Announcements

• HW1 due on 9/14
• Read papers before the class starting next lecture
• P1 will go out in a few days, start early!
• Start thinking about P2 groups
Resource Sharing in a Network

• Shared resources
  – Buffers (Queue)
  – Links

• Supply > Demand
  – Could under-utilize resources

• Why sometimes demand > supply?
Congestion

• Demand exceeds available resources
• Congestion Collapse
  – Offered load approaches 0
• Reasons
  – Incorrect estimate of resources we have
  – Users not cooperating
  – Trying to send packets too fast
  – Under provisioning
  – Certain loss events followed by retransmission
Congestion Collapse

From [Chiu89]
Managing Bottlenecks

• Possible bottlenecks
  – Receiver
  – Network

• Type of actions
  – Drop packets from the queue
  – Give fair access to the resources
  – Receiver asks the sender to send fewer packets
  – Network asks the sender to send fewer packets
  – Reservations

• Congestion Control: Manage network congestion
Queue Management

• Which packets to queue?
• The order in which packets are served
  – FIFO, etc.
• Which packets to drop?
  – Which and When?
  – E.g., Drop tail
Fair Use of Resources

• Many types of fairness
  – Equal share
  – Max-min
• Jain’s fairness Index

\[
\frac{\left(\sum_{1}^{n} x_i\right)^2}{n \sum_{1}^{n} x_i^2}
\]

• Allocate resources fairly
  – Need accounting system

1/5 for each flow
Calculate JFI

http://en.wikipedia.org/wiki/Fairness_measure
Flow Control

• Receiver bottlenecks
• Ask the sender to slow down
  – End-to-end signaling required
• Stop-and-wait with receiver window
• Do we still need network congestion control?

http://en.wikipedia.org/wiki/Flow_control
Congestion Control

• Prevent in-network congestion
• Enable optimal use of in-network resources

• Components
  – Determine the sending rate
  – Decreasing/increasing the rate in response to events
  – Signaling from the routers to the sender
React to Congestion

• Decrease the rate
• Possibilities
  – Reduce by a fixed amount (additive)
  – Reduced by a fraction (multiplicative)
Properties of Rate Adjustment Algorithms

From [Chiu89]
Fairness and Efficiency

From [Chiu89]
Convergence With AIMD

From [Chiu89]
Oscillation With AIAD

From [Chiu89]
Linear Control Algorithms

\[ W_{i+1} = aW_i + b \]

• Pick weights to match the desired behavior
• AIMD
  – Increase: \( a=\? \) \( b=\? \)
  – Decrease: \( a=\? \) \( b=\? \)
• AIAD
  – Increase: \( a=\? \) \( b=\? \)
  – Decrease: \( a=\? \) \( b=\? \)

http://en.wikipedia.org/wiki/Additive_increase/multiplicative_decrease
AIMD with Window Based Control

• Receiver Window (rcvwnd)
  – Buffer size at the receiver
  – Receiver capacity
• Congestion Window (cwnd)
  – # unacked packets allowed
  – Network capacity
• # pkts sender allowed to send = min(cwnd,rcvwnd)
  – Minimum of network or receiver capacity
Packets in Transit

- $cwnd=5$

- What happens in high bandwidth-delay links?

Clocking the Packets

Sender

pkt

ack

Receiver
Conservation of Packets

• Put a new packet when a packet leaves
• Equilibrium
  – Stable rate, # pkts in transit = full window size
• Reasons for packet conservation to fail
  – Equilibrium not reached
  – Inject a packet before a packet leaves
  – Equilibrium cannot be achieved (unstable)

From [Jacobson88]
Getting to the Equilibrium

• Need to probe the network to find its capacity
• Also, need to know the delays to avoid spurious retransmission
Slow Start

- Upon RX each ack
  - Cwnd = cwnd + 1
  - In Practice: incr = MSS*MSS/cwnd
- Effect is we double cwnd every round trip
- We are still sending min (cwnd,rcvwnd) pkts
- Continue until ssthresh estimate or pkt drop

http://en.wikipedia.org/wiki/Slow-start
Slow Start

Figure 3: Startup behavior of TCP without Slow-start

Figure 4: Startup behavior of TCP with Slow-start

From [Jacobson88]
Congestion Avoidance

• $ssthresh = \frac{cwnd}{2}$
• Set $cwnd=1$, restart slow start
• When, $cwnd \geq ssthres$, additive increase
  – Congestion avoidance

• During congestion avoidance
  – increment $cwnd$ by 1 per RTT
Fast Retransmit

• Upon dupack, do not enter slow start
• Wait for a few dupacks (3), and retransmit just the missing segment
Fast Recovery

• After fast retransmit halve the cwnd
• Use congestion avoidance after fast retransmit, not slow start

• Why?
TCP Retransmission

• Retransmit lost segments
• When to retransmit?

• Under-estimate RTT
  – Spurious retransmission
• Over-estimate RTT
  – Long recovery
TCP Retransmit Timer

• Estimate RTT
• Low pass filter
  – $\text{RTT}(t) = \alpha \times \text{RTT}(t-1) + (1-\alpha) \times \text{newEst}$
• Getting it to work
  – Update how often
  – Which packets to count
  – Take into account variation

http://www.ietf.org/rfc/rfc2988.txt
TCP Connection

1

{SYN, seq1}

{SYN, seq2
 ACK, seq1+1}

2

{ACK, seq2+1}