Queue Management
Jennifer Rexford
COS 461: Computer Networks
Lectures: MW 10:10-10:50am in Architecture N101
http://www.cs.princeton.edu/courses/archive/spr12/cos461/

Monday: Congestion Control
What can the end-points do to collectively to make good use of shared underlying resources?

Today: Queue Management
What can the individual links do to make good use of shared underlying resources?

Packet Queues

Router

Line Cards (Interface Cards, Adaptors)
• Packet handling
  – Packet forwarding
  – Buffer management
  – Link scheduling
  – Packet filtering
  – Rate limiting
  – Packet marking
  – Measurement
Packet Switching and Forwarding

Queue Management Issues
• Scheduling discipline
  – Which packet to send?
  – Some notion of fairness? Priority?
• Drop policy
  – When should you discard a packet?
  – Which packet to discard?
• Goal: balance throughput and delay
  – Huge buffers minimize drops, but add to queuing delay (thus higher RTT, longer slow start, …)

FIFO Scheduling and Drop-Tail
• Access to the bandwidth: first-in first-out queue
  – Packets only differentiated when they arrive

  • Access to the buffer space: drop-tail queuing
    – If the queue is full, drop the incoming packet

  Early Detection of Congestion

Bursty Loss From Drop-Tail Queuing
• TCP depends on packet loss
  – Packet loss is indication of congestion
  – TCP additive increase drives network into loss
• Drop-tail leads to bursty loss
  – Congested link: many packets encounter full queue
  – Synchronization: many connections lose packets at once

Slow Feedback from Drop Tail
• Feedback comes when buffer is completely full
  – … even though the buffer has been filling for a while
• Plus, the filling buffer is increasing RTT
  – … making detection even slower
• Better to give early feedback
  – Get 1-2 connections to slow down before it’s too late!
Random Early Detection (RED)

- Router notices that queue is getting full
  - ... and randomly drops packets to signal congestion
- Packet drop probability
  - Drop probability increases as queue length increases
  - Else, set drop probability $f(\text{avg queue length})$

Properties of RED

- Drops packets before queue is full
  - In the hope of reducing the rates of some flows
- Drops packet in proportion to each flow’s rate
  - High-rate flows selected more often
- Drops are spaced out in time
  - Helps desynchronize the TCP senders
- Tolerant of burstiness in the traffic
  - By basing the decisions on average queue length

Problems With RED

- Hard to get tunable parameters just right
  - How early to start dropping packets?
  - What slope for increase in drop probability?
  - What time scale for averaging queue length?
- RED has mixed adoption in practice
  - If parameters aren’t set right, RED doesn’t help
- Many other variations in research community
  - Names like “Blue” (self-tuning), “FRED” ...

Feedback: From Loss to Notification

- Early dropping of packets
  - Good: gives early feedback
  - Bad: has to drop the packet to give the feedback
- Explicit Congestion Notification (ECN)
  - Router marks the packet with an ECN bit
  - Sending host interprets as a sign of congestion
  - Requires participation of hosts and the routers

First-In First-Out Scheduling

- First-in first-out scheduling
  - Simple, but restrictive
- Example: two kinds of traffic
  - Voice over IP needs low delay
  - E-mail is not that sensitive about delay
- Voice traffic waits behind e-mail

Link Scheduling
Strict Priority

- Multiple levels of priority
  - Always transmit high-priority traffic, when present
- Isolation for the high-priority traffic
  - Almost like it has a dedicated link
  - Except for (small) delay for packet transmission
- But, lower priority traffic may starve 😞

Weighted Fair Scheduling

- Weighted fair scheduling
  - Assign each queue a fraction of the link bandwidth
  - Rotate across queues on a small time scale
  - ![Diagram showing 50% red, 25% blue, 25% green]
- Work-conserving
  - Send extra traffic from one queue if others are idle

Implementation Trade-Offs

- FIFO
  - One queue, trivial scheduler
- Strict priority
  - One queue per priority level, simple scheduler
- Weighted fair scheduling
  - One queue per class, and more complex scheduler

Quality of Service Guarantees

Distinguishing Traffic

- Applications compete for bandwidth
  - VoIP and email sharing a link
  - E-mail traffic can cause congestion and losses
- Principle 1: Packet marking
  - So router can distinguish between classes
  - E.g., Type of Service (ToS) bits in IP header

Preventing Misbehavior

- Applications misbehave
  - VoIP sends packets faster than 1 Mbps
- Principle 2: Policing
  - Protect one traffic class from another
  - By enforcing a rate limit on the traffic
Subdividing Link Resources

- **Principle 3: Link scheduling**
  - Ensure each application gets its share
  - ... while (optionally) using any extra bandwidth
  - E.g., weighted fair scheduling

Reserving Resources, and Saying No

- **Traffic cannot exceed link capacity**
  - Deny access, rather than degrade performance
- **Principle 4: Admission control**
  - Application declares its needs in advance
  - Application denied if insufficient resources available

Quality of Service (QoS)

- **Guaranteed performance**
  - Alternative to best-effort delivery model
- **QoS protocols and mechanisms**
  - Packet classification and marking
  - Traffic shaping
  - Link scheduling
  - Resource reservation and admission control
  - Identifying paths with sufficient resources

Conclusions

- **Link resource allocation**
  - Buffer management
  - Link scheduling
- **Friday precept**
  - Practice exam questions on resource allocation
  - See six questions posted on syllabus page
- **Next week: routing dynamics**
  - Routing protocol convergence
  - Routing to mobile hosts