1. **Group members**

Group members: Ziqi Liu 9364-4917

Enyang Wang 6494-1124

1. **How to run**

Steps: 1. Enter root folder (project3).

2.type commend as: **mix run project.exs numNode numRequest**

1. **What is working**

The goal of our project is to print the maximum number of hops (node connections) that must be traversed for all request for all nodes. We implement and test “route to node” function and “node insertion” function described in the paper.

1. Node insertion

The aim of “node insertion” is to build our network. There are 5 steps described in the paper to implement function “node insertion”

1. Find the most surrogate node of the new insert node
2. Get the greatest common prefix between surrogate node and new node
3. Copy the neighbor map from surrogate node
4. Use function *ACKNOWLEDGEDMULTICAST* to acknowledge other nodes in the network. Each node that is acknowledged should check their neighbor map then fill the “hole” by new node if need.
5. Recursively get neighbors from other nodes in the network that have same prefix.
6. Route to node

The aim of “route to node” function is finding the number of hops that one node must traversed to find another node. For a node that want to find another node, it checks the first level of its neighbor table. If there is a node that have same prefix with the starting node, it checks the second level of its neighbor table and find next nodes. Messages are forwarded across neighbor links to nodes whose nodeIDs are progressively closer until find the destination node. If a node cannot find a node that have same prefix at that level, it will find the other node in that level. If there is no node in that level, it will check next level in its neighbor table until max level. If there is still no node match the prefix, it will output an error.

1. **About max number of hops**

According to the algorithm described in the paper(routToNode), the maximum number of hops should be (*N* is the largest namespace of node ID, *b* is the radix of node ID). Since the SHA-1 is construct by hexadecimal and our node ID is the first 4 digits of SHA-1 code. Our maximum number of hops should be less or equal to

=4

In practical, the max number of hops will stable as 4 if the size of network is greater than 100.

1. **The largest network**

We assign each node ID by first 4 digits of SHA-1 code which encoded by node’s process ID. Therefore, the largest number of nodes is 16 to the power of 4, which equal to 65536 theoretically. However, the program requires long time to build the network. So, the largest number of nodes that we test are 4000.

The relationship between nodes number and time shown below (requests number is 10)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 500 | 1000 | 1500 | 2000 | 2500 | 3000 | 3500 | 4000 |
| time | 3.229 | 9.576 | 21.319 | 39.318 | 61.617 | 84.455 | 115.993 | 152.384 |