CS 336 Lecture 20– 4/8/98

Announcements

- Handout describing simulation project is now online. (If you did not receive a copy of the Feldmeier paper before break, see me.)
- A "lite" written assignment (Exercise 6 is also now available (due Friday).
- Give me your "team rosters" on Friday.

The IEEE 803.1 Bridge Spanning Tree Algorithm

1. The details of bridge election are fairly simple.
   - Each bridge has a unique "priority" assigned to it actually a non-unique priority with a unique ID appended to it) and each port has a "cost". These can be set by the network administrator or left at default values supplied by the factory.
   - When the bridges recognize the need for an election, they all start sending packet announcing the candidate they endorse including:
     - The candidate's priority (Initially this will be their own priority.
     - Their ID number.
     - The cost of reaching the candidate from the sending hub.
   - As bridges receive such packets, they look for the candidate with the lowest priority/ID pair and given equal priorities they look for the lowest path cost.

     Each bridge maintains
     - The priority/ID of the best candidate for root it has seen (i.e. the smallest priority/ID seen).
     - The cost of and port leading to the best known route to the best candidate for root.
     - For each port attached to the bridge, the contents of the campaign packet received on that port advertising the least cost to reach the root.

     This information is collected as follows:
     - If a packet arrives with a priority/ID greater than the bridges current "candidate, it ignores the packet.
     - If a packet arrives with a smaller priority/ID than the bridges current candidate
       * Replace the bridge's candidate for root with the priority/ID contained in the packet.
       * Set the bridge's "least cost to reach the root" value to infinity.
       * For each port, set the best advertised "reach the root" cost to infinity.
     - Compare the cost advertised in the current packet to the best "reach the root" cost for the port on which it arrived. If its cost is smaller, store it as the new best "reach the root" advertisement.
     - Add the cost associated with the port on which the packet arrived to the cost advertised in the packet. If this cost is smaller than the cost of the best previously known path the the current root candidate, save the cost and port number as the new best route to the root info.
   - When the process is over, every bridge knows the ID of the winner (i.e. the new root bridge) and the cost of and the first step in the least cost path to the root.
2. Once the root is selected, the remainder of the loop-cutting process occurs locally within each sub-LAN.
   - For each of its ports, each bridge compares its cost to reach the root to the smallest cost that had been advertised on the LAN attached to that port.

     If some other bridge advertised a smaller cost, then the original bridge will turn off forwarding of packets received on that port.

Routing in Wide Area Networks

1. For networks that are very large, either in terms of the number of hosts or the distances between hosts, the "shared medium" approaches used in LANs just don't scale.
   - No matter how efficient the protocol for sharing the medium may be, having all message go to all stations involves too much unnecessary transmission of information when the number of stations is very large.
   - We have seen that the efficiency of most medium sharing protocols decreases as the physical size of the network increases.
2. In such networks, it is better to depend on intermediate nodes to forward packets selectively along paths leading to their destinations. This is similar to the work performed by bridges on a LAN except:
   - We are no longer concerned that the process be transparent to sending stations or to network administrators, so...

- We would rather use multiple (i.e. cyclic) paths in the network to maximize performance and improve reliability rather than depend on a spanning tree.

3. To make all this work, when machine A wants to talk to machine B, somehow the network has to find a path from A to B.

   There are several major characteristics of the process of finding such paths that vary in significant ways from network to network:

   Dynamic Nature of Routing The book talks of the choice between establishing static routes when a network is created or using algorithms that dynamically vary routes as components fail, become busy or even get repaired. In fact, I think it is best to say that static routing is something of a strawman whose main role should be to point out the need for dynamic routing. In a large network, the ability to adapt to changes in network load and component failure makes routing protocols that adjust to changes in the state of network links essential.

   Distributed Routing In some early networks a central system computed routes for the whole network. Such a system is both a threat to the reliability of the network and a bottleneck at high load. The alternative is to somehow distribute the task of computing routes among all the nodes of the network. While this is preferable to the alternative of a centralized system, we will see that much of the complexity of the routing problem is due to the use of distributed algorithms.