DOS PROJECT 2 BONUS REPORT

Team Members:

Sanchit Deora (8909 – 4939)
Rohit Devulapalli (4787 – 4434)

Aim:

The aim of this project is to implement the Gossip Algorithm and the Push-Sum protocol for the following topologies:

- Full Network
- Line Network
- Random 2D Grid
- 3D Torus Grid
- Honeycomb Network
- Random Honeycomb Network

Failure Model:

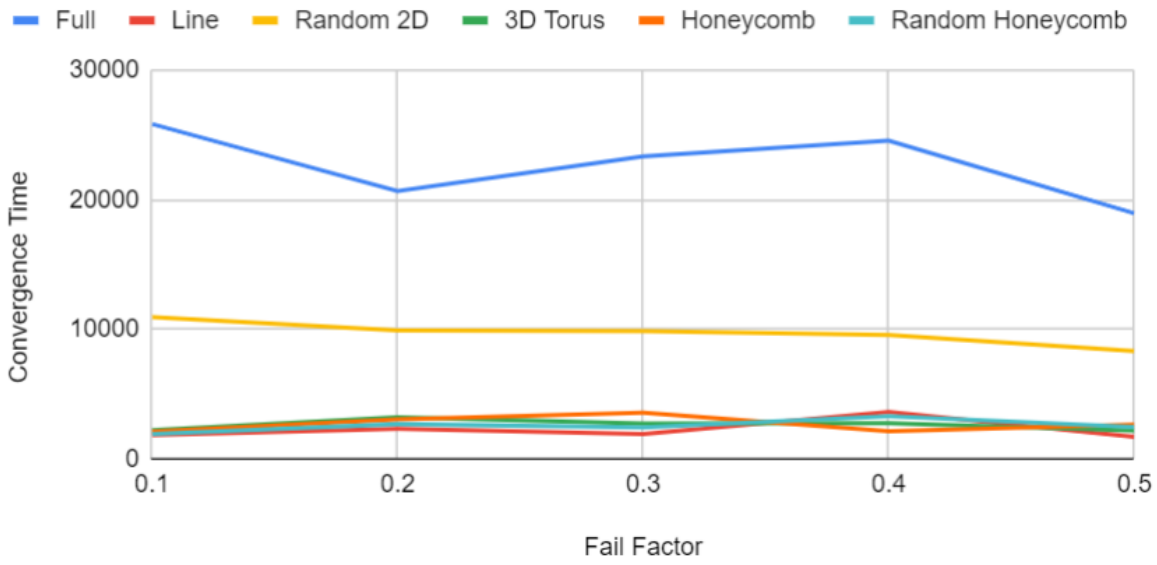
We have implemented a Permanent Failure Model. In this model, there is a Master Actor which is asleep for a random time. This Master GenServer once awake, gets the number of nodes (**numNodes**) in the network. Parameter (**failFactor**) is set to 0.2. This **failFactor** is multiplied to **numNodes** which determines the number of nodes to be failed (**toBeFailed**) in the network permanently. These nodes are selected at random, and are discarded from the network.

Observations:

We executed the Gossip Protocol and PushSum algorithm for different values of kill ratios for each topology keeping the numNodes fixed (= 1000 for Gossip Protocol; = 500 for Push Sum Algorithm).

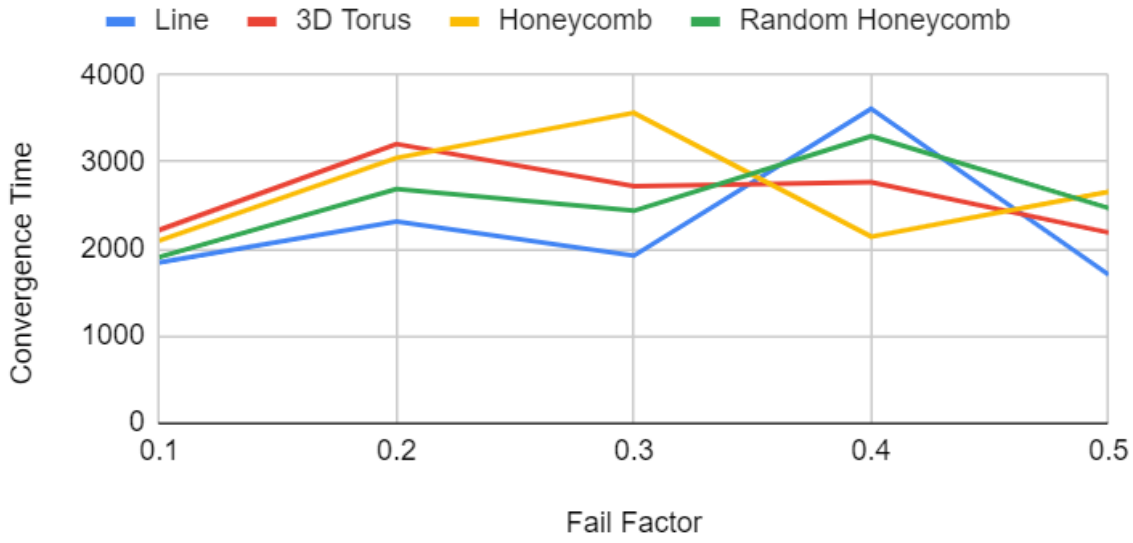
Fail Factor vs Convergence Time (Gossip)

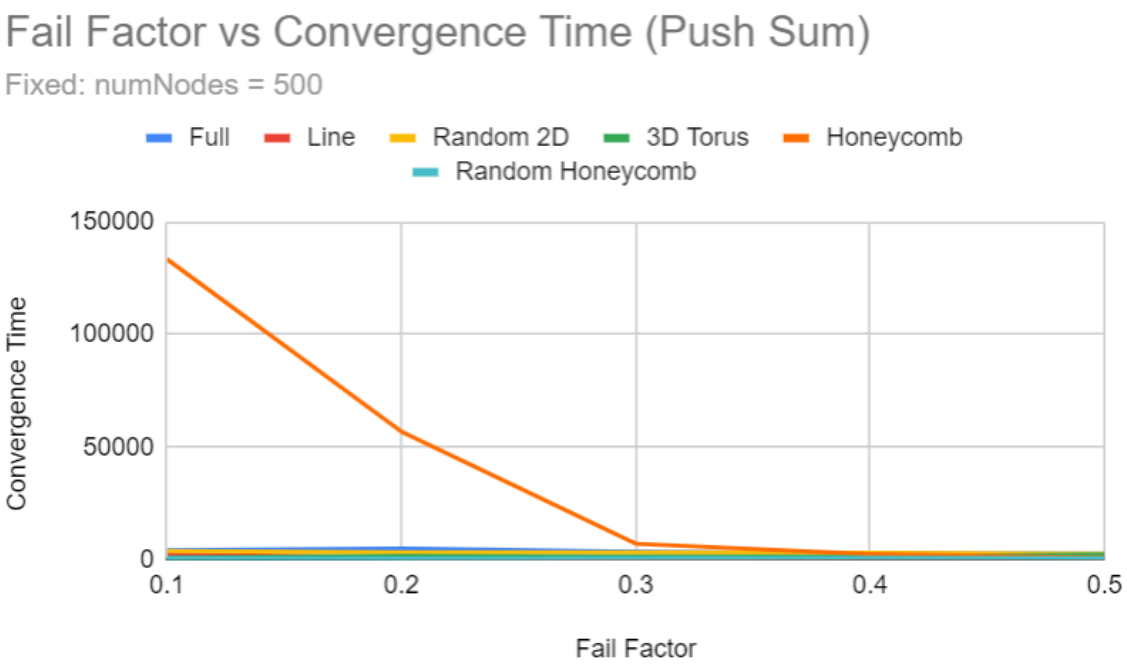
Fixed: numNodes = 1000

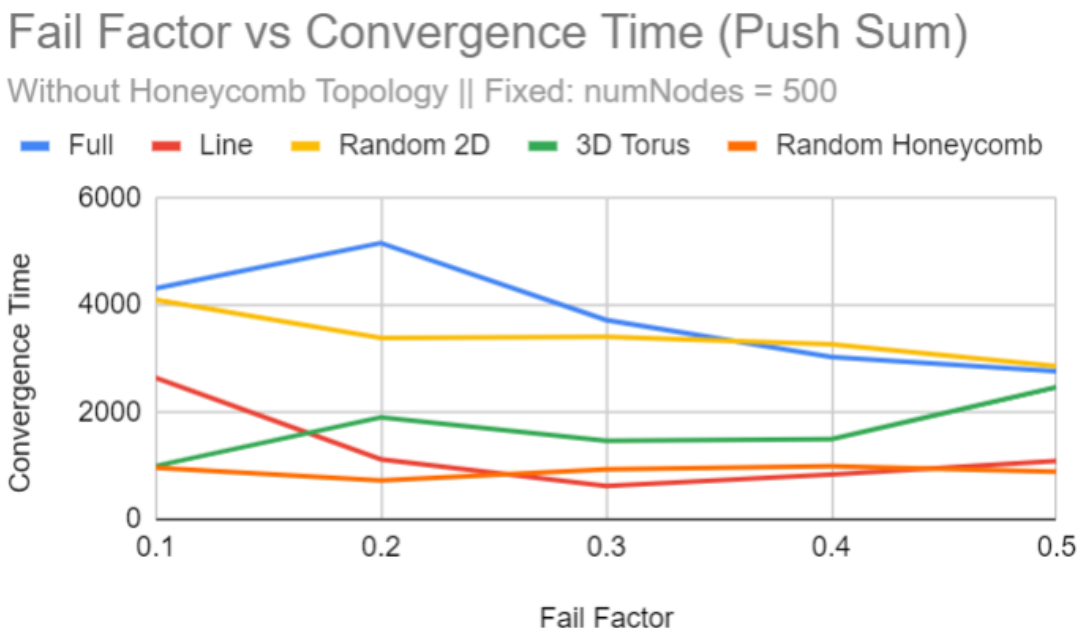


Fail Factor vs Convergence Time (Gossip)

Without Full and Random 2D Networks ||Fixed: numNodes = 1000







From the figures(Figure 1, Figure 2) above, the convergence time does not vary by a significant factor when the kill failure ratio is further increased from 0.1. For Gossip protocol, full network is taking largest time to converge for all the fail factors.

In case of pushsum, the convergence time of honeycomb topology decreases significantly as we increase the fail factor. The convergence time of random honeycomb network is minimal across all the failure factors.

This means that the pushsum and gossip algorithms very resilient to permanent node failure. The same trend is observed across all topologies. We can also see that even a sparsely connected topology like line topology exhibits the same behavior as

that of a strongly connected network like full topology for our failure model.