Jibram Jimenez-Loza

Alberto Cerpa

CSE 160 – Computer Networks

4 March, 2019

Project 2 – Link State Forwarding Discussion Qs

1. There's no problem with using shortest path algorithms on a graph with single-direction connections. However, it might be difficult to find if the destination node is a neighbor if it can't reach the source node with a reasonable ping reply. A B might exist and A should know that B is a neighbor, but B has no access to A because it is a leaf node, so it won't be able to reply to A saying it is a neighbor.
2. No, because each computation is asynchronous and therefore multiple same-cost paths may be calculated simply due to the chaotic nature of constructing a single path with multiple entries. This could be remedied by referencing the lowest node ID in order to break ties, thus possible creating symmetric routes, but that was not in my design choice.
3. If a node advertised itself as having neighbors but never forwarded packets, I'd assume the path was broken and simple recalculate a path while avoiding that link.
4. It's possible this happens, but after a number of failed attempts to forward a packet using lost/corrupted information, it would be overwritten by new, (hopefully) correct information.
5. I would have to essentially cut off the node as it is "faulty" in a sense that it is causing heavier load to the network than it's usefulness as a node. If a node has a high number of "dropouts" per unit time, it is useful to cut this node out of the network.

Project 2 – Link State Forwarding Design Choices

I've gone ahead and used arrays to hold local neighbors and propagate them through the network. I send these arrays every 3 seconds and run dijkstra's algorithm every 10 seconds to create a forwarding table.