COP5615 – Fall 2019

PROJECT - 3

1. Group Members

Name	UFID
Floura Angel Nadar	2303-6958
Gurpreet Singh Nagpal	0698-9051

Steps to run:

1. Unzip the file and navigate inside the folder
2. Open Terminal (with elixir installed)
3. For running:
   Use the following command on Windows: mix run main.exs < numNodes> < numRequest>
   numNodes=Number of nodes
   numRequest= Number of requests

What is working

1. Tapestry generally uses a 40 digit hash which leads to a very sparse network when we test for the limited number of nodes (<100000), hence to encounter this we have used a smaller hash of **8 digits**. The hash values being generated are unique. These hash values have been mapped to process id and re-calculated incase another with same id is formed.
2. To **build the routing tabl**e we have used a map of map data structure where the outer map represents the number of rows (0 to 8) and the inner map represents the number of columns(0 to 15)
3. The designed application has two main processes the Boss process and the process which represents the nodes. Initially, the boss process is called to build the network. Once all the nodes have their routing table initialised the boss process initiates the node join process. While filling the routing table, especially in the case of the 0th level where multiple nodes can be inserted at a single position we have considered the one which is closest to the node whose routing table is being built, this is done using the distance calculation method.
4. The **node joi**n process is carried on as per **Dynamic Node Algorithm** (section 3 C) of the research paper. The **parent** is searched by the longest prefix match, the parents routing table is updated first then the routing table of the parent is copied to the routing table of the newly joined node, until the level of p(longest match level) from 0. Further, the nodes up to the level where new node is inserted in the parent are updated.
5. Finally, the **routing** is initiated each node randomly selects one node at a time(total number of nodes selected are equal to the number of requests, the nodes could be a neighbor or could be some distinct node which is reaching by using different number of hops). Each node maintains a counter to keep track of the number of requests it has been completed. Further, the nodes report their hop count to the boss process by casting a reply to the boss function.
6. The boss process filters the max hop count by comparing the hop count it receives to the previous max value it has stored. While performing the routing we have also **implemented the surrogate routin**g mentioned in Routing and Object Location Section (Section 3B).

What is the largest network you managed to deal with?

Due to system constraint could not run for bigger values, it runs but very slow but tried and tested for below:

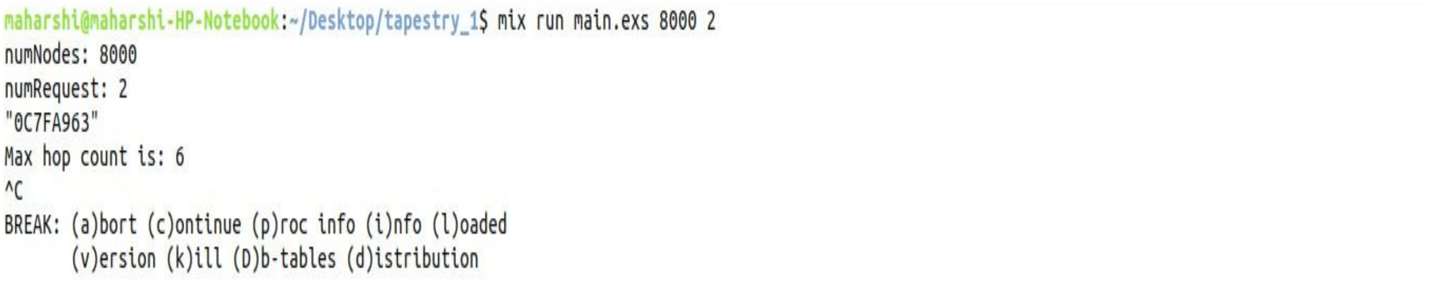
Number of Nodes	**Number of Requests**	**Max Hop Count**
10	2	1
100	2	3
1000	2	5
4000	2	5
7000	2	6
8000	**2**	**6**
10	10	1
100	10	3
1000	10	5
4000	10	5
7000	10	6
100	20 - 100	3
1000	20	5
4000	20	5
7000	**20**	**6**

Maximum Number of nodes tested: 10000

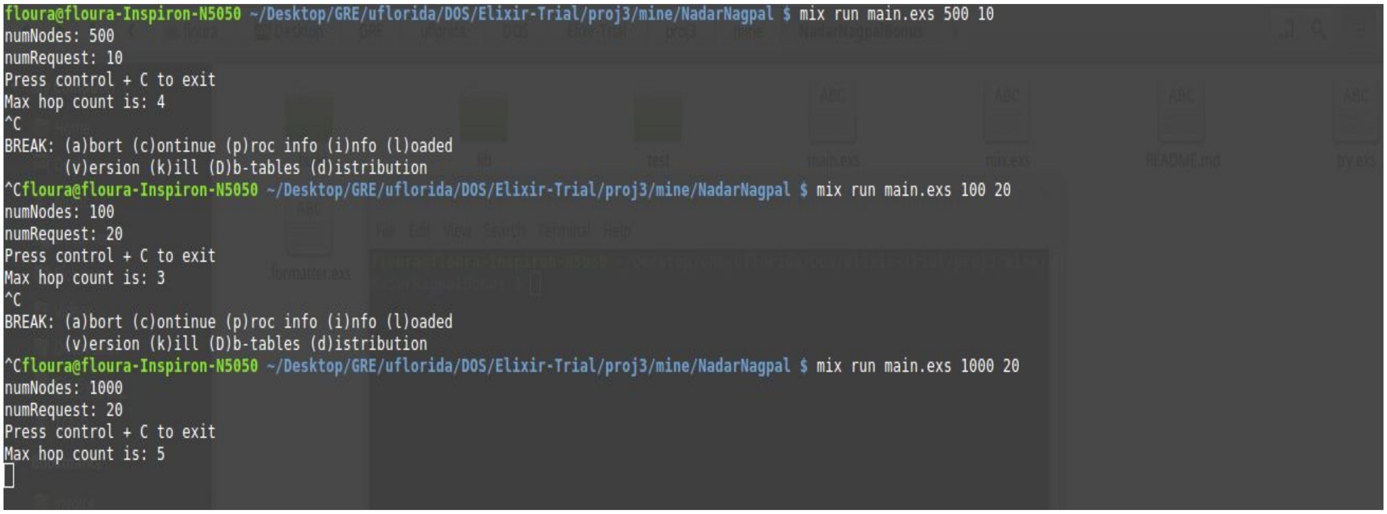
Maximum Number of requests tested: 20 requests on a 7000 node network

The maximum number of hops for all the networks tested never increases the number of levels.

Screenshots:



```
maharshi@maharshi-HP-Notebook:~/Desktop/tapestry_1$ mix run main.exs 8000 2
numNodes: 8000
numRequest: 2
"0C7FA963"
Max hop count is: 6
^C
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded
       (v)ersion (k)ill (D)b-tables (d)istribution
```



```
floura@floura-Inspiron-N5050 ~/Desktop/GRE/uflorida/DOS/Elixir-Trial/proj3/mine/NadarNagpal $ mix run main.exs 500 10
numNodes: 500
numRequest: 10
Press control + C to exit
Max hop count is: 4
^C
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded
       (v)ersion (k)ill (D)b-tables (d)istribution
^Cfloura@floura-Inspiron-N5050 ~/Desktop/GRE/uflorida/DOS/Elixir-Trial/proj3/mine/NadarNagpal $ mix run main.exs 100 20
numNodes: 100
numRequest: 20
Press control + C to exit
Max hop count is: 3
^C
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded
       (v)ersion (k)ill (D)b-tables (d)istribution
^Cfloura@floura-Inspiron-N5050 ~/Desktop/GRE/uflorida/DOS/Elixir-Trial/proj3/mine/NadarNagpal $ mix run main.exs 1000 20
numNodes: 1000
numRequest: 20
Press control + C to exit
Max hop count is: 5
```