Homework 5 for CSE/EE 461 (Winter 2001)

Due: Friday, Feb 23, 2001, at the beginning of class. (Out: Fri, Feb 16, 2001.)

Note: be sure to show how you derived answers so that you are eligible for partial credit.

1. Packet Trace.

The following packet trace was output by tcpdump, which is a common program for monitoring network activity. It shows the exchange of packets seen by the machine "me" while serving a 9287 byte web page. The output is fairly terse. The tcpdump tool normalizes the sequence numbers in the conversations so that they are small integers. That is, the machine "them" chooses 1629852695 as its Initial Sequence Number, but tcpdump subtracts this number from all future sequence numbers seen from "them." Other notation describes (in rough order) timestamps, whether the packet had the SYN (S) or PUSH (P) flags set, sequence number ranges and packet lengths (e.g., 1:726 (725)), whether the packet is acknowledging data (e.g., ack 726), the advertised flow control window (e.g., win 16060), and that the packet was marked to "don't fragment (DF). The first thing you should do for this question is to read the tcpdump man page (documentation), which is linked from the course web page, to make sure you understand what the trace output means.

```
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)
```

a) Draw a packet time sequence diagram (of the kind shown in lecture and Peterson with time moving down the page) that shows all packets of the transfer. Your diagram should be approximately to scale. For each packet, label it with the type (SYN, ACK) and sequence number range.

2. RTT Estimators.

a) For the trace, calculate the six RTT samples. (Note that TCP actually only takes one sample per round trip time, but you are gathering all available samples in order to work with a smaller trace!)

b) Using the samples, calculate the estimated timeout given by the exponentially-weighted moving average based algorithm. Use a = 0.8, a multiplier of 2 between the estimated RTT and the timeout, and an initial timeout of 500ms. Present your answer as a graph of timeout versus time. Your graph should be roughly to scale, and should include the RTT samples too.

c) Repeat part b) using the Jacobon/Karels algorithm instead. Use d = 1/8, m = 1 and f = 4. Present your answer overlaid on the same graph as part b).

3. Flow Control.

a) For the trace, calculate the byte ranges versus time held in the sender buffer (sent but unacknowledged) as seen from the server. For convenience, you may refer to times as T1, T2, T3, and so forth, meaning the first, second and third timestamp value, and so forth.

b) Similarly, calculate the byte ranges versus time held in the receiver buffer (received but not removed from the flow control buffer by the application) as seen from the server.

4. Connection State.

a) Again with the trace, use the TCP state machine to track the state of the server connection versus time. For each state change, give the time, ending state and the transition (e.g., received SYN and send SYNACK) that caused the change. For convenience, you may refer to times as T1, T2, T3, and so forth, meaning the first, second and third timestamp value, and so forth.

5. Connections. Peterson 5.17

6. Flow Control. Peterson 5.19

7. Connections. Peterson 5.21

□□□