**Design Process:**

Initially I added the necessary additions provided during the lab section for the fundamental neighbor discovery functions. My partner worked on the main neighbor discovery process, however this process consisted of establishing in neighbordiscovery.nc the fundamental interface that is used for the whole project. The logic in neighbordiscoveryP.nc that boots with a discovery function that checks a node has a neighbor in order to avoid duplication of the packets. The packets would be sent through the simplesend module, this happens through a queue and then a delayed send at random time increments from 0-300ms. In order for the packets to be sent out and received properly the ND (neighbordiscovery) has a function that periodically broadcasts packets. All the neighbors receive a packet and then this enables the flooding process to work alongside it. The flooding has a seenpackets that marks the src and seq numbers that it checks through to ensure that this packet hasn’t already been received. If the packet is “new” it continues to get rebroadcasted. If the packet gets rebroadcasted to the same node from a different neighbor then the packet is dropped. The flooding logic is in floodingP.nc, initially I made the mistake of just implementing straight into node.c which defeats the purpose of needing wiring to connect all the modules together. The flooding works as follows, it is stored in the seenpackets cache with the (src, seq) numbers. The checks that the packets go through are first the TTL (time to live) if TTL = 0 then the packet is immediately dropped. Then the second check is the one I mentioned earlier where it checks for duplicates with the src and seq numbers. If the TTL is not 0 then the packets hop to each node each hop decreases the (TTL -1). The two checks are needed because the program needs to not infinitely loop with these packets and the packets need to expire and drop so new ones can be sent and received.