Project 2 Bonus Report

Aniketh Sukhtankar (UF ID 7819 9584) Shikha Mehta (UF ID 4851 9256)

Robustness of Gossip

I implemented a failure model in which I randomly deleted a certain number of nodes (taken as input from user) and saw how much time the remaining number of nodes took to converge. Convergence in this case was achieved when 100% of the remaining nodes received the gossip message at least once. The network had 100 nodes initially.

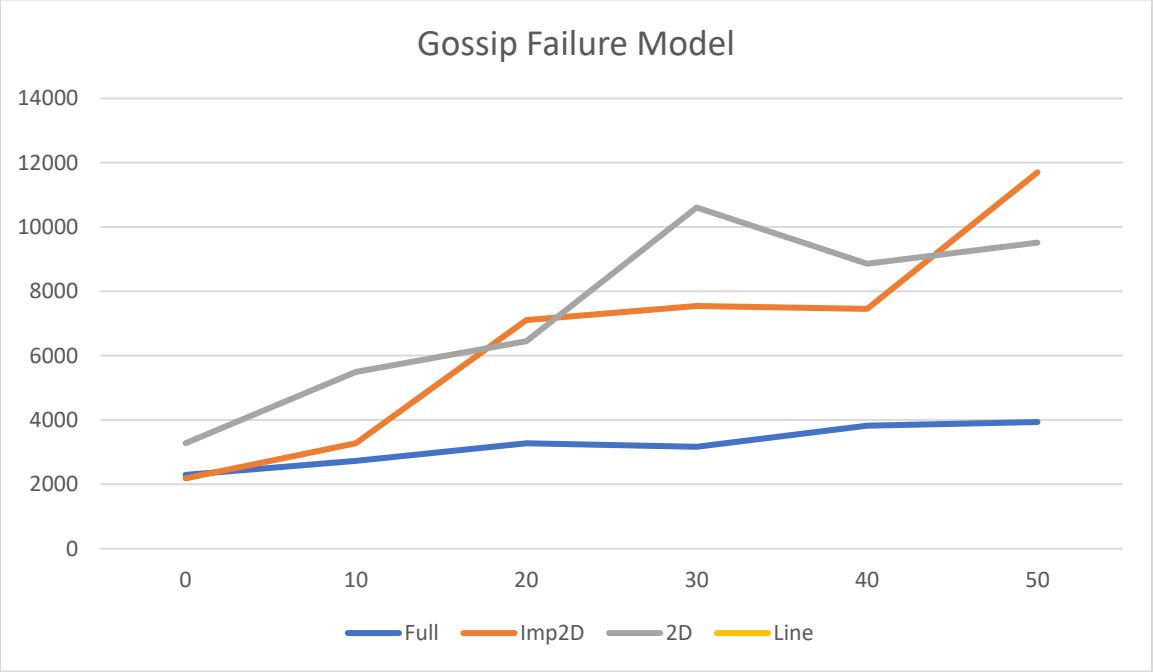
- Even the removal of a single node from line topology causes the network to fail to reach convergence. This was tested by killing a certain number of processes and checking the percentage of nodes that converged.
- After removing one node from the network in line, it breaks into 2 different networks.
- E.g.: Let us assume there are 100 nodes in the network and the randomly chosen node to start the gossip is 3. The randomly chosen node to kill is node 50. In such a scenario all the nodes from 1 to 49 will converge and the rest will not receive any gossip message.
- As such killing even 10 – 20 % of nodes in line topology will cause it to not converge.

Gossip Protocol Failure Model Convergence Times

Percentage Nodes Killed	Full	Imperfect 2D	2D	Line
0	2296	2187	3281	11078
10	2734	3281	5500	4484*
20	3281	7110	6453	3515*
30	3172	7546	10610	2078*
40	3828	7453	8860	2406*
50	3938	11703	9516	2625*

Gossip Protocol Failure Model Percentage of nodes that failed to converge

Percentage Nodes Killed	Full	Imperfect 2D	2D	Line
0	0	0	0	0
10	10	10	10	70
20	20	20	20	74
30	30	30	30	93
40	40	40	43	98
50	50	50	51	94



- The full,2D and Imperfect 2D topologies handled node failures extremely well and almost always converged irrespective of how many nodes were killed.
- Full was the fastest at converging, followed by Imperfect 2D and then 2D.
- The shocking part was that even after deleting 90% of the nodes in the network the gossip message was received by all the remaining nodes in each of these 3 networks. Moreover, the convergence occurred extremely quickly as shown in the graph.

Robustness of Push Sum

I implemented a failure model in which I randomly deleted a certain number of nodes (taken as input from user) and saw how much time the remaining number of nodes took to converge. Convergence in this case was achieved when 100% of the remaining nodes achieved the Push Sum criteria. The network had 100 nodes initially.

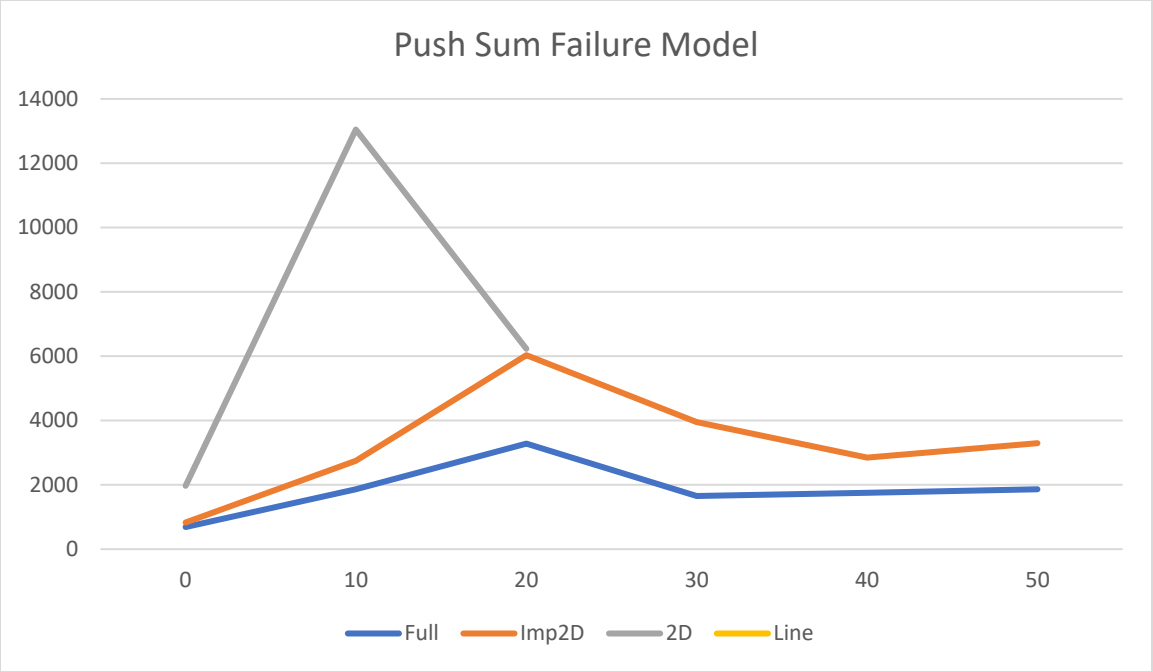
- Even the removal of a single node from line topology causes the network to fail to reach convergence. This was tested by killing a certain number of processes and checking the percentage of nodes that converged.
- After removing one node from the network in line, it breaks into 2 different networks.
- E.g.: Let us assume there are 100 nodes in the network and the randomly chosen node to start the gossip is 3. The randomly chosen node to kill is node 50. In such a scenario all the nodes from 1 to 49 will converge and the rest will not receive any gossip message.
- As such killing even 10 – 20 % of nodes in line topology will cause it to not converge.
- The more the number of neighbors -> denser the network -> more fault tolerant it is.
- Full topology is the most fault tolerant followed by Imperfect 2D and then 2D. Line being the least fault tolerant among all.

Push Sum Protocol Failure Model Convergence Times

Percentage Nodes Killed	Full	Imperfect 2D	2D	Line
0	687	828	1969	3500
10	1859	2750	13047	1969*
20	3281	6031	6234	1656*
30	1656	3953	4390*	875*
40	1750	2843	6391*	766*
50	1859	3297	125*	344*

Push Sum Protocol Failure Model Percentage of nodes that failed to converge

Percentage Nodes Killed	Full	Imperfect 2D	2D	Line
0	0	0	0	0
10	10	10	10	86
20	20	20	20	85
30	30	30	30	93
40	40	40	67	96
50	58	50	99	98



- The full and Imperfect 2D topologies handled node failures extremely well and almost always converged irrespective of how many nodes were killed. 2D had a hard time converging when over 30% nodes had failed while line would not converge even for small failures.
- Full was the fastest at converging, followed by Imperfect 2D.
- The shocking part was that even after deleting 90% of the nodes in the network the push sum values of s and w were received by all the remaining nodes in each of these 3 networks. Moreover, the convergence occurred extremely quickly as shown in the graph.