PROJECT 3 - PASTRY

COP5615

TEAM MEMBERS:-

1. Monica Mavoori (UFID – 46309228)
2. Rohini Kar (UFID – 67308301)

Command line arguments given:

numNodes numRequests

where numNodes is the number of peers in the system and numRequests is the number of requests that each peer has to make.

As an example,
Run the program as following

```
scalac Project3.scala
scala Project3 10 5
```

where number of nodes are 10 and each node has to send 5 messages.

Working:

The program works as follows-

Initially, the network is empty and there are no nodes in it. When we provide the argument for number of nodes, that many number of nodes will join the network with their unique node id generated by hash method. Each node upon joining the network builds up its own routing table and leaf set. When the routing of messages start, a node chooses a random destination node id and sends a message to it. During routing it first checks if the destination node id is present in its leaf set. If so, it sends the message directly to that node or else it searches in its routing table. If there is a node in the routing table whose node id matches with that of the destination id by at least one more digit than that with the current node, then the current node sends the

message to that node. Otherwise, it sends the message to a node which is numerically closer to the destination node id than the current node id.
Count for the number of hops is incremented as each node receives a routing message and also keeps track of number of messages sent so as to find the average number of hops at the last. When all the actors have sent the input number of requests each, the system shuts down.

The code is working for number of nodes given a small integer. It is taking a lot of time on increasing the nodes.

The largest network managed to deal with is 30 number of nodes. Beyond this the run time of the program is more.

Output:

It prints the average number of hops.

Example:

```
 Process for all nodes completed, system exiting
Average number of hops:5
```