COP5612 – Fall 2015
Project 2 – Gossip Simulator
Report

Points to note:

1. For every number_of_nodes, we have run the program 5 times and considered average of the 5 results as the result plotted on graph
2. Considering the fact that there might not be a convergence time for the system as a whole, we have considered the convergence time when system doesn't converge any more for 30 seconds

Findings & Observations:

1. The order in which convergence takes time for a typical topology as per our observation:
   Full (fastest) < Imperfect3D < 3D < Line (slowest)
   Reason: We conclude that the more active neighbors you have, the faster the convergence is

2. The order in which convergence takes time for an algorithm as per our observation:
   Gossip < Push-Sum
   Reason: Compared to a simple self-counter for Gossip, Push-Sum implements a stringent check on message variation and accordingly increments self-counter.

3. As number of nodes increase, for a given time *'t'* the convergence rate achieved in time *t*:
   Full (most) < Imperfect3D < 3D < Line (least)
   Reason: This is mainly because the topologies convergence time discussed in observation #1 above

[BONUS]

As part of bonus failure model implementation, we have included checks in actor gossip calls to avoid sending messages to the nodes that have closed down. This is in line with the concept of Gossip wherein, the rumor stops disseminating if the corresponding neighbor is down (failed).

Observation: In case the gossiper node randomly selects a neighbor which is down (failed), the convergence time for the gossiper node increases because it has to go through the process of selecting an active neighbor.

Please find below graph that depicts convergence time as a function of the size of the network for each of the algorithms*

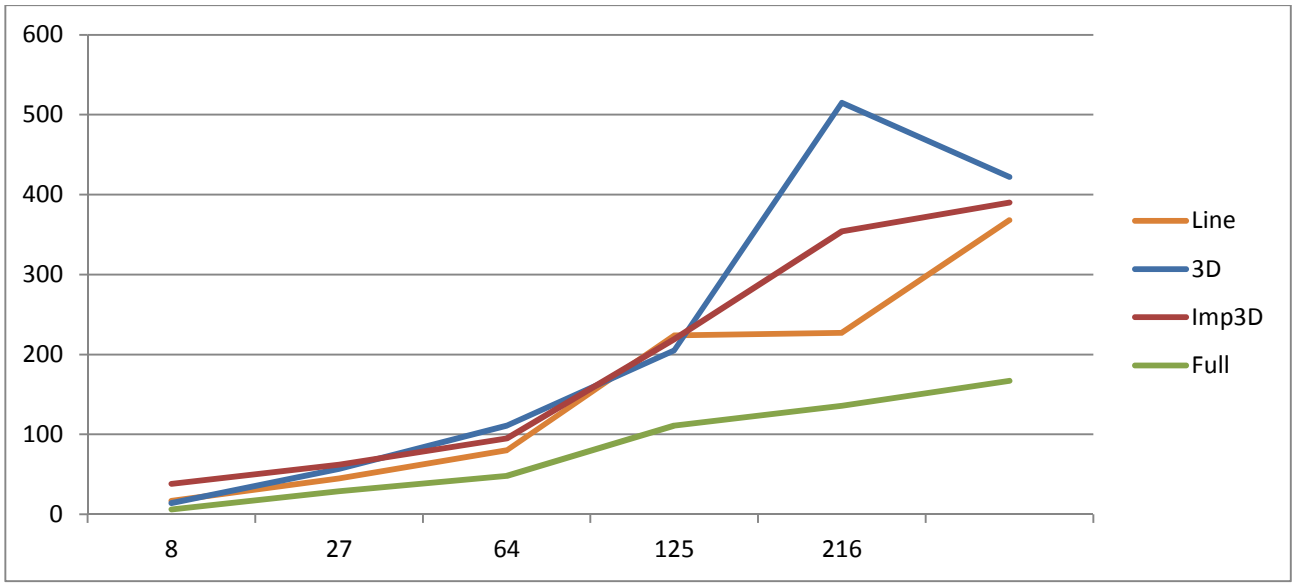
**Disclaimer: The graph just plots convergence time vs network size. In our opinion, this graph cannot be used to correctly interpret the performance of the (algorithm,topology,network size) combination as there is no guarantee that we'll encounter convergence of 100% of the nodes. Therefore, we have also included the percentage of nodes converged in our findings (table).*

Gossip:

The below table illustrates the number of nodes and the time it took for the nodes to converge for Gossip algorithm.

Nodes	Line		3D		Imperfect 3D		Full	
	Time(ms)	% convergence	Time(ms)	% convergence	Time(ms)	% convergence	Time(ms)	% convergence
8	17	62.5	14	87.5	38	100	6	100
27	45	77.7	57	96.2	62	96.2	29	100
64	80	87.5	111	100	95	100	48	100
125	224	96.8	205	99.2	219	96	111	100
216	227	97.68	515	99.07	354	97.2	136	99.5
343	368	99.4	422	99.4	390	98.5	167	100

Graph to represent the convergence time (vertical axis) taken as a function of number of nodes(horizontal axis):



Push-Sum:

The below table illustrates the number of nodes and the time it took for the nodes to converge for Push-Sum algorithm.

Nodes	Line		3D		Imperfect 3D		Full	
	Time(ms)	% convergence	Time(ms)	% convergence	Time(ms)	% convergence	Time(ms)	% convergence
8	483	75	808	75	1052	87.5	18	87.5
27	2245	81.45	4128	74.07	4343	88.88	91	96.29
64	7572	54.69	6414	81.25	6430	82.81	194	98.44
125	2617	20	9346	74.4	9873	71.2	205	99.2
216	3061	10.18	9261	56.48	24523	68.52	304	99.54
343	5132	16.29	38871	40.82	23947	57.72	595	99.71
512	808	99.8						
729	1014	99.8						
1000	1841	99.9						
1331	6476	99.92						

Graph to represent the convergence time (vertical axis) taken as a function of number of nodes(horizontal axis):