Project Report

Dhruv Mahajan, UFID: 42111994
Ashvini Patel, UFID: 47949297

1) Gossip

Implementation Details & Findings:

- **Time:** As mentioned in the documentation, we do not measure the time to spawn the network. Spawning the network is a very time-consuming step. To measure convergence time, **System.system_time(:milliseconds).** The one mentioned in the document is a Java function.

- **Entry point:** We pick a random node as the entry point for all our networks/topologies. The entry point selection is not mentioned in the documentation and this seemed like the best choice to improve performance.

- **Convergence:** For gossip based networks, the purpose of the network is information dissemination. Hence, once every node in the network has received the message/rumor once, we stop the network. **Note:** Individual nodes do not stop transmitting till they hear the message 10 times but the whole network terminates as soon as each node hears the message at least once.

- Listen to the rumor once and keep gossiping/transmitting strategy.

- The Convergence time had the following order (increasing) : **full < imp2D < 2D < line**

All the time values in the table and graph are the middle values of 3 trial runs. This is a good representative of the time values observed.

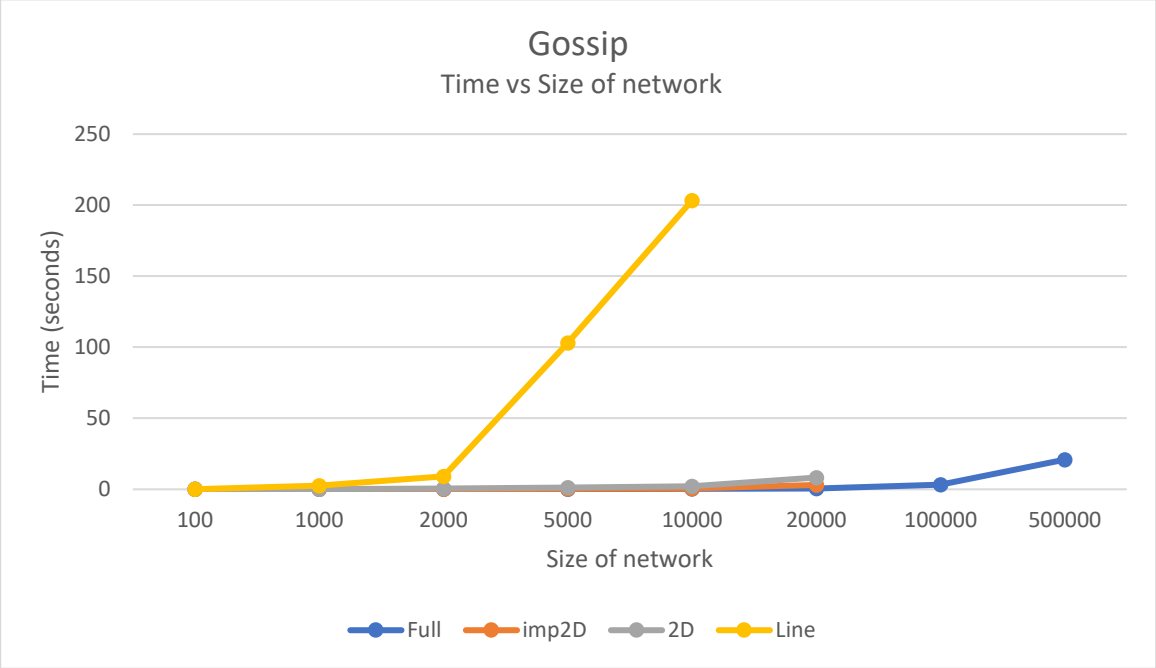
Justification: The other option was taking the mean for the 3 values, but since the mean value was not actually observed, it can't be reported as an observed value.

Table:

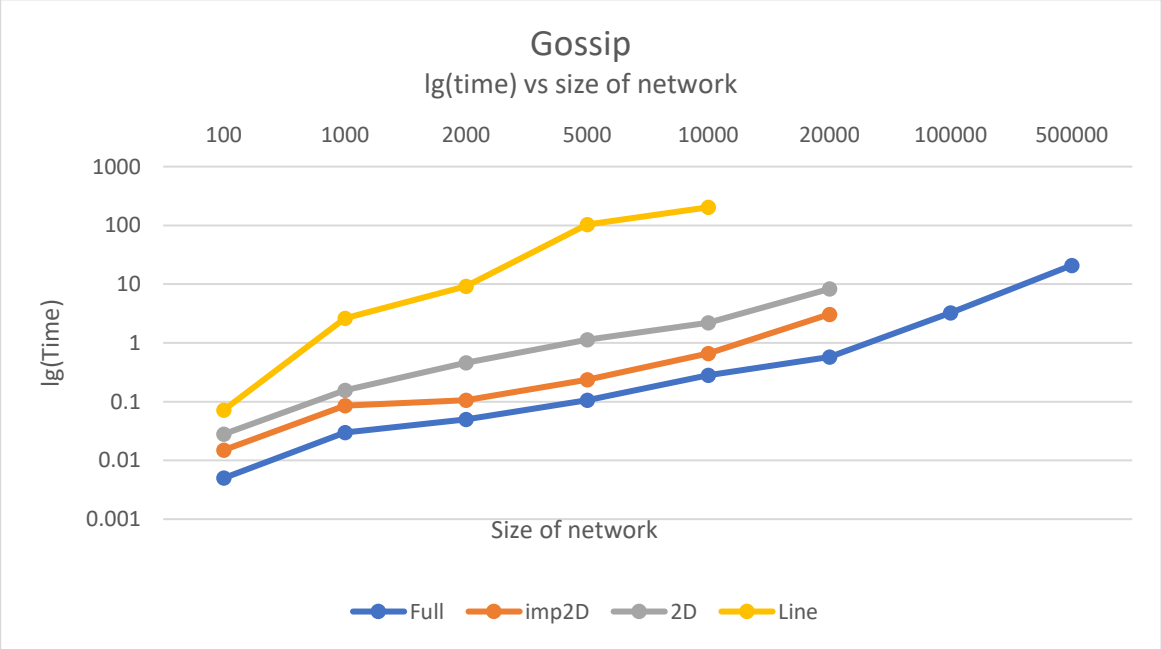
Nodes	Full	imp2D	2D	Line
100	0.005	0.015	0.028	0.072
1000	0.03	0.085	0.157	2.637
2000	0.05	0.106	0.458	9.168
5000	0.106	0.236	1.139	103.112
10000	0.282	0.658	2.196	203.23
20000	0.577	3.068	8.277	
100000	3.251			
500000	20.866			

Table with time in Seconds

Graph:



Time in Seconds vs. No. of nodes



Log (Time in Seconds) **vs. No. of nodes**

Note: For Line and 2D, the algorithm does not converge reliability and the gossip can get stuck halfway and then the execution may not end.

2) Push-sum

Implementation Details & Findings:

- **Computed Average:** Setting all weights to 1 computes average. Project description asks us to set all weights to 1. To compute sum, all weights should be zero, only weight of the starting node should be 1.
- **Random walks:** As mentioned in the Project description, we implemented a single random walk. The walk terminates at random node which then reports it's s/w ratio. We check that this is the correct s/w ratio. IE s/w = Avg.
- **Termination condition**: *If an actor's ratio s/w did not change more than 10 ^ −10* ***in 3 consecutive rounds*** *the actor terminates.* The parameter 3 produces extremely accurate results for fully connected topologies. But for other topologies, this needs to be tuned. We noticed much more **accurate results for line and 2D with the value 5** instead of 3. But the graph is based on the value 3, as specified in the project document.
- **Entry point:** We pick a random node as the entry point for all our networks/topologies. The entry point selection is not mentioned in the documentation and this seemed like the best choice to improve performance.
- The Convergence time had the following order (increasing) : **full ==~ imp2D < 2D < line**

All the time values in the table and graph are the middle values of 3 trial runs. This is a good representative of the time values observed.

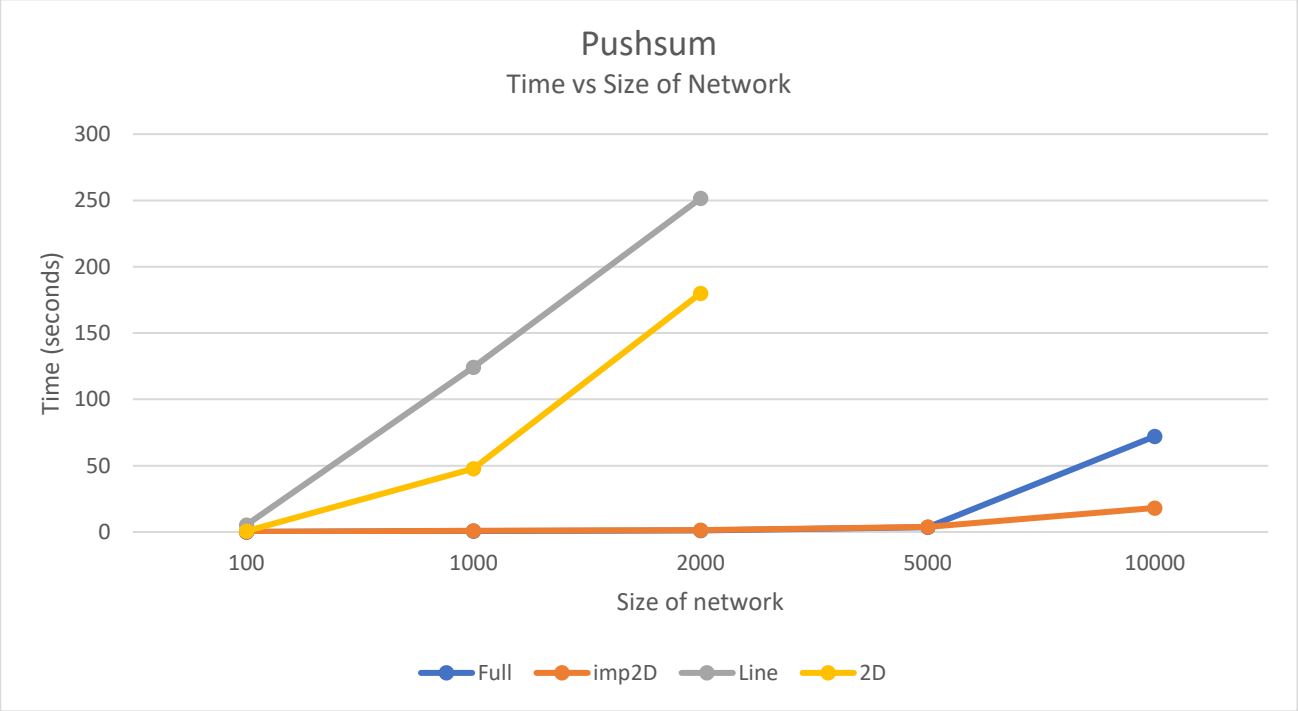
Justification: The other option was taking the mean for the 3 values, but since the mean value was not actually observed, it can't be reported as an observed value.

Table:

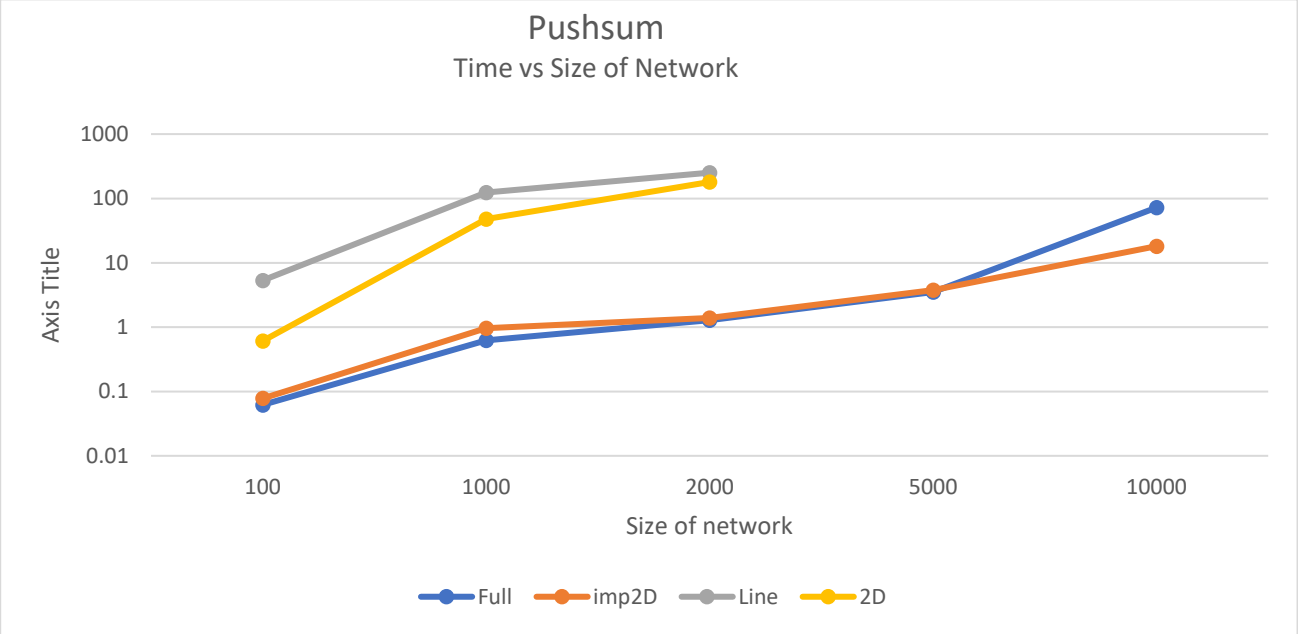
Nodes	Full	imp2D	Line	2D
100	0.062	0.078	5.328	0.61
1000	0.625	0.968	124.11	47.625
2000	1.275	1.391	251.515	179.937
5000	3.5	3.781		
10000	72.04	18.11		

Table with time in Seconds

Graph:



Time in Seconds vs. No. of nodes



Log (Time in Seconds) **vs. No. of nodes**