**Project 3 -Tapestry Algorithm**

COP5615 - Fall 2019

**Group Members**

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

**Implementation description:**

**Goal:** The goal of the program 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 GenServer to create multiple instances of new child parallelly. It initializes the all initial nodes and generate their hash ID’s using SHA-1 hashing. Then it creates all the children processes of supervisor one by one by creating their routing table. The routing table is completely empty during initialization. Then we call the current instance of that table and use MakeRoutingTable () method to fill the routing table based on their longest prefix matching logic. The Table is filled by traversing all the nodes one by one and check their position in the current node. Every node fills its routing table by checking its position in the current node.

After creating the routing table for all the initial nodes, it will create a node which will join the existing network by using Proj3.Node.newNodeRoutingTable() method . The new node will find the nearest node based on the longest prefix matching. It then stores the common prefix levels of the nearest node from the nearest node routing table. It then computes the rest of its routing table based on the existing hashNames.

Then the network spread the message across the network, each node chooses five random destination nodes from the existing network and pass the message for the hops counting. It then calculates the number of hops from the source to destination by using our Proj3.Route.route() method. //rest maharshi will add about the max hops logic

**Largest Network Tested**

Graphs will be pasted here