual Circuits

- Admission control
  - once traffic reaches a threshold, don’t admit more VCs
  - doesn’t correct current problem, but prevents additional congestion
- Alter routes
  - admit new connections
  - route them around “trouble” areas
- Negotiate traffic
  - establish parameters for volume and shape of traffic

Fair Queuing

- Local (per router) congestion control
  - each output link has n queues, one for each sender
    - need to limit max queue size or buffers will be exhausted
  - use round-robin to select next packet to queue
    - can use per-packet or per-byte



(a)

Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

- Weighted Fair Queuing
  - can give different links different priorities
  - give higher priority length multiple slots per round

Choke Packets

- Monitor link utilization
  - keep an estimate (*u*) of average utilization over time
  - $u_{new} = au_{old} + (1 - a)f$
    - f is a 0/1 sampling of link state
    - a is a parameter to control history
  - can also use queue length or buffer utilization
- When utilization is above a threshold
  - for each new packet to be sent over congested link
    - send “choke” packet back to sender
    - tag forwarded data packet to prevent more coke packets
  - when sender receives choke packet
    - must reduce rate to “choked” destination
- Hop-by-hop coke
  - on path back to sender, each router reduces traffic
  - consumes buffer space along path to sender
  - provides faster relief to congested router/link