Project Report
Implementation of Gossip Protocol

Team Members -

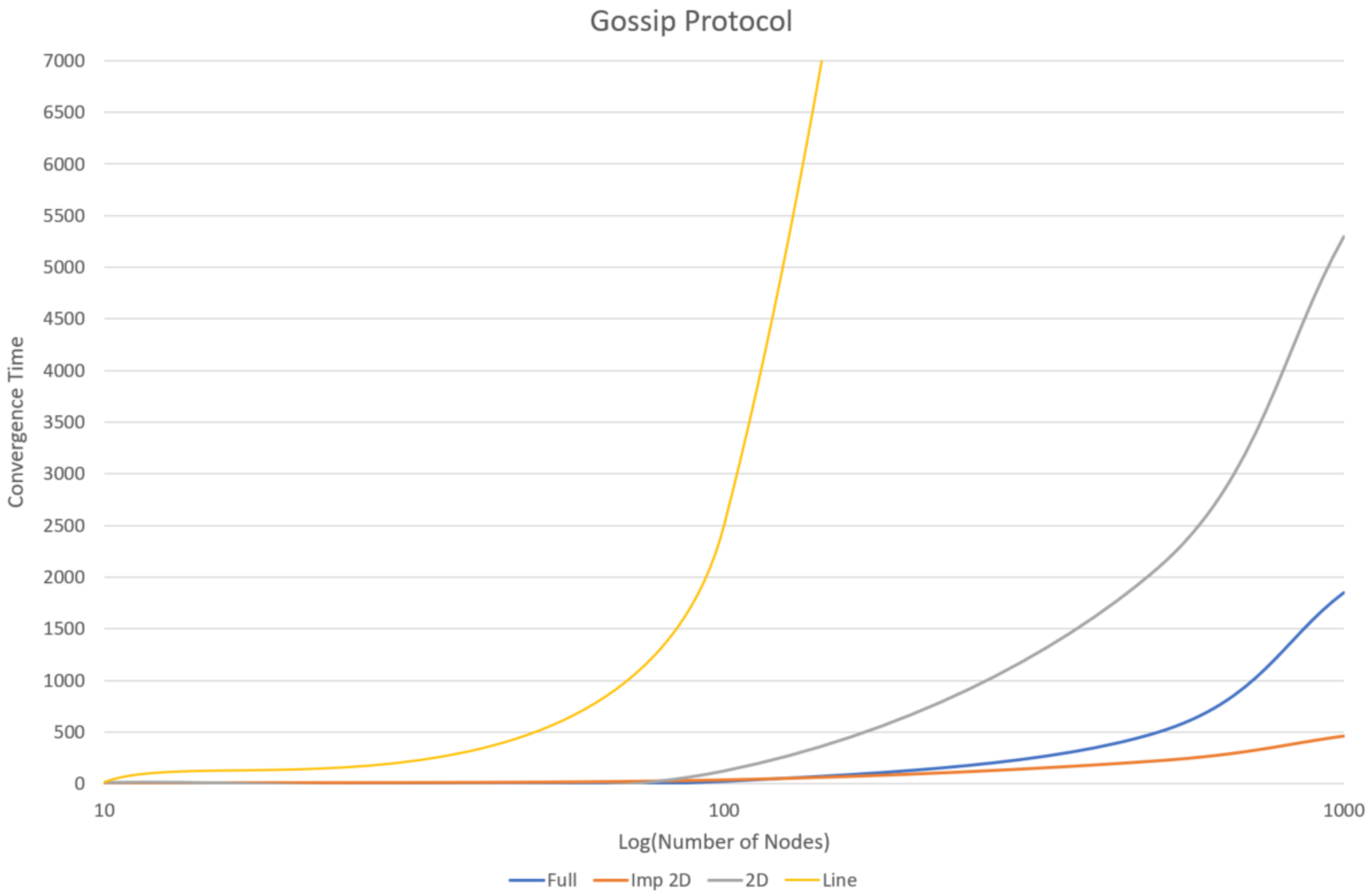
1. Kanika Gupta, UFID - 96977046, kanikagupta@ufl.edu
2. Nikhil Chopra, UFID - 98973831, nikhilchopra60@ufl.edu

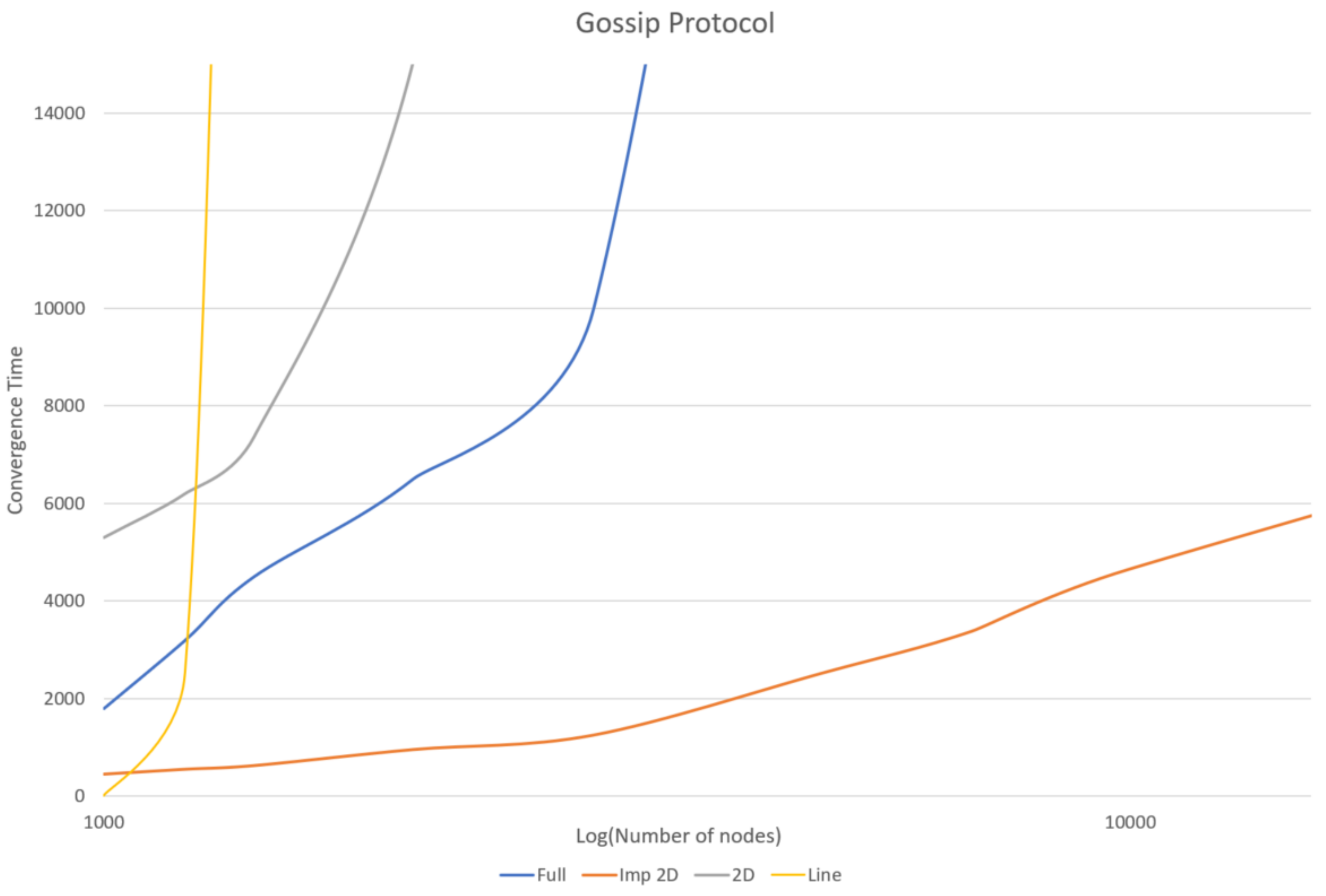
Implementation Details -

We have assumed that convergence of Gossip Algorithm occurs when All the nodes in the network have heard the rumour at least once. In our implementation, when a node hears the rumour for the first time, it marks itself as converged but keeps sending and receiving rumours. When a node hears the rumour for the 10th time, it stops transmitting the rumour further. When convergence is achieved, the program prints the convergence time and terminates.

For Push-sum algorithm, we have assumed that convergence of a node happens when it its s/w ratio (Average Estimate) has not changed more than a factor of 10^-10 in three consecutive message receive rounds. We terminate the program after all the nodes have achieved convergence and then calculated the total time taken to converge the algorithm. We do this because on a large network, if we terminate the program after the convergence of very first node then the average estimate of the other nodes in the network will differ from that of the converged node. So, to provide more fault tolerance we have terminated the program after the convergence of all the nodes in the network.

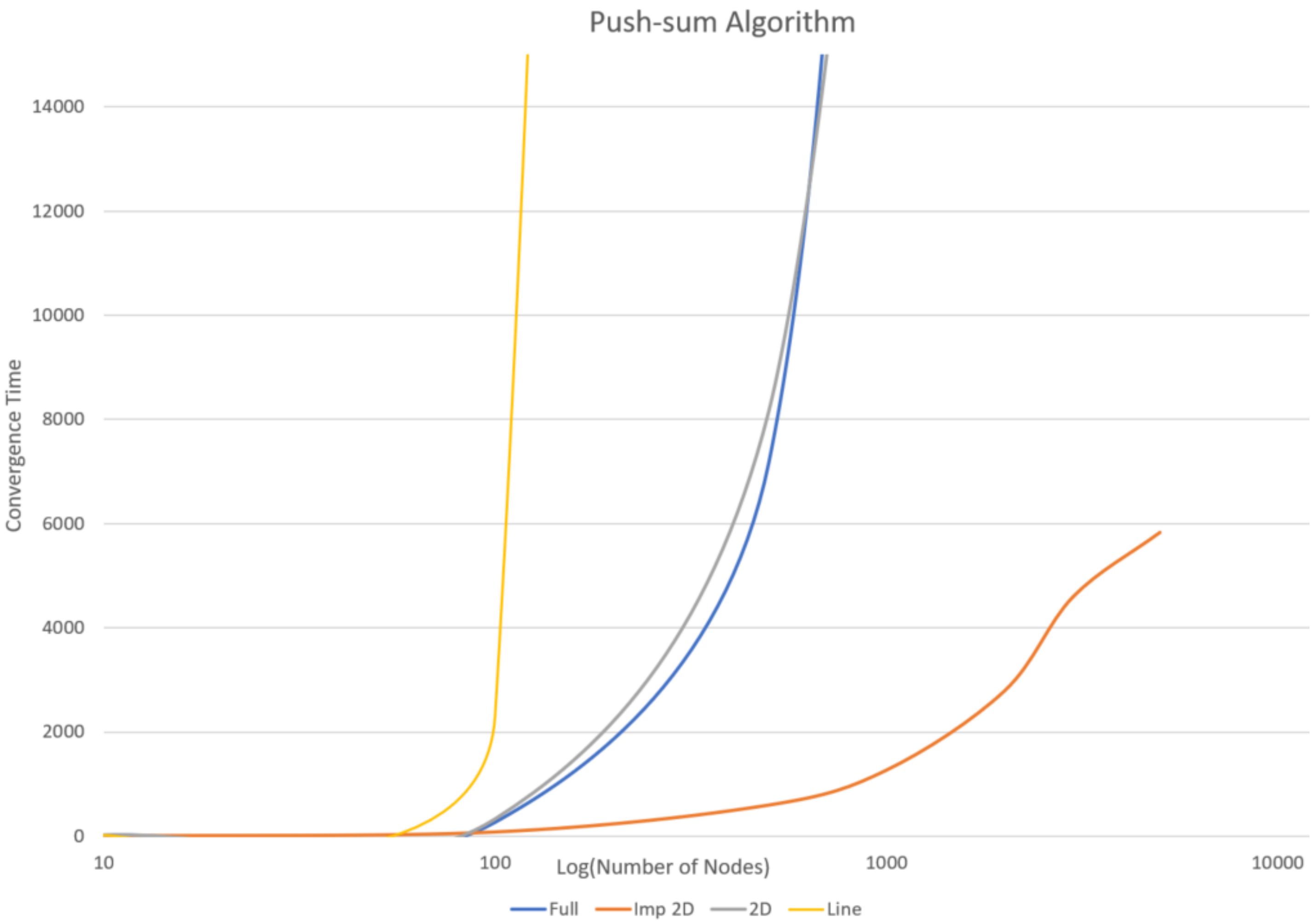
Graphs -


Gossip Protocol
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
Convergence Time
10
100
1000
Log(Number of Nodes)
Full
Imp 2D
2D
Line


Gossip Protocol
14000
12000
10000
8000
6000
4000
2000
0
Convergence Time
1000
10000
Log(Number of nodes)
Full
Imp 2D
2D
Line

This graph plots the convergence time of different topologies vs the size of the network for Gossip protocol. Inference from the graph –

1. The line topology has the largest convergence time out of all.
2. Full topology has least convergence time for smaller networks of 100 nodes but for larger networks its convergence time increases exponentially due to large memory occupied by the adjacent node lists for every node. Surprisingly, Improper 2D gives better performance for large number of nodes because size of adjacent list remains small for every node and we are even considering randomness factor in this.



This graph plots the convergence time of different topologies vs the size of the network for Push-sum algorithm. Inference from the graph -

1. The line topology has the largest convergence time out of all.
2. Full topology has least convergence time for smaller networks of 100 nodes but for larger networks its convergence time increases exponentially due large adjacency list for every node. Again, Improper 2D wins in the case of large number of nodes.

Interesting findings

Convergence time for both algorithms in the case of line, 2D and Full Topology increases exponentially whereas in the case of improper 2D the convergence time increases gradually as the number of nodes increase. This behaviour was completely opposite to what we thought as Full topology is choosing random nodes and sending them the messages but here memory constraints come into picture in the case of large network. Whereas in the case of Imp 2D which is a perfect blend of randomization with 2D network, the adjacency list of each node just has 5 nodes which does not pose any memory constraints, making imperfect 2D the fastest in the case of large networks.