Queue Management and Congestion Control

CSci551: Computer Networks
SP2006 Thursday Section
John Heidemann

Where are we?

- Problem: traffic grows until it will fill the network
- Solutions:
  - end-hosts: backoff TCP and DECbit
  - routers: drop packets, RED
  - XCP uses both hosts and routers
  - do we need something beyond best effort?

What About Application Needs?

- Congestion is a problem for best-effort, congestion reactive traffic
- *Real-time* or *multimedia* traffic has other needs:
  - bound delay, required bandwidth, control loss rate (or maybe not)
  - might want some kind of Quality of Service (QoS)
- Solutions for RT tfc:
  - end-hosts: adaptivity (ex. playout point)
  - routers: queueing, resource reservation
  - same kind of interactions

Best Effort and Better-Effort Traffic

- Up to now (congestion control/TCP) and next (int-serve/diff-serve) use the *same* basic mechanisms
  - adaptive applications & router queueing
- goal before: e2e adaptivity & fairness
- goal now: providing guarantees (QoS)

Taxonomy

- router-centric vs. host-centric
  - routers: queueing, reservations, adaptive routing
  - hosts: congestion control
  - variant: only at the *edge-routers*
- reservations vs. feedback
  - reservations: hosts or apps ask for resources (admission control: yes/no)
  - feedback: hosts send, routers tell to slow down

Taxonomy (cont.)

- Window-based vs. rate-based
  - really the same thing:
    - W/RTT = rate
  - but two different mechanisms

Service Models

- In practice, fewer than eight choices
  - router/host x reservation/feedback x window/rate
- Best-effort vs. guaranteed service
  - best-effort: can drop anytime
  - guaranteed: will never ever drop (“integrated services”)
  - also better-than-best-effort (“differentiated services”)
- Best-effort networks: hosts, feedback, window
- Others: router, reservations, rate

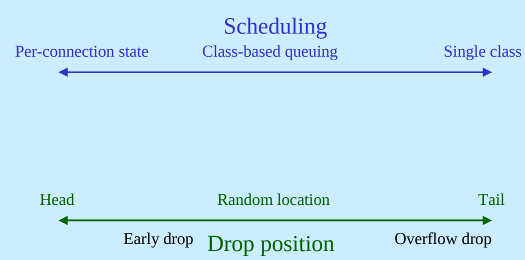
Queuing disciplines

- All router have *some* queueing discipline
- Queuing allocates bandwidth, buffer space, and promptness:
  - bandwidth: which packets get transmitted
  - buffer space: which packets get dropped
  - promptness: when packets get transmitted

Examples

- FIFO/drop tail: don’t split bandwidth, keep whatever arrives (FCFS), send in order of arrival
- FQ: split bandwidth fairly, potentially separate queues per flow (maybe), round-robin like order
- RED: like FIFO expect, drop randomly (vs. keep whatever arrives)
- drop head:

Queueing Design Space



What’s next: Integrated Services

- Integrated services
  - resource reservations (Internet: RSVP)
  - guaranteed or probabilistic bandwidth/delay
- Pros:
  - good match for real-time traffic (ex. VOIP)
  - perfect for VPNs (ISPs can sell “virtual pipes”)
  - make the most use out of your bandwidth
- Cons:
  - too much state for backbone routers
  - difficult policy issues between ASes?
  - ⇒ not widely deployed

Differentiated Services

- Differentiated services
  - assumes some overprovisioning
  - very simple service model
    - *best-effort* and *preferred* (better-than-best-effort)
    - or *in* and *out* (best-effort and less-than-best-effort)
- Pros:
  - easy to implement and fast (no per-flow state)
  - ISPs can charge extra for preferred
- Cons:
  - no guarantees