Queue Management

Mike Freedman

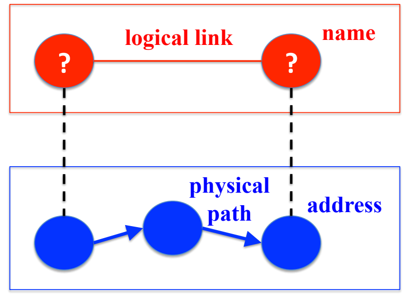
COS 461: Computer Networks

Lectures: MW 10-10:50am in Architecture N101

http://www.cs.princeton.edu/courses/archive/spr13/cos461/

Monday: Congestion Control

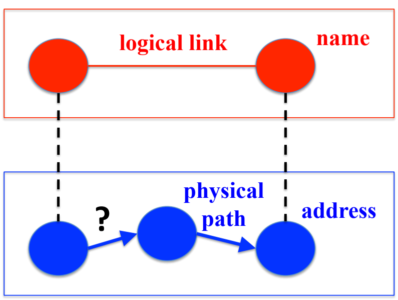
What can the *end-points* do to collectively to make good use of shared underlying resources?



2

Today: Queue Management

What can the individual *links* do to make good use of shared underlying resources?

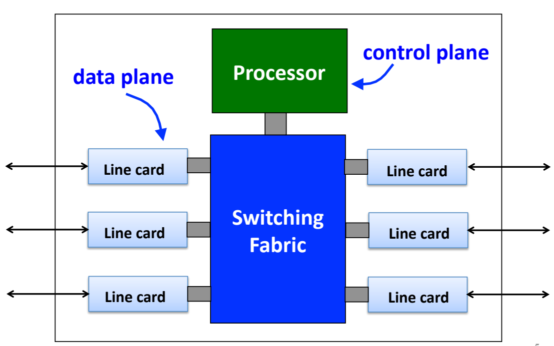


3

Packet Queues

4

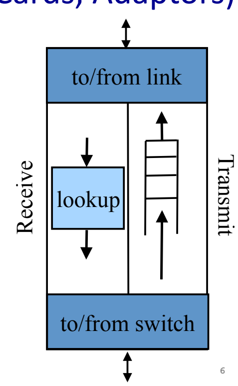
Router



5

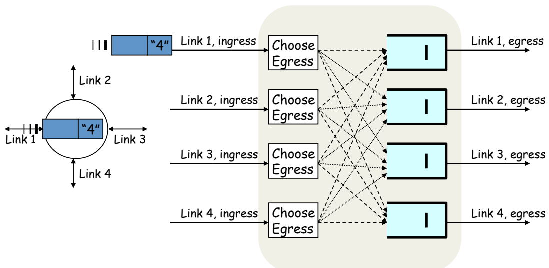
Line Cards (Interface Cards, Adaptors)

- Packet handling
  - Packet forwarding
  - Buffer management
  - Link scheduling
  - Packet filtering
  - Rate limiting
  - Packet marking
  - Measurement



6

Packet Switching and Forwarding

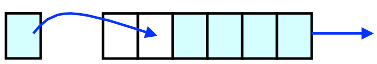


Queue Management Issues

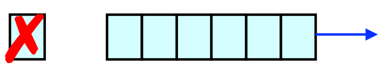
- Scheduling discipline
  - Which packet to send?
  - Some notion of fairness? Priority?
- Drop policy
  - When should you discard a packet?
  - Which packet to discard?
- Goal: balance throughput and delay
  - Huge buffers minimize drops, but add to queuing delay (thus higher RTT, longer slow start,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

FIFO Scheduling and Drop-Tail

- Access to the bandwidth: first-in first-out queue
  - Packets only differentiated when they arrive



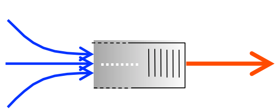
- Access to the buffer space: drop-tail queuing
  - If the queue is full, drop the incoming packet



Early Detection of Congestion

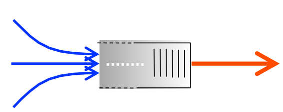
Bursty Loss From Drop-Tail Queuing

- 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



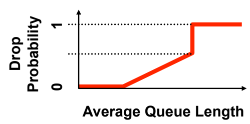
Slow Feedback from Drop Tail

- 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*(avg queue length)



Properties of RED

- Drops packets before queue is full
  - In the hope of reducing the rates of some flows
- Tolerant of burstiness in the traffic
  - By basing the decisions on average queue length
- Which of the following are true?
  - (A) Drops packet in proportion to each flow's rate
  - (B) High-rate flows selected more often
  - (C) Helps desynchronize the TCP senders
  - (D) All of the above

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"…

From Loss to Notification

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
  - Router marks the packet with an ECN bit
  - Sending host interprets as a sign of congestion

Explicit Congestion Notification

- Needs support by router, sender, AND receiver
  - End-hosts check ECN-capable during TCP handshake
- ECN protocol (repurposes 4 header bits)
  1. Sender marks "ECN-capable" when sending
  2. If router sees "ECN-capable" and congested, marks packet as "ECN congestion experienced"
  3. If receiver sees "congestion experienced", marks "ECN echo" flag in responses until congestion ACK'd
  4. If sender sees "ECN echo", reduces *cwnd* and marks "congestion window reduced" flag in next packet

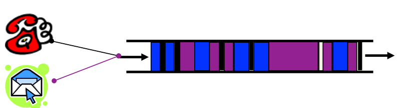
ECN Questions

- Why separate ECN experienced and echo flags?
  - (A) Detect reverse path congestion with "experienced"
  - (B) Congestion could happen in either direction, want sender to react to forward direction
  - (C) Both of the above
- Why "echo" resent and "congestion window reduced" ACK?
  - (A) Congestion in reverse path can lose ECN-echo, still want to respond to congestion in forward path
  - (B) Only should apply backoff once per cwnd
  - (C) Both of the above

Link Scheduling

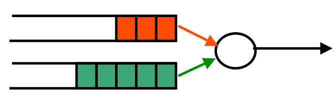
First-In First-Out Scheduling

- 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



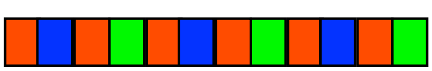
Strict Priority

- 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



50% red, 25% blue, 25% green

Weighted Fair Scheduling

- If non-work conserving (resources can go idle)
  - Each flow gets at *most* its allocated weight
- WFQ is work-conserving
  - Send extra traffic from one queue if others are idle
  - Algorithms account for bytes, not packets
  - Results in (A) higher or (B) lower utilization than non-work conserving?
- Algorithm accounts for bytes, not packets
- WFQ results in max-min fairness
  - Maximize min rate of each flow

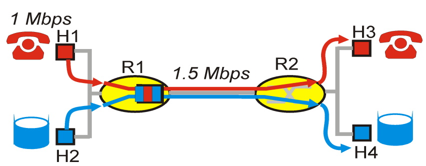
Implementation Trade-Offs

- 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

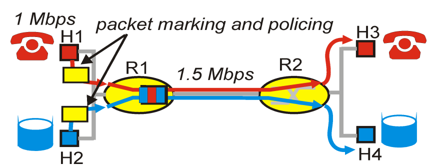
Distinguishing Traffic

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



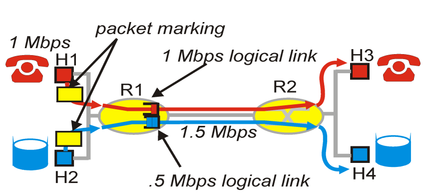
Preventing Misbehavior

- 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



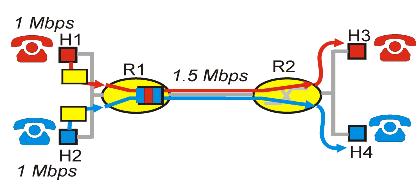
Subdividing Link Resources

- 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

31

Conclusions

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

32