Christian Doud
Phien Nguyen

For starters we wanted to do the link state flooding which was similar to regular flooding. This time we would generate the neighbor table using neighbor discovery which the table was sending to linked state flooding to use to send out the packets to the other nodes. We would search where the neighbors is not infinite and sent it out. Then those node would send it to their neighbor and checks that the TTL is not zero. It would also check if the source is also not itself to prevent forwarding its own flood package. Till the TTL reaches zero, each node would keep on flooding making sure it doesn't send to it self using the sequence number. Each node would store the unseen sequence number every time it forwards so it doesn’t send the same packet form a different node with a different TTL. We would also use a protocol floodlinked to make sure we are reading only flooded packets that are sent within the packets

To do dijkstra, we made sure we would always the neighbor list first. Then we would update the table after we have receive packets from neighbor discovery broadcasting their neighbor table to all neighbor nodes. Each node would then calculate the paths given by the neighbor table to find if they can reach a new node given the new table. Then it would prompt that the table has changed and request the node to broadcast the new table. Each time it would broadcast, its waiting for the neighbor to send their table back and update their table as well. Each iteration till each node is unable to change and have filled their neighbor routing table and either slow down the broadcasting of their routing table or stop because there is a flag set telling the node not to broadcast as there is no change.

The dijkstra algorithm takes a while to run as each nodes need to keep broadcasting to each other that their table is changing therefore trying to update their neighbor and the surrounding nodes. Overall it should end after tables are filled and there are no changes to the routing table.