Homework 3 for CSE/EE 461 (Winter 2003; Wetherall)

Due: Mon, Mar. 3, at the beginning of class. (Out: Wed, Feb 19.)

1. Packet Trace. The following packet trace was output by tcpdump, 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, and explained by the tcpdump man page, which is linked from the course web page for convenience.

```
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.

b) Calculate the six RTT samples and use them to compute two estimators for the timeout versus time. The first is an exponentially-weighted moving average based algorithm. Use alpha = 0.8, a multiplier of 2 between the estimated RTT and the timeout, and an initial estimated RTT of 500ms. The second estimator is the Jacobson/Karels algorithm. Use delta = 1/8, mu = 1, phi = 4, the same initial estimated RTT and an initial deviation of zero. Present your answers as a graph, including the RTT samples.

c) Calculate the servers' view of flow control. Give the byte ranges versus time held in the sender buffer (sent but unacknowledged) as seen from the server. Give 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.

d) Follow 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.

2. Packet Pair. TCP uses losses and trial-and-error to estimate the available bandwidth between a sender and receiver, but we can also use packet timing information to estimate the bottleneck bandwidth (the capacity of the smallest link along a network path).

a) Consider a pair of data packets sent along the path "back-to-back", immediately following one another with no timing gap between them, and are immediately acknowledged by the receiver. Give an expression for the gap between the acknowledgement packets that return to the sender. You may ignore other traffic in the network.

b) We can use the above expression and "packet pairs" as the basis for a procedure for estimating the bandwidth of a path. However, TCP doesn't use this estimator to gauge its send rate. Why not? That is, what factors make this estimator unsuitable for TCP?

c) The above notwithstanding, suggest how TCP could make use of packet-pair information.

3. Sliding Windows. Consider a standard sliding window protocol. The size of the sliding window affects the transfer rate that transport protocols such as TCP can achieve.

a) Give an expression for the throughput as a function of the size of the sliding window. What would this throughput be in practice if stop-and-wait were used on a cross-country connection (100ms round trip time)?

b) TCP's maximum receive buffer size for flow control is 64K without the use of extensions. (Defaults are typically 16 or 32K too.) At what throughput will flow control become the limiting factor for cross-country connections? What sliding window size would be needed to support a 1Gbps cross-country connection?

c) How does TCP slow-start change your answer for the expression in part a)?

4. TCP Trace Peterson 6.25

—END—