

PROJECT REPORT

Team Members:

Saurabh Verma - 60874081

Adil Shaik - 69985592

This project has implementations of Gossip and Push-Sum algorithms with the following topologies - Fully Connected, Line, Random-2D, 3D Torus, Honeycomb and Random Honeycomb.

We set the convergence criteria for gossip algorithm of about 90%. i.e, the timing shown here is when 90% of the nodes have received the rumor more than 10 times.

Below are the findings:

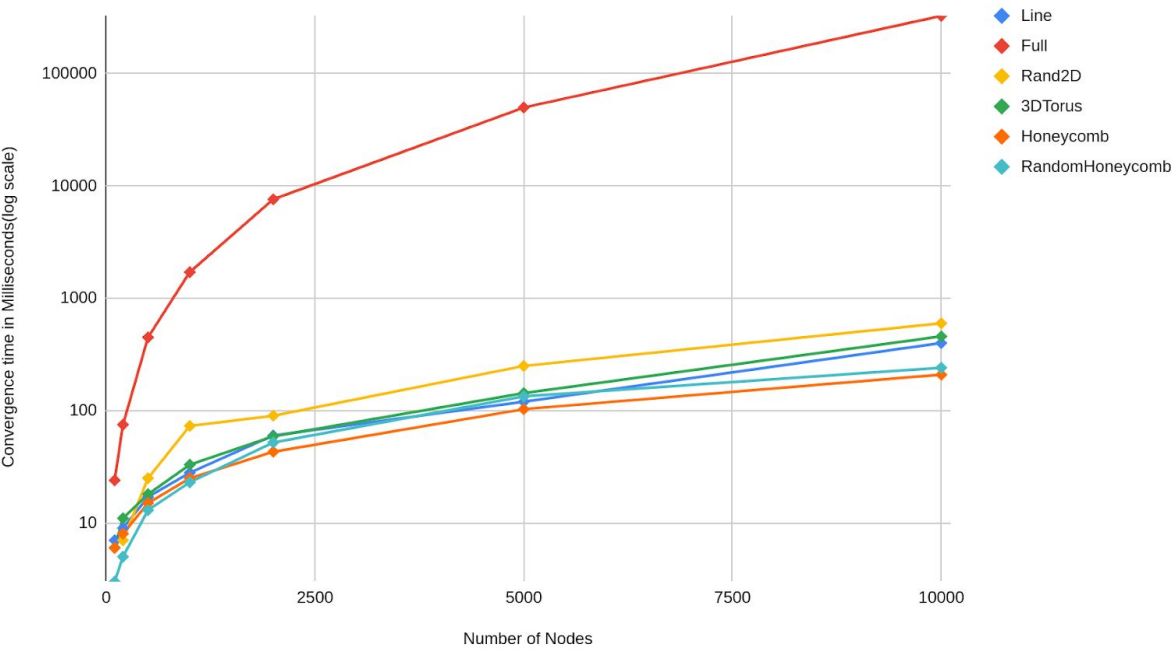
Gossip (Time in milliseconds)

Node Count	Line	Full	Rand2D	3DTorus	Honeycomb	RandomHoneycomb
100	7	24		6	6	3
200	9	75	7	11	8	5
500	17	449	25	18	15	13
1000	28	1712	73	33	25	23
2000	60	7615	90	59	43	52
5000	120	50034	250	143	103	134
10000	400	327884	600	458	209	241

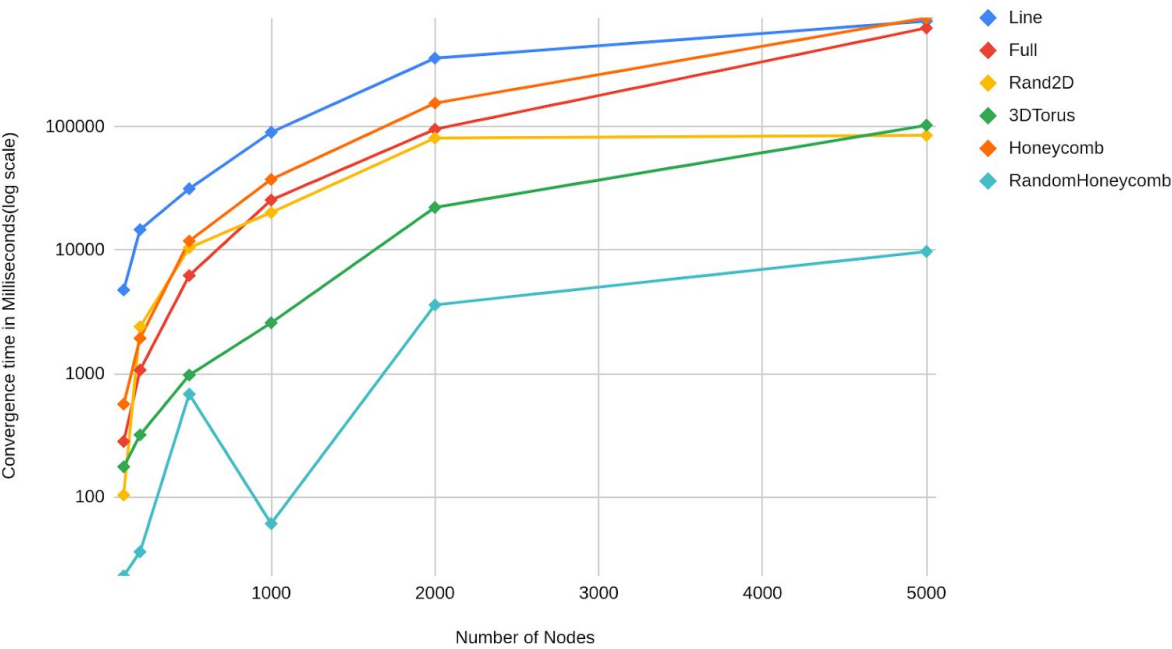
Pushsum (Time in milliseconds)

Node Count	Line	Full	Rand2D	3DTorus	Honeycomb	RandomHoneycomb
100	4771	282	104	176	566	23
200	14705	1071	2397	319	1936	36
500	31510	6238	10469	974	11889	683
1000	90445	25538	20212	2581	37538	61
2000	360221	95647	81167	22245	155725	3611
5000	720423	632800	85342	103040	765811	9772

Gossip Algorithm Performance



Push Sum Algorithm Performance



OBSERVATIONS-

- All the topologies implemented in this project, work well.
- The least-time consuming algorithm for push-sum turns out to be RandomHoneyComb.
- RandomHoneyComb is several orders of magnitude faster than normal HoneyComb for push-sum. Just adding one random neighbour, causes the convergence to be faster. It was also observed that we were obtaining different convergence times with the same number of nodes. If we get lucky, the algorithm converges in milliseconds. Other times it took several seconds. It also seemed that the topology might show faster convergence for larger node count. Maybe the random neighbour causes it to be somewhat less dependent on the node count. We tried experimenting with node sizes 500 and 1000. For these two cases, either of the two might be faster to converge on random trials. This can be seen in the graph itself as a “kink”
- Line topology doesn’t get converged several times for gossip. Sometimes we got lucky and it converged in milliseconds other times, it didn’t converge at all. We were totally dependent on the random function to get convergence out of line. If a corner node gets pinged and it stops, then the transmission got stopped altogether.
- The most-time consuming algorithm for push-sum was the line topology. It was the slowest in convergence when compared with other topologies.
- In gossip algorithm, when taking higher number of nodes, the fully connected topology performs the worst. It has the highest convergence time