Project Bonus Report 2

Date: October 7th 2017

Project Team mates

1. **Shubham Agiwal- UFID 20562669**
2. **Karan Sharma- UFID 00174451**

Topologies Implemented

1. **Full Topology -** A node has all the other nodes in the network as its neighbors.
2. **Line Topology-** A node knows only its immediate front and back neighbor except in the case of first and the last node.
3. **2d Topology-** Actors form a 2D grid. The actors can only talk to the grid neighbors.
4. **Imperfect Topology-** Grid arrangement but one random other neighbor is selected from the list of all actors (4+1 neighbor).

Failure Models Implemented

For the given project we have implemented two failure models which are defined below:

Before Kill

In Before kill, we ask the user to enter the integer percentage for nodes you need to kill before starting the protocol. The main process keeps track of the all the alive nodes in system. Based on the input, we kill a percentage of the random nodes in the architecture for different algorithms like push sum or gossip respectively. They are removed from the alive node list maintained in the main process. We then try to pick an alive node from the list and start the protocol on different topologies and see if they are able to converge based on convergence criterion.

Convergence criterion for **gossip**-we have assumed that convergence is reached when all the nodes have heard the rumor for the given gossip count or different nodes has sent continuous multiple dead messages for the number of nodes**.**

Convergence criterion for **push sum**-we have assumed that convergence is reached when all the nodes have reached an s/w ratio of less than 10^-10 for three consecutive counts

After Kill

In After kill, we ask the user to enter the integer percentage for nodes to be killed after starting the protocol. The main process keeps track of the all the alive nodes in system. Here when a node dies or terminates, it sends a cast to the server process that it's dead. The server process removes the node from the alive node list maintained at the server. After that, it picks up a random alive node in the list and kills it for the respectively algorithm. This mechanism is applied for different algorithms for different topologies to see if the convergence criterion for the given algorithm is met.

Convergence criterion for **gossip**-we have assumed that convergence is reached when all the nodes have heard the rumor for the given gossip count or different nodes has sent continuous multiple dead messages for the number of nodes**.**

Convergence criterion for **push sum**-we have assumed that convergence is reached when all the nodes have reached an s/w ratio of less than 10^-10 for three consecutive counts

Analysis and Observations

The above failure models were applied to small networks ranging from 100-500 to large scale networks ranging to 1000 nodes for both gossip and push sum algorithm.

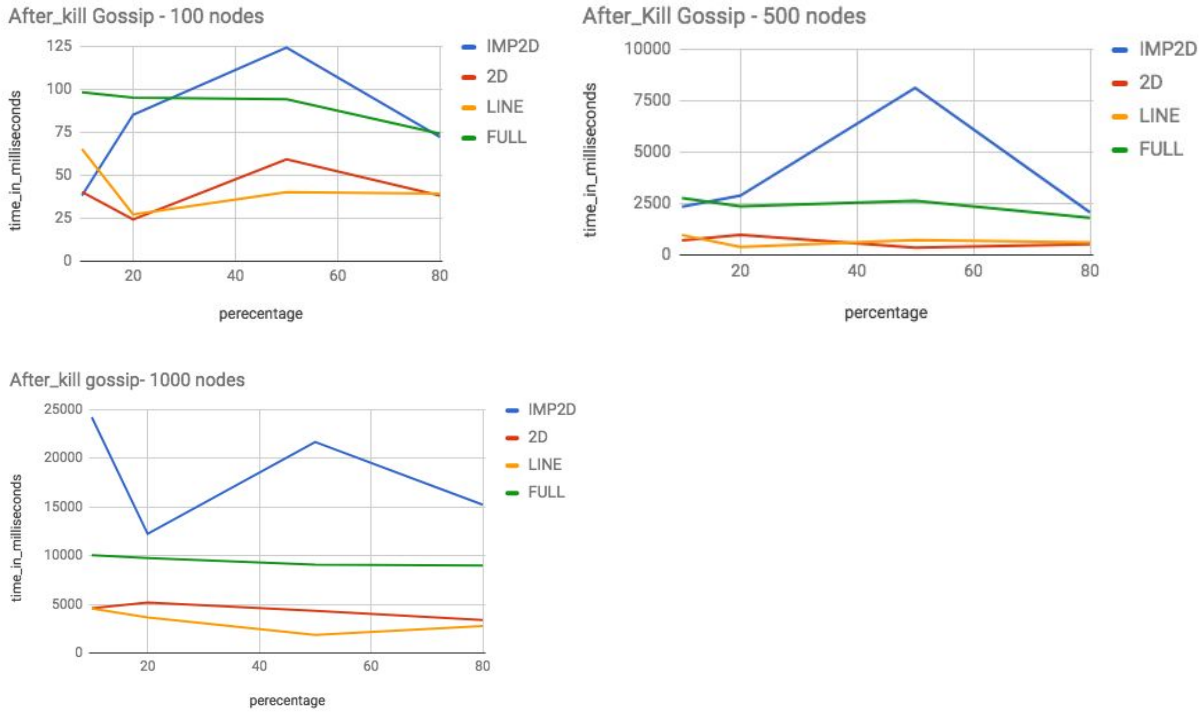


Fig 1. Plot showing convergence vs percentage of node failure for networks ranging from 100 to 1000 for Gossip Algorithm for after kill process.

From the above graphs we can conclude that from small networks ranging from 100 to 500 and large networks, there is no significant difference in the convergence for topologies full,2D and line on the gossip algorithm. Although we can see that there is a significant increase for the convergence for the imp2D topology for 50% after kill process.

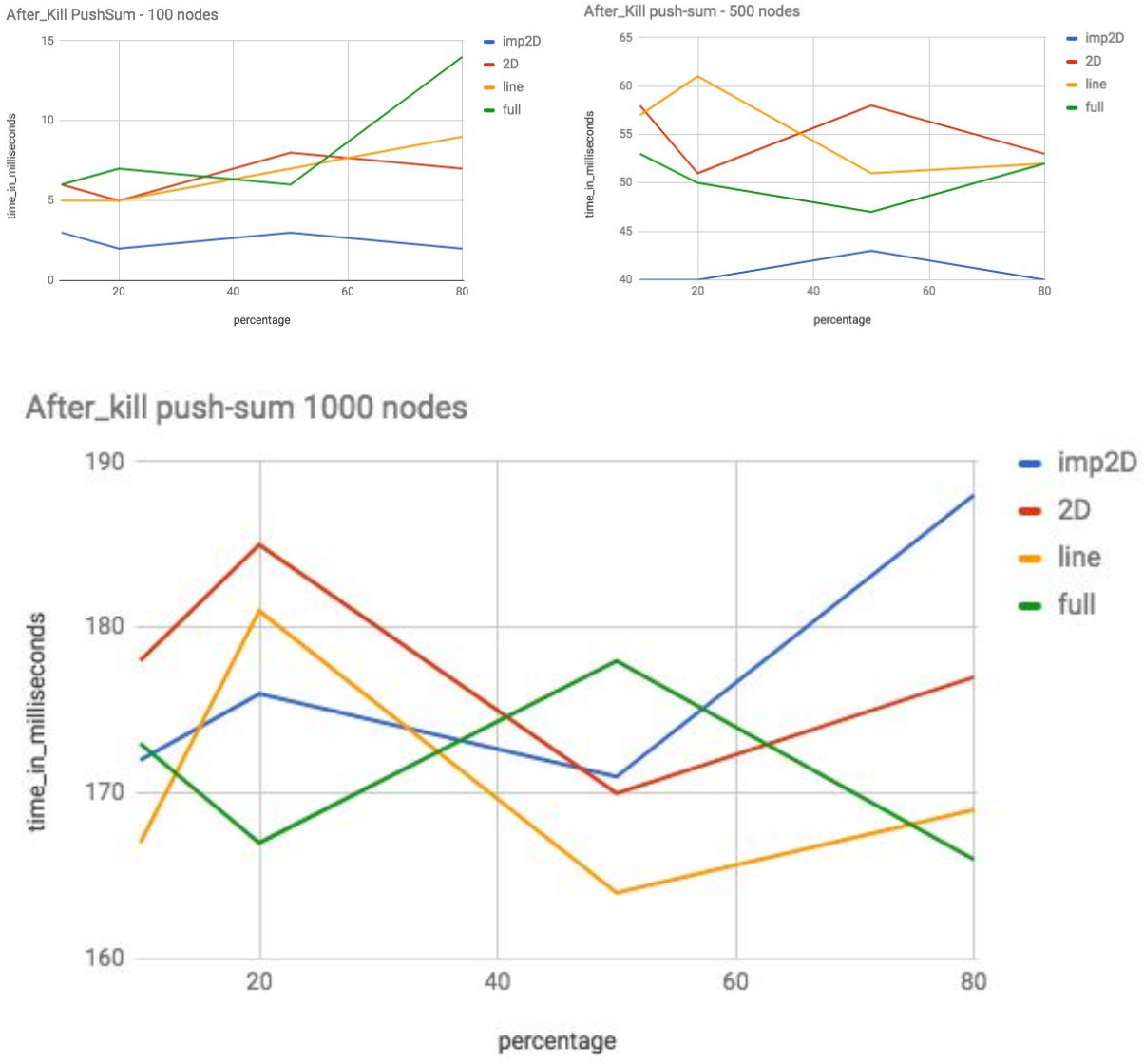
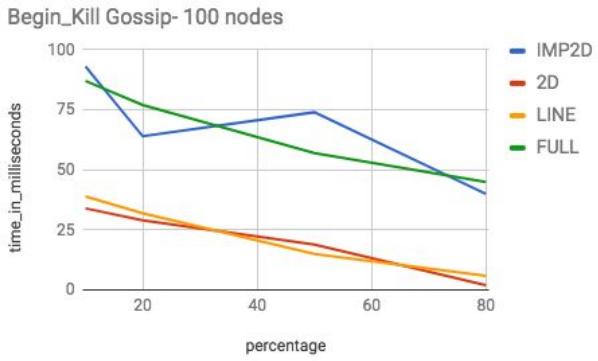
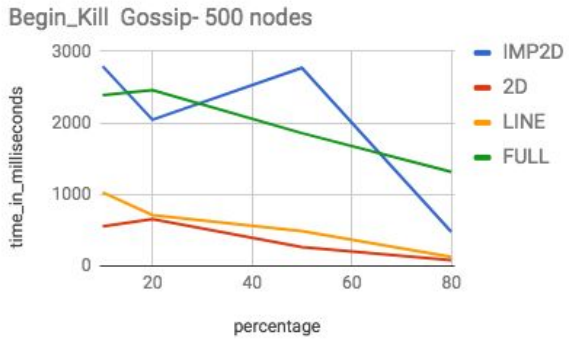


Fig 2. Plot showing convergence vs percentage of node failure for networks ranging from 100 to 1000 for Push sum Algorithm for after kill process.

The one interesting observation from the above graphs is observed for the imp2D topology. For small networks, its behavior is that the convergence time will decrease initially and then increase until the 50% of nodes are killed and then decrease. But as the number of nodes keep increasing the network, the inverse of the above mechanism is observed.





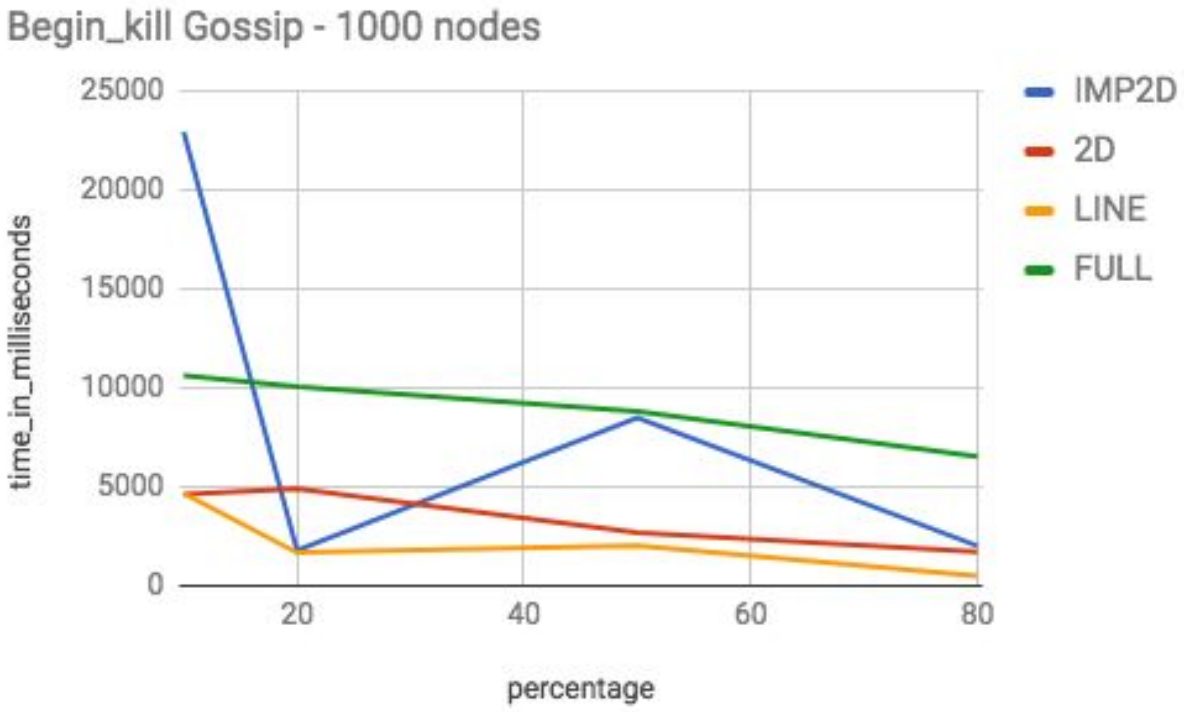
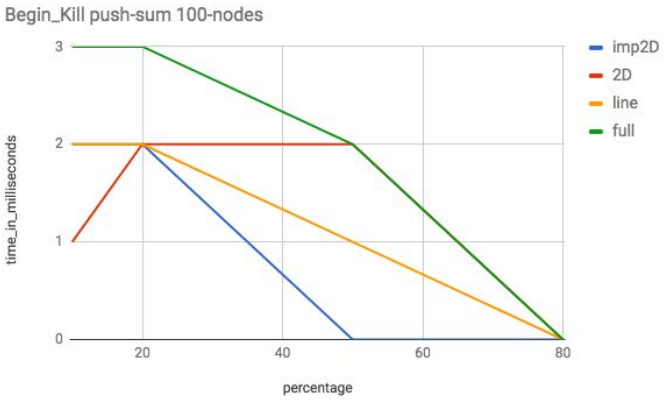
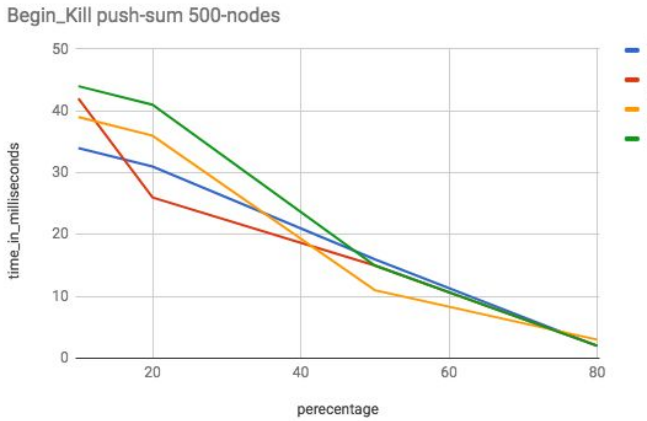
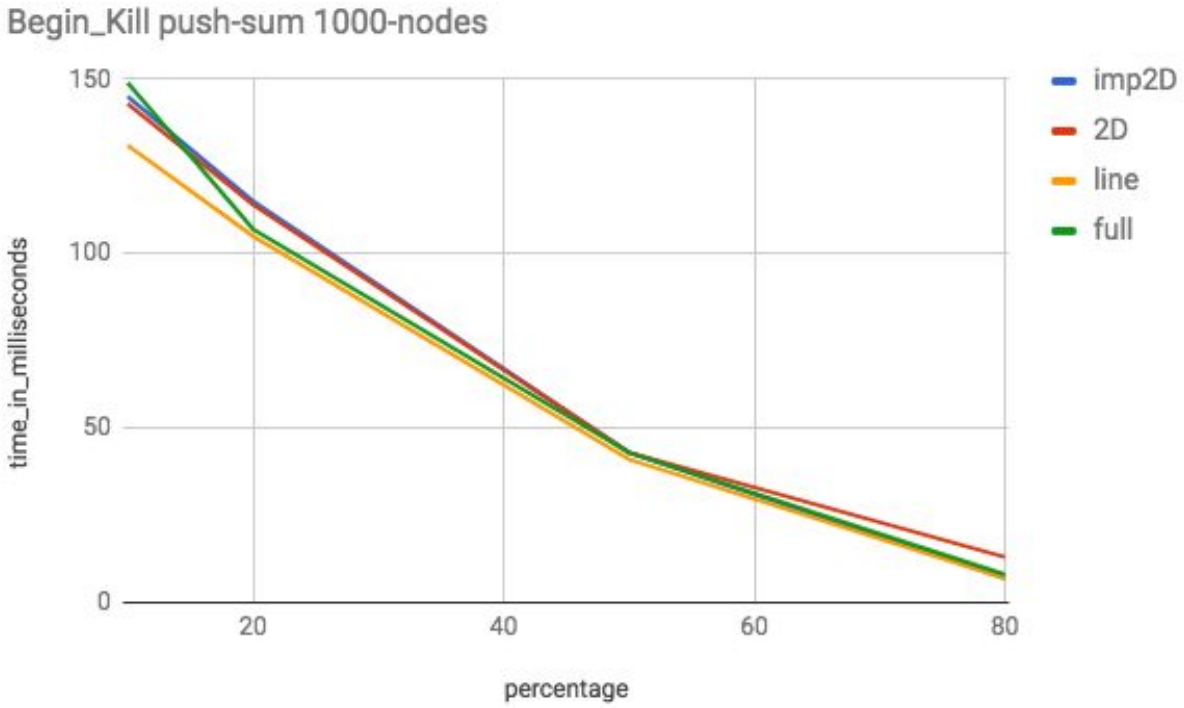


Fig 3. Plot showing convergence vs percentage of node failure for networks ranging from 100 to 1000 for gossip Algorithm for begin kill process.

The one interesting observation from the above graphs is observed for the imp2D topology. For small networks, its behavior is that the convergence time will decrease initially and then increase until the 50% of nodes are killed and then decrease. But as the number of nodes keep increasing the network, the inverse of the above mechanism is observed.







Plot showing convergence vs percentage of node failure for networks ranging from 100 to 1000 for push sum Algorithm for begin kill process.

The one interesting observation from the above graphs is observed is that all the topologies have a uniform decrease in the achieving the convergence time for small and large networks.