Devanshu K., Jonathan S.

March 9, 2020

Project 2 Discussion Questions

1. What are the pros and cons of using distance vector routing compared to link state routing?

An advantage that the link state routing (OLSR) has over distance vector routing (RIP) is that OLSR only needs to know its neighbors, while RIP needs to know every node and link in the whole network. Therefore, RIP introduces large CPU loads as the size of the routing table increases. However, since RIP has information about every node and its links, it can confidently give you the most accurate and efficient path from one node to another. In OLSR, each node must trust its neighbors that it has calculated its routes correctly, so inaccuracies can occur.

1. Does your routing algorithm produce symmetric routes (that follow the same path from X to Y in reverse when going from Y to X)? Why or why not?

RIP does not produce symmetric routes. A route is selected based on the costs inputted in the routing table. For instance, in our code the selected route is the first path that has a lower or equal cost.

1. What if the node advertised itself as having a route to some nodes, but never forwards packets to those nodes? Is there anything you can do in your implementation to deal with this case?

The scenario described above is on par with the flooding algorithm. One way to change the implementation would be to make sure to flood all the neighbors if the target node is farther than one hop.

1. What happens if a distance vector packet is lost or corrupted?

Obviously, the target node will not receive the packet. However, the target node would as for the packet again on a higher layer than distance vectoring with timeout just in case of excessive requests.

1. What would happen if a node alternated between advertising and withdrawing a route to a node every few milliseconds? How might you modify your implementation to deal with this case?

The scenario described above would likely cause a packet to be sent/routed to a target node that has no way of being reached. A way we can modify our implementation for this case would be to add an event that triggers whenever the availability of a node is changed. Whenever the availability of the node changes, the neighbor discovery takes care of the rest of the issues.