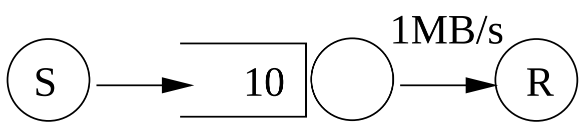
CSE 461 Homework 4
Due WEDNESDAY November 24, 1999

1. How many round trip times would it take to transfer 7 packets of data using TCP during slow start:
   (a) without delayed acks?
   (b) with delayed acks? The delayed ack rule is as follows: on receipt of a packet, start a 200ms timer. When a second packet arrives, or the timer expires, send the acknowledgement. That is, an acknowledgement is sent after every other packet, unless there is only one packet, in which case the acknowledgement is (in theory) delayed.[1]

   Ignore setup/teardown cost. Pretend that the round trip time of the connection is around 2 seconds if you need a number to toss around in your head. That is, round trip time dominates. Assume there is no loss.

2. Before a packet is considered lost by timeout, cwnd=8 and ssthresh=16. When the packet is retransmitted, what are the new values of:
   (a) cwnd
   (b) ssthresh



3. In the network above, the bottleneck bandwidth is 1 mega*byte* per second. The one way latency is 5 ms. The queue length at the bottleneck link is 10 packets. The MSS of the connection between S and R is 1500 bytes.
   (a) Using TCP Reno, what is the maximum value of cwnd (in packets) you would expect to see in the congestion avoidance phase? That is, ignore the spurious cwnd overshoot in the initial slow start phase, and consider only how many packets and acknowledgements will successfully fit in the network at a time.
   (b) Using TCP Vegas, with $\alpha = 1$ buffer ($\alpha = 150$ kB/s) and $\beta = 3$ buffers ($\beta = 450$ kB/s), what is the approximate value (within +/-1) of cwnd you would expect to see most often? That is, what value should cwnd converge to? (Hint: First guess the value of Diff between α and β. Solve for the expected rate. Then solve for w in packets, and round to an integer.)

[1] in practice, delayed acks are disabled during slow start. please ignore this bit of trivia when working out this problem

4. The following packets were output by tcpdump, when run on the machine “me” while serving a 9287 byte web page. The tcpdump tool normalizes the sequence numbers in the conversations so that they are small integers. That is, the machine “them” chooses 1629852695 as it’s ISN, but tcpdump subtracts this number from all future sequence numbers seen from “them.”

   Match packets sent by “me” with their acknowledgement to generate 6, often overlapping, samples of the round trip time of the connection. Remember that a TCP acknowledgement contains the sequence number of the next byte expected, so acknowledges everything up to, but not including, the sequence number in the acknowledgement.

   What are the 6 round trip time samples?

```
18:16:35.149595 them > me: S 1629852695:1629852695(0) win 32120  (DF)
18:16:35.149648 me > them: S 2210326433:2210326433(0) ack 1629852696 win 16060  (DF)
18:16:35.242646 them > me: . ack 1 win 32120  (DF)
18:16:35.243773 them > me: P 1:726(725) ack 1 win 32120  (DF)
18:16:35.243809 me > them: . ack 726 win 15335  (DF)
18:16:35.244689 me > them: P 1:1449(1448) ack 726 win 16060  (DF)
18:16:35.244702 me > them: P 1449:2897(1448) ack 726 win 16060  (DF)
18:16:35.332742 them > me: . ack 1449 win 31856  (DF)
18:16:35.332780 me > them: P 2897:4345(1448) ack 726 win 16060  (DF)
18:16:35.332791 me > them: P 4345:5793(1448) ack 726 win 16060  (DF)
18:16:35.334370 them > me: . ack 2897 win 30408  (DF)
18:16:35.334401 me > them: P 5793:7241(1448) ack 726 win 16060  (DF)
18:16:35.334412 me > them: P 7241:8689(1448) ack 726 win 16060  (DF)
18:16:35.334423 me > them: FP 8689:9536(847) ack 726 win 16060  (DF)
18:16:35.425453 them > me: . ack 5793 win 31856  (DF)
18:16:35.425456 them > me: . ack 8689 win 30408  (DF)
18:16:35.425458 them > me: . ack 9537 win 30408  (DF)
18:16:35.440199 them > me: F 726:726(0) ack 9537 win 31856  (DF)
18:16:35.440230 me > them: . ack 727 win 16060  (DF)
```

5. In the tcpdump trace above, what TCP state (from the TCP state machine handout) is “me” in at:

   (a) 18:16:35.1

   (b) 18:16:35.425456

   (c) 18:16:35.5