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><failure percentage>
   numNodes=Number of nodes (should be more than 10)
   numRequest= Number of requests
   failure percentage = Percentage of failure nodes (not more than 100)

What is working

1. In order to implement failure, we introduced a failure percentage. We randomly select the nodes that fail as per the percentage. We maintained a list of these failed nodes.
2. Once the routing starts we first check in the failure list. If a node exists in the failure list we find its parent and divert the request to its parent. Hereby parent we mean a node with maximum prefix match. Or it could be considered as **backup node**. So one node can **at max have 3 parents** basically we are maintaining 3 nodes at a slot. (As mentioned in the paper)
3. Incase a node has no parent i.e a node with prefix match the request is diverted to any random node in the network.
4. There could be a case where the parent could die to handle that case we look for the parent if it exists in the live node

Flow of work

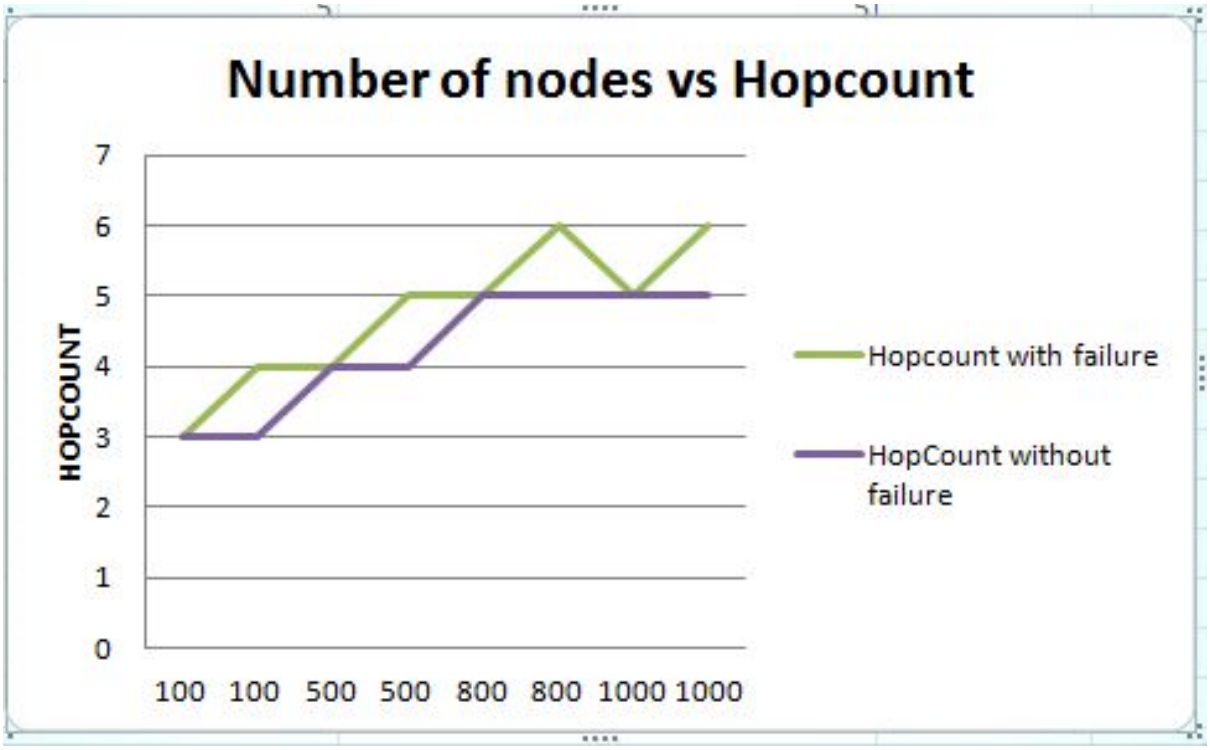
1. First we determine the number of failure nodes using the failure percentage mentioned
2. Then we determine randomly which node to fail and send it to those nodes that fail
3. The node will then create a backup node at most 3 nodes and save it in it's state.
4. When a route is done and it encounters the failure node, it simply uses the backup or the backpointer and routes through that.
5. Every node can maintain it's parent node.

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	Failure percentage	Hopcount with failure	HopCount without failure
100	10	3	3
100	80	4	3
500	10	4	4
500	40	5	4
800	20	5	5
800	90	6	5
1000	50	5	5
1000	80	6	5

Observations: As the number of percentage of failure increased the number of hop count kept increasing.

Graphs:



Screenshots: