

## **Project 1: Flooding And Neighbor Discovery Design**

Our implementation of flooding takes place in the Node.nc file. Our flooding implementation was based off of two possibilities. One is what the node is to do if it receives a packet that was not meant for it, and second, is receiving a packet that was intended for that node. Our implementation of the Received event begins with a TTL check, to make sure the packets time to live has not expired, and will not propagate indefinitely. A second check we do is check if the package has been seen by that node already. This is achieved through a separate helper function isKnown() that simply checks the packet list for matching source, destination and sequence id. If either of these two conditions were met the packet would be dropped. Next what we decided to do was to check the protocol of the message. A correct destination would simply stop sending the packet, while a ping with no reply would insert the packet into the seen list, make a new packet with increased seq id and then resend the packet. If the packet does not belong to the current node the packet will be rebroadcasted.

Our neighbor discovery implementation takes on a similar structure. First each node would have a struct that contains the node id and hops or number of pings to it. A timer was wired to NodeC.nc that would fire and call a helper function neighborList() that updates a node's neighbor list. It does this by first looping through our list of neighbors and increases its number of pings since last heard. A second loop goes through and drops any neighbor that has its hops greater than 5, and is then added to our dropped neighbor list. The packet is then repackaged and sent out the AM\_BROADCAST\_ADDR channel. Back in our received function the broadcast address is compared to the message destination, if a match is found it will then continue onto a switch statement that will match its protocol. This will determine which node is a neighbor and which is not by being able to respond correctly to the packet or protocol type.