

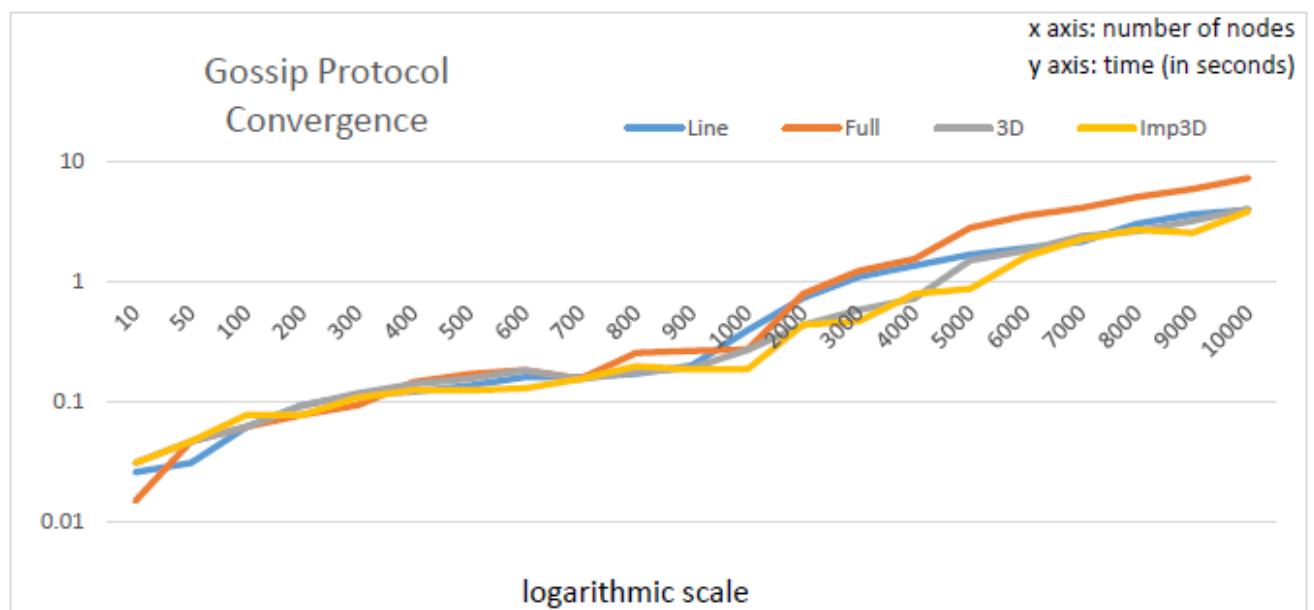
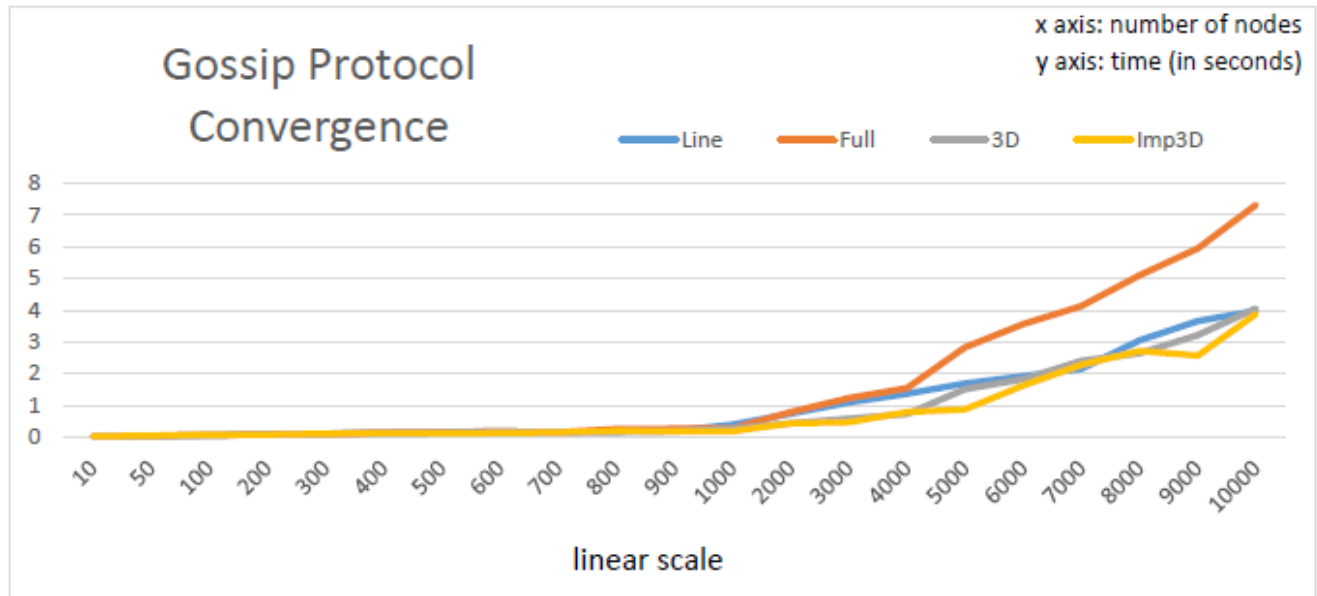
PROJECT 2: GOSSIP SIMULATOR

Team Members:

Mahesh Kumar Mahadev

Shantanu Kande

GOSSIP ALGORITHM



Tabular Data:

No. of Nodes	Line	Full	3D	Imp3D
10	0.026	0.015	0.031	0.031
50	0.031	0.047	0.047	0.047
100	0.062	0.062	0.062	0.078
200	0.094	0.078	0.093	0.078
300	0.114	0.094	0.118	0.109
400	0.122	0.147	0.141	0.125
500	0.137	0.171	0.156	0.125
600	0.161	0.185	0.183	0.13
700	0.161	0.156	0.155	0.155
800	0.172	0.256	0.177	0.199
900	0.202	0.266	0.191	0.187
1000	0.395	0.273	0.272	0.188
2000	0.738	0.797	0.439	0.44
3000	1.099	1.234	0.582	0.469
4000	1.368	1.549	0.727	0.794
5000	1.69	2.831	1.512	0.877
6000	1.92	3.567	1.833	1.627
7000	2.156	4.123	2.404	2.28
8000	3.051	5.094	2.644	2.719
9000	3.655	5.945	3.219	2.573
10000	3.981	7.317	4.051	3.879

The above graphs represent the convergence for the Gossip algorithm for the following topologies: Line, Full, 3D Grid, Imperfect 3D grid. (Note: The first graph is on a linear scale while the second graph is on a logarithmic scale for feasibility).

Observations and Findings:

Full topology: The full topology takes the maximum time for convergence among the given topologies

Line Topology: The line network is faster than full and slightly faster than 3D grid.

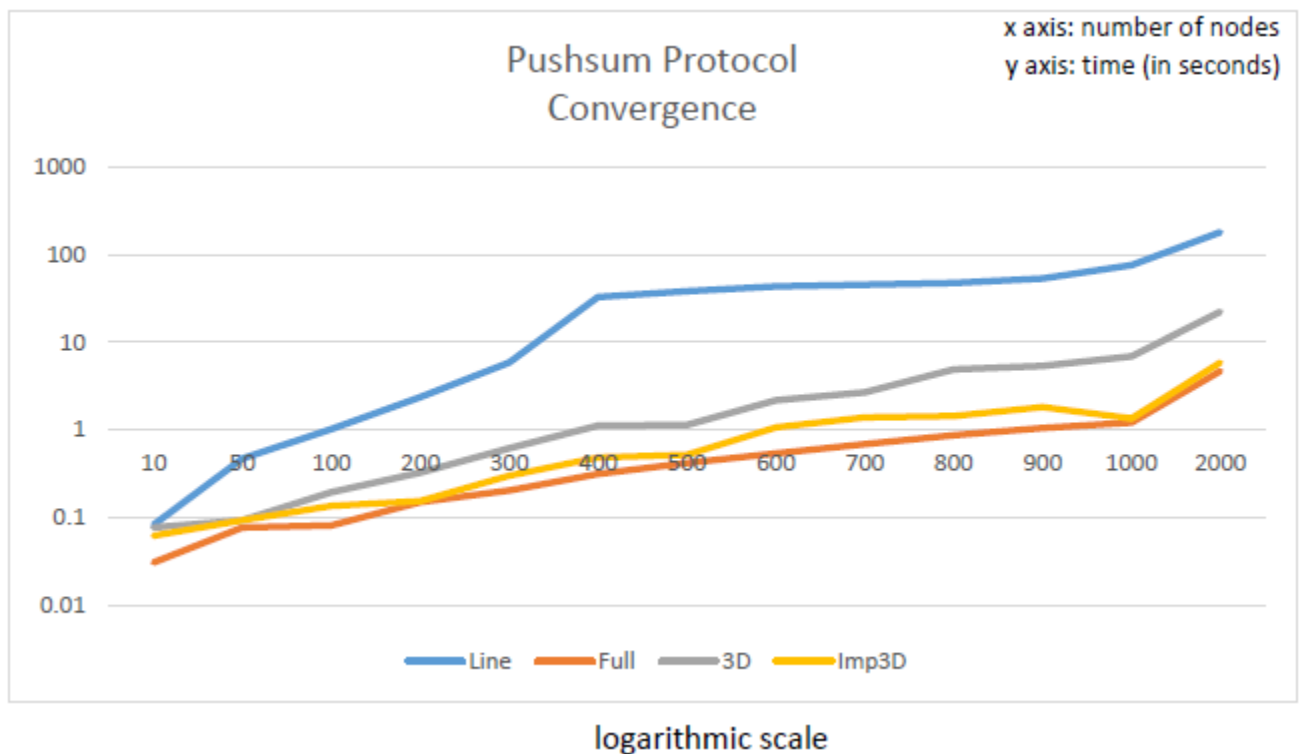
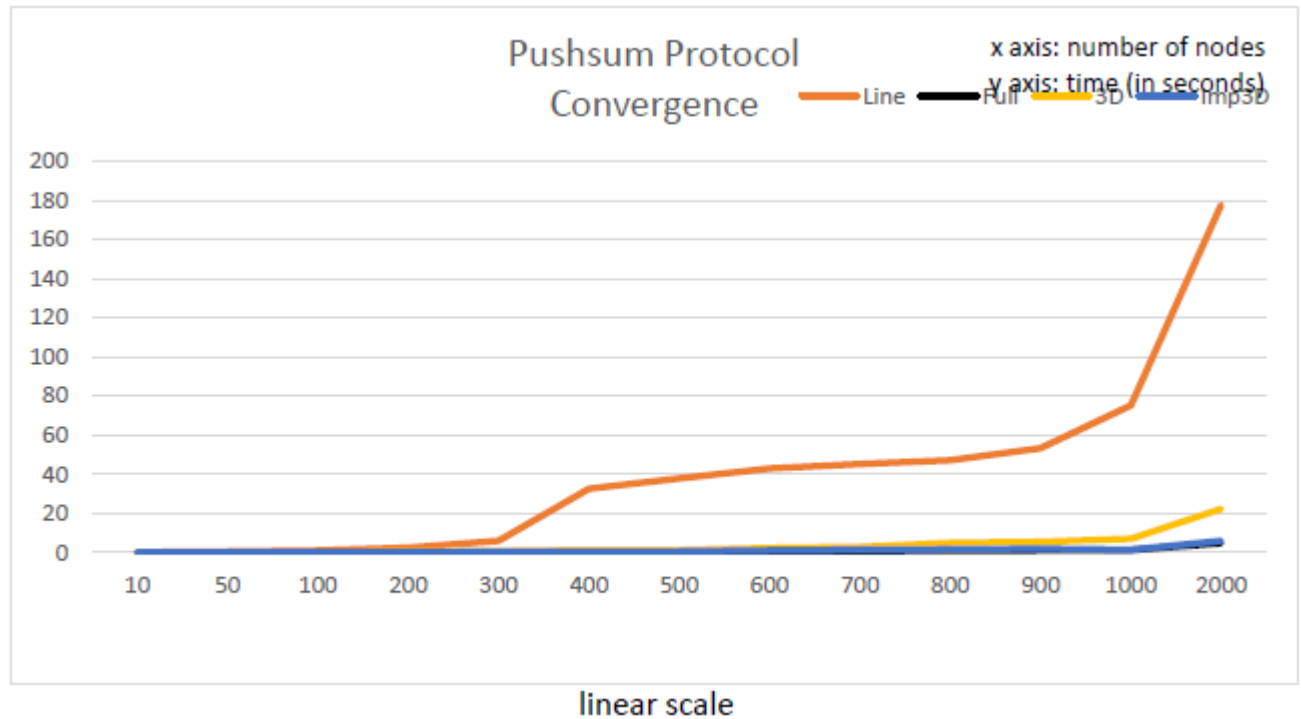
3D Grid: The convergence for 3D Grid is almost comparable to Imperfect 3D Grid for an input size of 10000 nodes.

Imperfect 3D Grid: This topology has the best convergence among the available topologies. Thus, we can conclude that the Imperfect 3D Grid takes the lowest convergence time whereas full topology has the highest convergence time for Gossip algorithm.

An **interesting observation** is that the convergence time for line topology is taking the least whereas it should have been the highest (by theory). However, after digging deeper into this curious anomaly on our implementation, we think this might be the case because our line topology selects its immediate neighbor i.e. if it is an array, it would transmit the message either to its neighbor on the left or on the right. Again, since this is random and it can choose only from a maximum of two nodes, there is a high probability that its neighbor retransmits it to the current node rather than the other neighbor. There is a high chance that this might

continue 10 successive times thus achieving convergence with or without sending message to all the nodes.

PUSHSUM ALGORITHM



Tabular Data

No. of Nodes	Line	Full	3D	Imp3D
10	0.084	0.031	0.078	0.063
50	0.472	0.078	0.094	0.094
100	1.027	0.082	0.197	0.137
200	2.366	0.151	0.328	0.156
300	5.881	0.205	0.619	0.302
400	32.523	0.315	1.127	0.489
500	37.92	0.422	1.141	0.524
600	42.956	0.543	2.191	1.075
700	45.229	0.691	2.69	1.391
800	47.121	0.876	4.89	1.44
900	53.244	1.056	5.38	1.823
1000	75.191	1.214	6.88	1.356
2000	177.711	4.67	22.191	5.889

The above graphs represent the convergence for the PushSum algorithm for the following topologies: Line, Full, 3D Grid, Imperfect 3D grid. (Note: The first graph is on a linear scale while the second graph is on a logarithmic scale for feasibility).

Observations and Findings:

Full Topology: The full network takes the lowest time to converge in the case of pushsum protocol

Line Topology: As can be observed, the line topology takes maximum time to achieve convergence for the pushsum algorithm.

3D Grid: The 3D grid is only faster than the line topology in terms of the amount of time taken to converge.

Imperfect 3D Grid: It is only slightly slower than the full topology for an input size of 2000 nodes.