6.263 Problem Set 6

MIT, Fall 2005

Issued: Thursday, Oct 27
Due: Thursday, Nov. 3

Problem 6.1

(a) Define the round trip time (RTT) and the congestion window (cwnd). Also, define the end-to-end propagation delay. How does the RTT change as the queue build up?

(b) TCP congestion control uses a sliding window algorithm, i.e., TCP controls the sending rate implicitly by controlling the congestion window and by relying on the fact that rate = cwnd/RTT. Louis, who has taken 6.263 last year, argues that if the sending rate of a window-based control protocol is slightly larger than the bottleneck link capacity and the router buffer is large enough, then the sender does not need to reduce the congestion window to match the sending rate to the capacity. This will naturally happen without any reaction from the sender. Do you agree? Explain.

(c) The TCP sender uses 3 dupacks as a sign of a packet drop. This is called *fast retransmission*. Before fast retransmission, TCP used to solely rely on timeouts to discover drops. The use of 3 dupacks as an indication of a dropped packet allows the sender to react quickly to the drop (it retransmits the lost packet and reduces cwnd). First, explain why the sender might receive 3 dupacks even though there was no drop. Second, it is not always possible to detect a drop from 3 dupacks, and TCP still needs use timeouts sometimes. Explain why it is not always possible to detect a drop from 3 dupacks.

(d) We said that the objectives of congestion control are efficiency and fairness. Also, we argued that by doing *additive-increase multiplicative-decrease* (AIMD) TCP approaches over time a fair and efficient bandwidth allocation. If we replace the AIMD in TCP by *multiplicative-increase and multiplicative-decrease* (MIMD) control law, would TCP still approach efficiency over time? Would it approach fairness?

(e) Assume two TCP senders share the same bottleneck link. Given the current design of TCP over time the two senders' average rate will be more or less the same. However, assume that we want sender 1 to achieve twice as much average throughput as sender 2. Change the TCP at sender 1 so that the average throughput of sender 1 is twice as much as sender 2.

(f) Assume a single TCP traverses a bottleneck link of capacity C packets/sec. The 2-way propagation delay (from sender to receiver and back to sender) is D secs. If we want the link to be fully utilized, how big should the buffer be? (Hint: think of the TCP sawtooth.)

Problem 6.2

(a) Explain why in RED max_thresh is not the maximum buffer size.

(b) Why does RED use the average queue rather than the instantaneous queue?

(c) The designers of RED do not provide a proof or an argument that RED+TCP will be a stable control system. But assume that RED converges to an equilibrium in which the aggregate sending rates of the TCP sources sharing the link matches the capacity of the link. How would the equilibrium queue size change as a function of the number of TCPs sharing the link? (Hint: use the TCP throughput equation and assume RTT is constant.)

Problem 6.3 Alissa Hacker has an idea for improving TCP's performance. Alissa has heard that using packet drops as a sign of congestion works badly in very high bandwidth environments. Further, drops waste network resources and do not explicitly tell the sender the severity of congestion. Alissa thinks that TCP should use increased delay as a signal of congestion. In particular, she argues that as long as the peak input traffic at the bottleneck does not exceed its capacity, there will be no queues and the RTT of the sender should stay close to the propagation delay, which is constant. If the input rate at the bottleneck exceeds its capacity, the RTT will grow due to queuing delay.

(a) Alissa designs a scheme in which the sender keeps increasing its congestion window as long as the RTT stays constant and slows down once the RTT starts increasing. The sender tries to figure out the maximum sending rate at which the RTT stays constant. Assume that there is only one source of traffic and that it follows Alissa's protocol. Ben argues that Alissa's scheme may not work if the sender is bursty and that the sender needs to pace its traffic. Explain why Ben is right.

Alissa agrees with Ben and makes the sender pace its traffic to ensure that the instantaneous send rate is close to the average over an RTT. Alissa replaces the AIMD update rules in TCP with a new `cwnd` update rule. She doesn't change slow start, and you should ignore slow start for this problem. The sender keeps a measure of the minimum RTT seen so far, called BaseRTT. This BaseRTT is used as an estimate of the propagation delay. The congestion window is computed in packets while RTTs are measured in seconds. Assume that round trip times are always less than 100ms.

Every T=100ms, the sender computes the average RTT seen in the last T, and:

$$cwnd = cwnd\left(\frac{BaseRTT}{RTT}\right) + \alpha$$

where $\alpha > 0$ is a constant. Alissa's protocol does not reduce cwnd in response to lost packets.

(b) Why it is important to have a positive α? If α were zero, what would go wrong?

(c) If Alissa's TCP shared the bottleneck link with a standard AIMD TCP that uses dropped packets as the congestion signal, which one do you expect to get more throughput? Why? Assume a shared drop tail queue and a very large queue limit.

(d) Assume the network contains only Alissa's TCPs. Assume also that everyone has the same propagation delay. Is Alissa's TCP fair? Explain. Assume that no packets are dropped.

(e) If the network contained only UDP and Alissa's TCPs, would Fair Queuing be useful? Explain.

(f) Alissa says that, theoretically, the congestion window in her protocol is less likely to show oscillations as wide as TCP's sawtooth. Explain why this is true.

(g) Alissa claims that since her TCP does not use drops as a sign of congestion, it will work well in both low-speed and very high-speed networks in which a single flow could send at gigabits/second.

    (1) Explain why current TCP has difficulties in very high bandwidth networks.

    (2) Ben argues that, though Alissa's protocol reacts to delay rather than drops, its cwnd update rules are not that different from AIMD. The increase constant is α per 100ms rather than one packet per RTT. Explain this statement.

    (3) Explain to Alissa why a single value of α won't work well in both low- and very high-bandwidth networks.