Distance Vector Routing

Jeffrey C. Slocum

In this project, the goal was to simulate the routing technique distance vector routing. In this, each node sends periodic DV packets, in this case this was considered ‘rounds,’ to each of its neighbors with all the information the sending node has about the topology of the entire network. Once the packets were received, each node would update its routing table in accordance with the new information. The node will first cycle through its known-nodes and add the node to its table if the new node was not previously there. Once done checking for new node additions, the node with go through its table, checking for faster ways to navigate the network. Once complete, the program will output the statistics of the converged network to the terminal, showing total nodes, total rounds, packets created, last node updated, and the path the data followed from node 0 to either node 3 / 7 / 23.

The data structures I used to create this network simulation were two structs: node, and packet. With these I used a linked list style approach to keep track of all nodes, packets received, and packets to be sent. These nodes contained 4 vectors: destination, cost, nextHop, and neighbors. These were used to keep track of destinations, routes, costs, and where to go next. The node structs also contained 3 pointers: link, received, and send. Link was used to link all the nodes together, received was a packet struct used to contain all packets received from the nodes neighbors, and send was used to create and store the packets a node was going to send to its neighbors.

I ran all three topology files, to generate statistics of each of the networks, storing the output in the ./completeTables folder in the project1 root folder. Each network produced different results, except for which node converged last. In which for all three topology files, it was node 0. The rounds each network made were topology1: 1, topology2: 3, topology3: 5. With the increased number of nodes in each network, the difference in DV packets sent is much greater, topology1: 28, topology2: 160, topology3: 720.