PROJECT-2-BONUS REPORT

Project partners:

1. Wins Goyal, UFiD – 73571559, *winsgoyal@ufl.edu*
2. Rahul Wahi, UFiD – 30536162, *rahul.wahi@ufl.edu*

Failure model for Gossip and Push-sum Algorithm:

- Failure Model

For the bonus part of this project, we have implemented one failure model for the algorithms and the topologies. In this model, we are passing a new parameter through CLI called 'fail-percentage'. Fail-percentage tell us the percentage of nodes in the topology that should go inactive. This percentage varies from [0,100]. As the program-flow reads this parameter, the supervisor will kills the random no. of nodes equal to the given percentage of Failing nodes. Once inactive/killed, the actors will stay as such and won't restart, meaning that they are permanently killed.

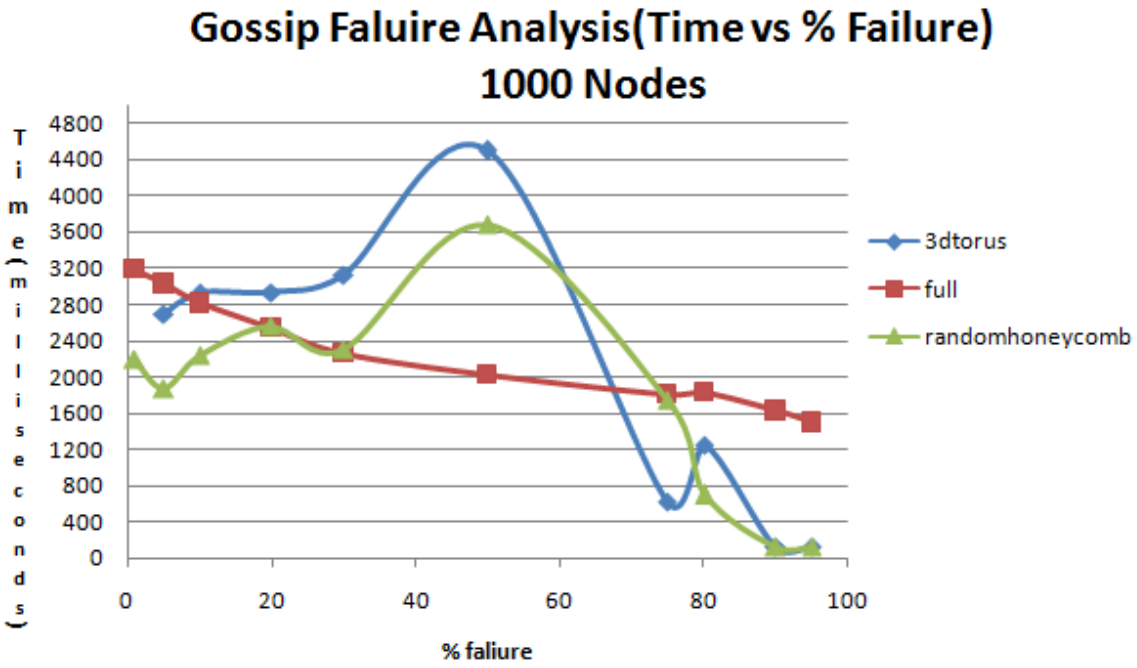
First all the actors get generated by the Supervisor → Genserver, and based on what topology and algorithm to run, the corresponding neighbors for each Actor process are set. The supervisor generates the Genserver according to the algorithm parameter value. Then after, the fail-percentage is used to calculate the no. of nodes to fail. This value is used to randomly select the nodes from the Actors lists, that should fail. The created list of nodes to fail is sent to the corresponding Genserver which will ask the supervisor to kill the given list of nodes to fail. Once inactive, the Actor PIDs and the corresponding neighbour lists are maintained as such. We just start the rumor process by starting the Gossip or Push-sum process.

- Analysis method

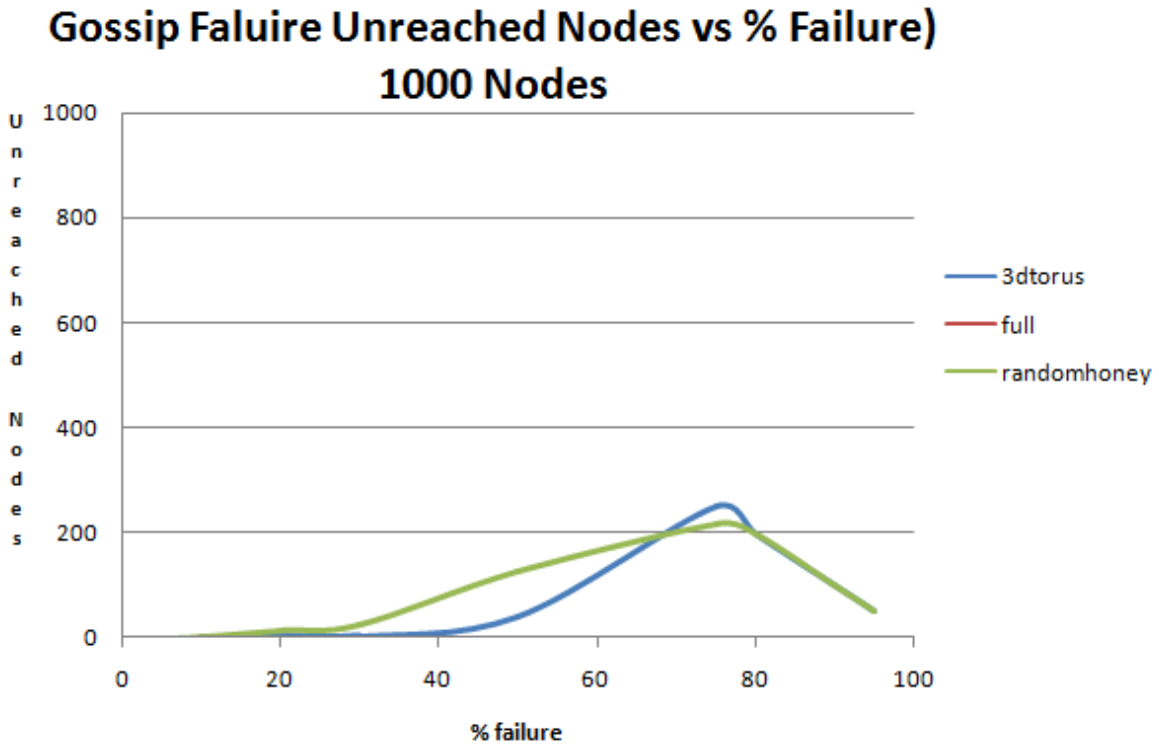
After running the program on CLI, we get two outputs → Convergence time and the number of nodes unreached. The latter tells how many nodes were not reached before the network converged with the given fail percentage. This gives us an idea about the spread of the rumor throughout the network. So, we plot two graphs for each algorithm. In one, we show how the "convergence time" varies w.r.t "fail percentage" for a given number of nodes in some of topologies. In other, we show the number of unreached nodes (opposite of the spread) w.r.t the varying values of "fail percentage"

Different Plots/experiements:

- For Gossip Algorithm,

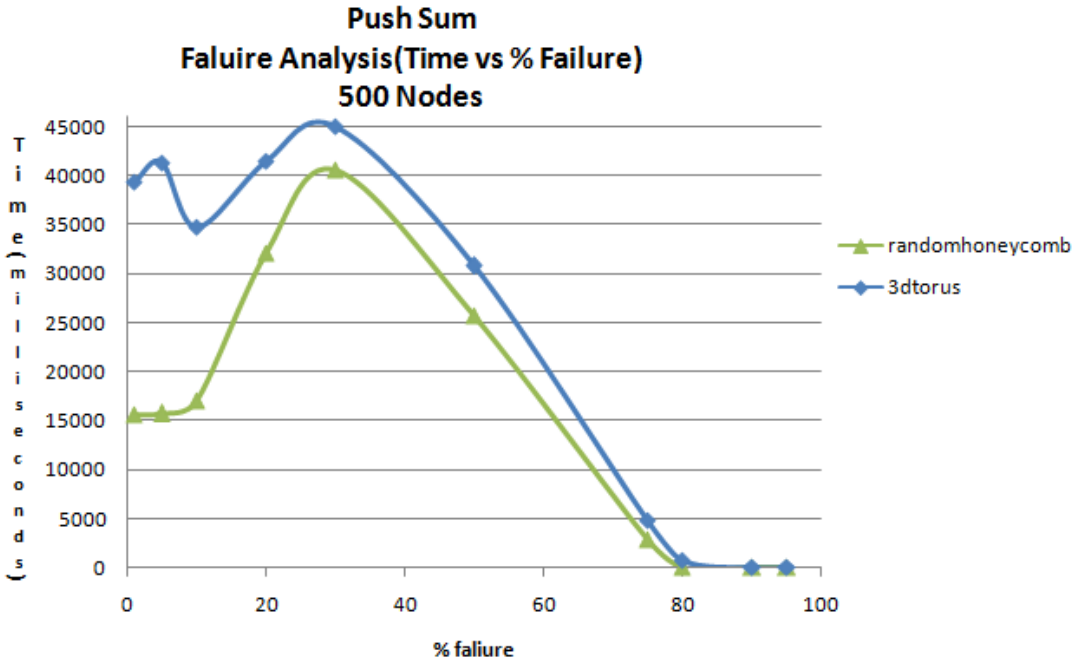


(Fig. 1) – Convergence Time vs % failure for 1000 nodes in *3D torus, full* and *Random honeycomb*

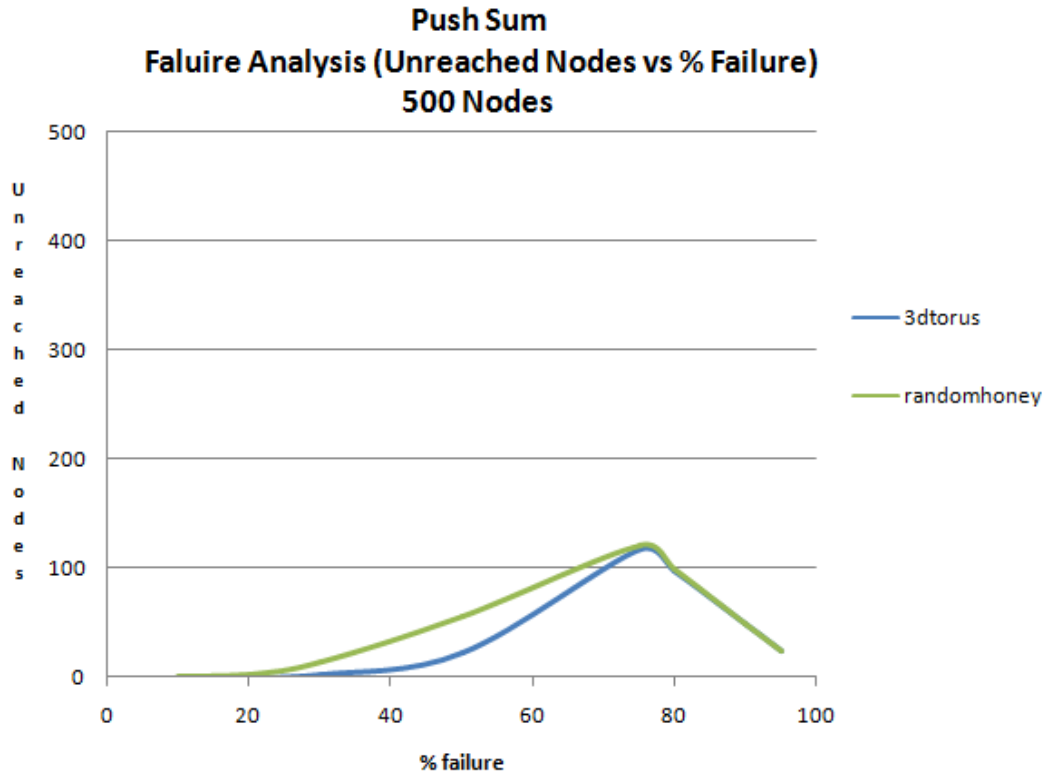


(Fig. 2) – No. of unreached nodes vs % failure for 1000 nodes in *3D torus, full* and *Random honeycomb*

- For Push-Sum Algorithm,



(Fig. 1) – Convergence Time vs % failure for 500 nodes in *3D torus* and *Random honeycomb*



(Fig. 2) – No. of unreached nodes vs % failure for 500 nodes in *3D torus* and *Random honeycomb*

Observations & Inferences:

- In Gossip Algorithm, interesting observation is found for *Full* network, and simple interpretation can be made for observing such behavior.
    (a) The convergence time is gradually decreasing with the increase in Failure percentage. While, the graph is not like that of a Bell curve, this graph tells that the network will converge with lesser and lesser time, as the nodes will always have some neighbors which are not inactive and connectedness in the graph will be maintained, and message has to be spread to lesser and lesser nodes as well.
    (b) With the higher failure percentage, the network will be sparse enough but still connected, making it converge even faster. This is also the reason why no distinguished curve is visible in the Fig(2) of Gossip Algorithm, i.e. because the no. of unreached nodes = 0, always. So, the full network always has a spread of upto 100% during the convergence.
- For both the Algorithms, *3Dtorus* and *random-honeycomb* show the bell-curve in Fig(1)'s. This tells that:
    (a) With the increase in failure percentage, convergence time first increases and then it drops.
    (b) In Gossip, max. convergence time is attained at around 50% fail percentage while in Push-sum, the max. convergence time is attained at around 35% fail percentage.
    (c) Random-honeycomb still performs better than 3Dtorus, for higher number of active nodes or if fail-percentage is less than 50%.
    (d) Convergence time approaches zero when the Fail percentage > 80%
- In Fig(2)'s for both algorithms, again the plots are bell-curves.
    (a) We get the maximum no. of unreached nodes for some fail-percentage. From both the plots, we find that the max. no. of unreached nodes is when fail-percentage is 70-80%. So, the convergence time is the least because the rumor spread is the least at this point.
    (b) After 80%, the no. of unreached nodes decreases because the no. of active nodes have significantly reduced by this time.