PROJECT 3: REPORT

Group Members

- Maharshi Rawal UFID: 9990-8457
- Mohit Garg UFID: 9013-4089

Implementation Description

Goal: The goal of the project is to find the maximum number of hops from one node to the other node in a tapestry-based network. The size of the actors and number of requests per actor is the input by the user.

Working: Our program uses Elixir GenServers to create multiple instances of new Tapestry nodes parallelly. It initializes the nodes and generates their hash ID's using SHA-1 hashing. The routing tables of each node is empty during initialization. Routing table of each node is filled once the whole network is established. The table is filled by the concept of prefix matching as given in the paper.

Once the initial nodes are set up in the network with their updated routing tables, a new node enters the network and finds its nearest neighbor by matching its prefix with the current nodes in the network. It then copies the nodes present in the matching node up to the level its prefix is matched and then computes the rest of its routing table from the nodes in the network.

To check for the maximum number of hops, all the nodes in the network send requests to random nodes in the network depending on the number of requests mentioned by the user. The max number of hops are computed and stored in the state of the main Tapestry GenServer.

Largest Network Tested

The largest network we tested was 9000 nodes which had a maximum hop count of 5 in the network.

Number of Nodes	Number of Requests	Maximum Number of Hops
10	5	1
100	10	2
100	20	2
500	20	3
1000	10	4
1000	20	4
2000	10	4
4000	10	4
7000	5	5
8000	5	5
9000	5	5