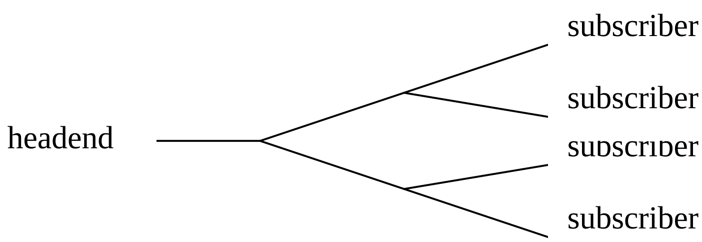
Homework 2 for CSE/EE 461 (Winter 2003; Wetherall)

Due: Fri, Feb 14, at the beginning of class. (Out: Mon, Feb 3.)

1. **Cable Modems.** Cable modems send data over the cable plant, which is modeled as a tree. Subscribers sit at the leaves of the tree and the root of the tree is the headend, which connects the cable plant the rest of the Internet. All communications between subscribers are sent via the headend. Transmissions from the headend are broadcast and reach all subscribers. Transmissions from each subscriber are not broadcast to other subscribers, but travel only to the headend along a path that is shared by different subscribers according to the structure of the tree. Your job is to design a feasible MAC protocol. It must use statistical multiplexing and make some attempt to improve efficiency by avoiding collisions. There are many possible solutions to this question, but the more efficient the MAC protocol, the better.

   a) How does your protocol send a message from the headend to a subscriber?
   b) How does your protocol send a message from a subscriber to the headend?



2. **Forwarding Tables.** This question explores the scalability of forwarding tables for the Internet under different addressing schemes.

   a) Assume that flat addressing similar to Ethernet is used, as in bridging, where each node gets a unique, unstructured address of 32 bits (the same size as IPv4). How large will the backbone forwarding tables be?

   b) Assume that hierarchical addressing is used together with routing, with the same class A, B and C allocations as IPv4. How large will the backbone forwarding tables be now? Your answer should compare this with the part above.

   c) Assume that route aggregation (CIDR in terms of IP prefixes) is used as part of the routing protocol instead of class A, B, and C allocations. How large will the backbone forwarding tables be now and on what factors does this depend? Your answer should compare this with the part above.

3. **Link-State Routing Convergence.** This question explores how quickly link-state routing converge to stable routes after a failure. Assume a simplified network in which all links have equal propagation delay and all routers process messages equally fast. 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?

4. **BGP Path Vector Convergence.** This question explores how quickly a path vector protocol converges after a failure. Assume a simplified network in which N ISPs are meshed to each other only at one Internet Exchange point, all links connecting the ISPs have equal propagation delay, and all routers process messages equally fast. Assume BGP is selecting the shortest route, and when there are equal length routes it is breaking ties by choosing the lowest next hop address. Consider the case of a single link failure elsewhere in the network that causes ISP 1 to lose reachability to an address P.

   a) Before the failure, what route is an ISP X using to reach P and what other routes is it hearing from its neighboring ISPs?

   b) After the failure and ISP 1 sends a withdraw message for the path 1 to reach P, what route will an ISP X be using to reach P?

   c) How long will it take until all ISPs realize that they have no valid route to P?

   d) How would you change the protocol to fix this problem to provide more rapid convergence?

5. **Distance Vector Loops.** Peterson 4.17

—END—