Gossip Simulator(Project-2 Bonus)

Distributed Operating Systems

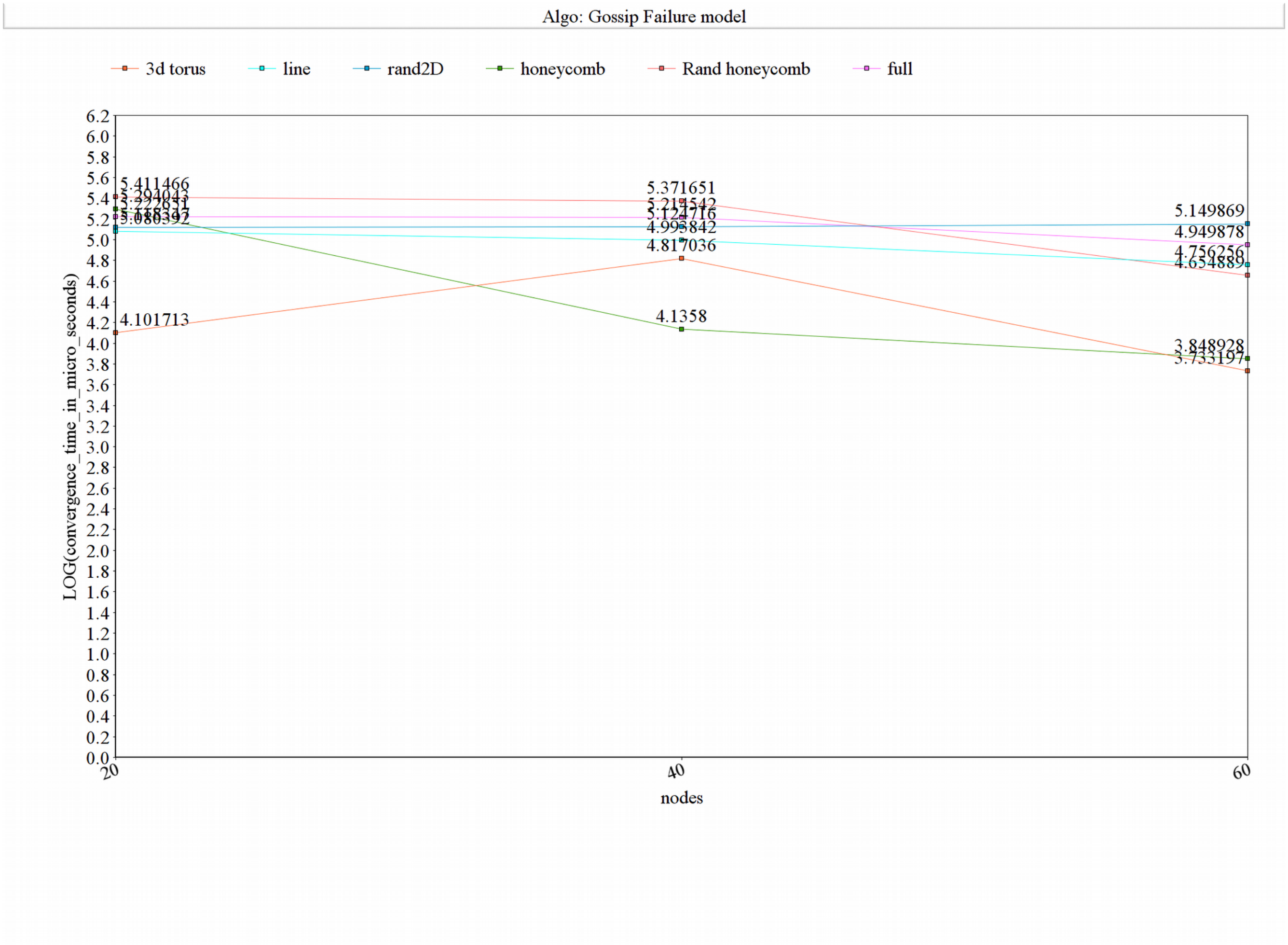
Group Members:

Chandan Chowdary (UFID-6972-9002)

Gayathri Manogna Isireddy (UFID-9124-0699)

Project Description and Approach:

To implement the failure model we have considered Gossip algorithm approach. The project gets started like how it starts in normal project. But in this failure model we analysed the performance of the network by introducing a parameter which the failure nodes. We performed failure model by killing 20%,40%,60% of nodes that are given as input. These failure nodes gets killed before the start of the algorithm. Below graph shows the failure percentage and the time that the network has taken to converge.



X axis is the percentage of nodes that are killed before the start of the algorithm and Y axis shows the logarithmic time that the network has taken to converge.

The experiment is conducted on the input nodes that is given as 100. Out of 100 nodes 20 nodes are killed at first and checked the convergence time for all the topologies and in the same way 40 and 60 nodes are killed sequentially. The time of convergence is noted down and is plotted on the graph.

Observations:

Generally the performance of any topology should improve as the number of nodes are getting reduced. But this is not the case. The performance depends entirely on the network.

Lets say we have line network. If we kill a node which is in middle of the network, then indirectly we are killing the node that connects the two halves of the network. Hence if this node gets killed there wont be any connection to other part and hence messages wont be sent to the nodes in that part. If this is the case the topology wont get converged and gets stuck.

For full network these failure models doesn't matter in the fact of convergence. As all nodes are connected with each other and if a node dies the message passing wont get stopped and convergence will be achieved as usual.

For random 2D, the case is the same as full network is all the nodes are densely connected. But as this algorithm is random, it assigns nodes as neighbours which are nearer to that. It could also be possible that 2 or more independent groups can be formed by killing a connecting node. In this case it is similar to line as there wont be any connection between the groups.

For honeycomb,randomhoneycomb and 3D torus, it is possible that some nodes are left behind without convergence. As the neighbour degree is very less we cannot expect convergence all the time.