

Jibram Jimenez-Loza

Alberto Cerpa and Ashish Yadav

CSE 160 - Computer Networks

11 February 2019

Project 1 – Flooding and Neighbor Discovery Discussion Questions and Design Choices

1. Describe a pro and con of using event driven programming.
 1. Pro:
 1. Several different objects can all simultaneously be listening for an event trigger and respond by running the event handler.
 2. Con:
 1. Can be more difficult to understand the flow of logic and requires much more connections for all to function. Order may sometimes be affected.
2. Flooding includes a mechanism to prevent packets from circulating indefinitely, and the TTL field provides another mechanism. What is the benefit of having both? What would happen if only had flooding checks? What would happen if we had only TTL checks?
 1. The benefit of having both minimizes the redundancy of passing the exact same payload over and over to the neighbors of each node. If we only had flooding checks, we would loop until it hit the end of the network at every edge. Having only TTL checks reduces the max propagation to whatever the programmer decides TTL to be.
3. When using the flooding protocol, what would be the total number of packets sent/received by all the nodes in the best case situation? Worst case situation? Explain the topology and the reasoning behind each case.
 1. Worst case:

1. Each node would send a two packets (ping and ping reply), but receive $4n$ packets back from every node in the graph it is connected to (all of them) because the entire network is maximally connected, aka, fully-connected graph.
2. Best case:
 1. Each node would send 2 packets (ping and ping reply) and receive 4 packets (2 redundant from ping and ping reply from the direction of the ping or the ping replier) because the entire network is minimally connected in a linear fashion, such as the topology provided to use in project 1. This could also be described as a linked-list. This is because as a message propagates from the first node out, and any one node will store the ping and the ping reply.
4. Using the information gathered from neighbor discovery, what would be a better way of accomplishing multi-hop communication?
 1. Having a neighbor list for each node, we could breadth first search the network (assuming all distances between nodes are equivalent) to find the node and track the route from original node to the desired node.
5. Describe a design decision you could have made differently given that you can change the provided skeleton code and the pros and cons compared to the decision you made?
 1. Separate the flooding from neighbor discovery and simply create a new module altogether for simply pinging individual nodes when desired.