Homework 2 for CSE/EE 461 (Winter 2004; Davis)
Due: Wednesday, February 11, 2004, at the beginning of class.

1. Flood and Learn. Suppose we modify the solution for Fishnet Assignment 1 so that when forwarding a packet, each node sends it to the last neighbor that forwarded a packet from that source address, and broadcasts it to all neighbors if the node has never seen a packet from that source address. (This is like learning bridges.) Other rules, such as discarding packets we’ve already seen and decrementing the TTL at each hop are kept the same.

   a) Is it possible packets would loop? If so, give an example. If not, explain why not.

   b) What happens when a node moves? Give a simple fix that allows the network to deliver packets to nodes that have moved.

2. TCP 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.

   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.

3. Spanning Tree Non-participation. Peterson 3.21

4. Distance Vector Loops. Peterson 4.23

5. Link State Routing Convergence. This question explores how quickly link-state
routing converges to stable routes after a failure. Assume a simplified network in which
all links have equal propagation delay and all routers take an equal amount of time to
process messages and to calculate shortest paths. Consider the case of a single link
failure that does not partition the network.

a) Describe the state of network connectivity immediately after the failure, before
the nodes have dealt with it.

b) How long will it take from the instant of actual failure until all nodes have
connected, loop-free routes to all destinations if we are using arbitrary link costs?
What network factors does your answer depend on? Describe all the components
of this delay.

c) How long will it take from the instant of actual failure until all nodes have
connected, loop-free routes to all destinations if we are using shortest-hops as our
cost metric? What network factors does your answer depend on? Describe all the
components of this delay.

d) How would your answers above differ if we were using distance vector rather
than link-state?