

Juan Benitez
Sigi Lopez
12 February 2019

Project 1: Design and Decisions

The design idea behind flooding was to use a List data structure to store packets that were being sent and received allowing us to keep a record of what's happening in the network. We used the List with the type package because packages contain a lot of information such as Package source, Package destination and more.

In the implementation of flooding some of the main headers were TTL, source and sequence numbers. The sequence number would increment by one every single time a package made a hop into another node. In the other hand the TTL would be decremented by one every single time you would increment the sequence number. The larger the sequence number the further the location of the current package was from the original source. Note that when the TTL run out means that the package has lived long enough and it would be dropped. This insured that the packets would not go down a path that was never ending so we kill the package after a predefined time. When a new node would be visited, the package would be stored in the list of all the nodes, if the goal destination of that packet was not the visited package then the package would continue to be broadcasted. If a node has already been visited, then the received node will notice that the incoming packet has a larger sequence number than it should be, so it then knows that it is a redundant transmission and thus ends receiving. When the package reached its destination the TTL would be reset, sequence incremented and that node would send a acknowledgment back to the source header with a changed protocol from `PROTOCOL_PING` to `PROTOCOL_PINGREPLY`. While traversing back the seq will decrement by one for every node it encounters. When the package sequence reaches zero the package will arrive to its original source node.

In order to know when you will start doing neighbor discovery a new protocol was created(`PROTOCOL_NEIGH_DISC`) which was also added to the `protocol.h` file. What we were trying to do in discovery was to broadcast packets and also have a separate List to hold the all the direct neighbors broadcasted to. The difference between neighbor discovery and flooding is that a packet is only stored when a acknowledgment is received, this ensures that only direct nodes will be added to the list. Because we are not flooding everywhere the TTL in neighbor discovery is kept small after all the nodes in this section only have to jump once.

One of the specifications of the project was to run neighbor discovery in the background, but as we know packages will have to be sent throughout the course of time. To insure that packages do not collide each node was given a random periodic timer. This begins immediately when the boot is initiated and will follow the limitations the random number throws at the node. With this implementation we are able to account for nodes dropping, because like previously

stated It will periodically check for its neighbors and adjust the list of nodes with nodes that were able to send back an acknowledgment