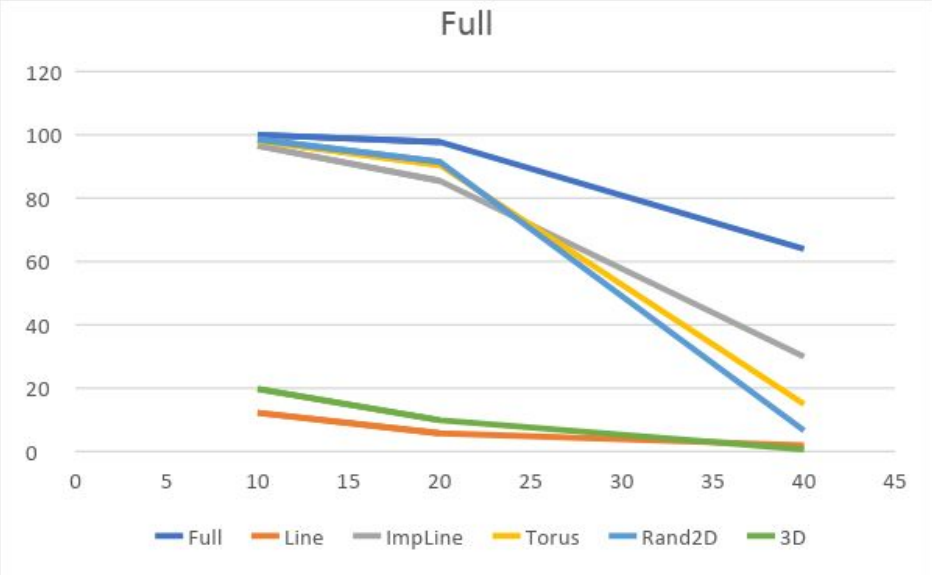
Bonus Graphs and Inferences

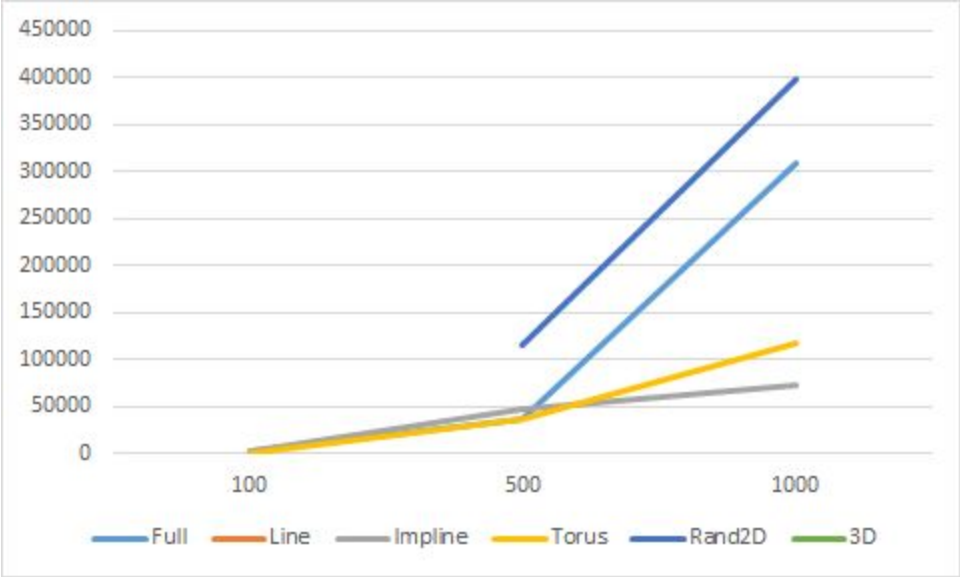


Implementation: Upon reaching a dead node, the parallel processes of other nodes sending/receiving are allowed to continue by keeping a check on convergence count. A timer every 5 seconds checks if the count has increased. The algorithm stops once the count ceases to increase.

Bonus Graph for Gossip: The graph is made as percentage of nodes covered(y-axis) vs percentage of nodes deleted(x-axis). We kept the number of nodes static as 500 for all topologies and then killed 10, 20 and 40% percent nodes and checked how many nodes were being converged.

Observation: As the number of dead nodes increased, the convergence number also decreased drastically for most topologies except full.

The performance was exceptionally poor in 3D and Line.



Push Sum Bonus Graph: The graph is made as number of nodes vs convergence time with 10% nodes killed for the node inputs.

Implementation: Once a dead node is reached, a non-dead neighbour is searched for and the algorithm continues. Convergence is reached for smaller number of nodes with a very high increase in time. We killed 10% nodes for 100, 500 and 1000 inputs.

Inference: 3D and Line don't reach convergence after 100 nodes. Other topologies reach convergence but take an exceptionally large amount of time when compared to no failure situation. Imperfect Line performed better than Torus and Full which took too much time with bigger nodes.