

## Project 1 Design Document

### Process

Initially, I basically had no clue how any of the components in the entire project worked together. I eventually figured out that the `Boot.booted` event in `node.nc` was the event called at the start of each node's life. Then, I went into my barebones neighbor discovery file and began trying to implement it. The first thing I did was get the timer interface to work, then I called the `timer.sendPeriodic` in a function in `neighbordiscovery`, and called that command in `boot.booted`. At this point I was somewhat comfortable with how the files were interacting with each other, and I just had to use the `makepack` function to send a neighbor discovery packet on the broadcast channel, and then I made a response function and I was almost done. The last thing I had to do was create fields at the top of `neighbordiscovery` for storing a list of 19 floats, and I updated each of these floats to represent each node's individual neighbor confidence with an exponential weighted moving average.

To implement flooding, I asked `neighbordiscovery` for all of the neighbor confidence values, and sent a flood packet to each neighbor with confidence  $>.5$ . When the packet is received, I discard it if this node isn't the immediate destination, or if the node has already seen a higher or equal sequence number, and then I sent out another set of flood packets to every neighbor except the one who just sent it to me. This prevents the network from being completely overloaded by flood packets. I also implemented a time to live of 18, this is the maximum number of steps a flood packet could possibly need to take, because it is the # of nodes in the network - 1.

I also implemented a flood command in the python `testsim` file, and `commandhandler`, so I could verify that flooding was working.