COP5612 – Fall 2020
Project 2 - Gossip Simulator

Kanika Sharma, UFID: 7119-1343

Instructions to run the program :

1. Unzip project2.zip
2. Ensure project2.fsx is included in the compile in .fsproj file
3. Through the command line, run the program using the command :

   dotnet fsi --langversion:preview project2.fsx numNodes topology algorithm

   (providing different values for numNodes, topology and algorithm)

   topology can be full, line, 2D or imp2D

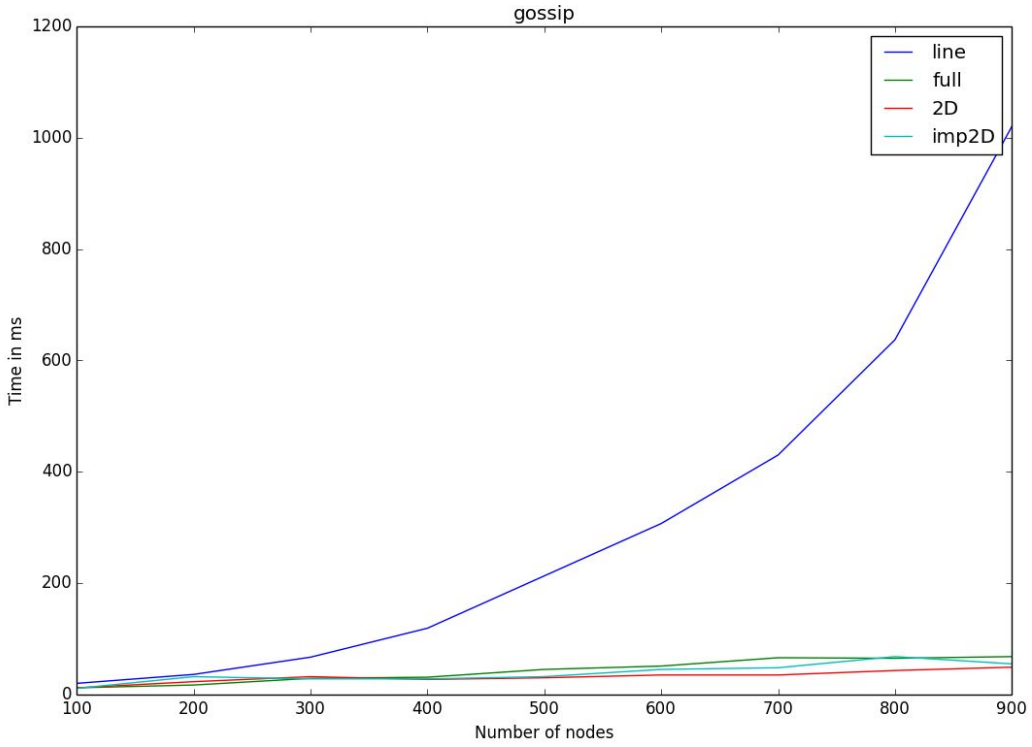
   algorithm can be gossip or push-sum

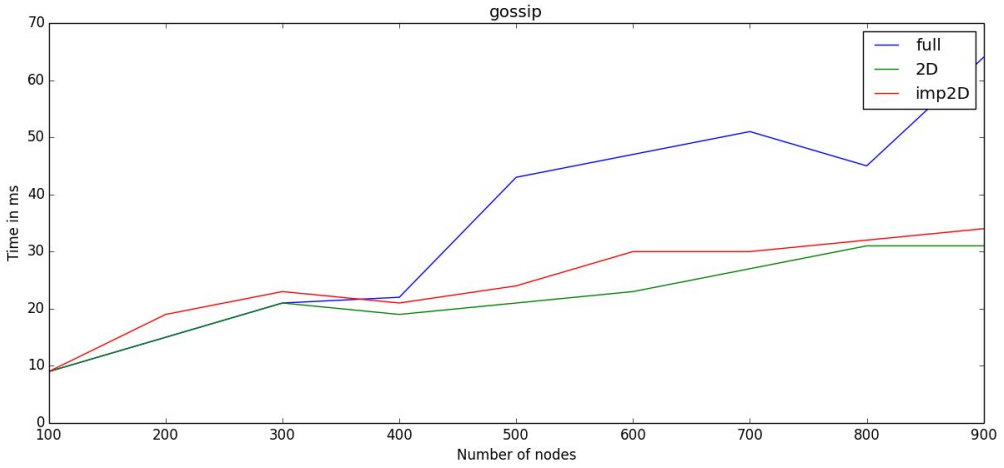
Dependency of convergence time as a function of the size of the network:

Graphs are plotted (Number of nodes vs Time in milliseconds) by considering nodes in the range 0 to 1000 with unit 100. We used a 1.4GHz quad-core Intel Core i5 processor, 8GB RAM computer to run the simulator.

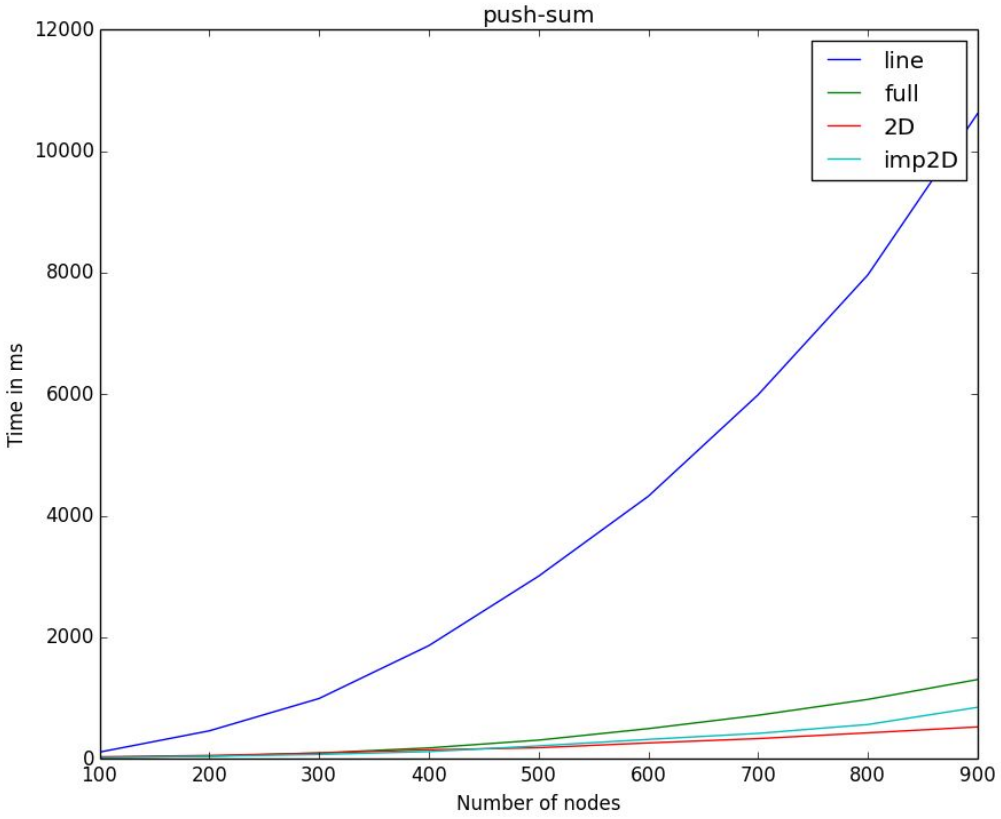
For each algorithm we have kept two graphs, one with all the topologies in and another with only full, 2D and imp2D to distinguish between these properly. The zip file contains a graph.py file which can be used to generate these graphs using the command :
python graph.py

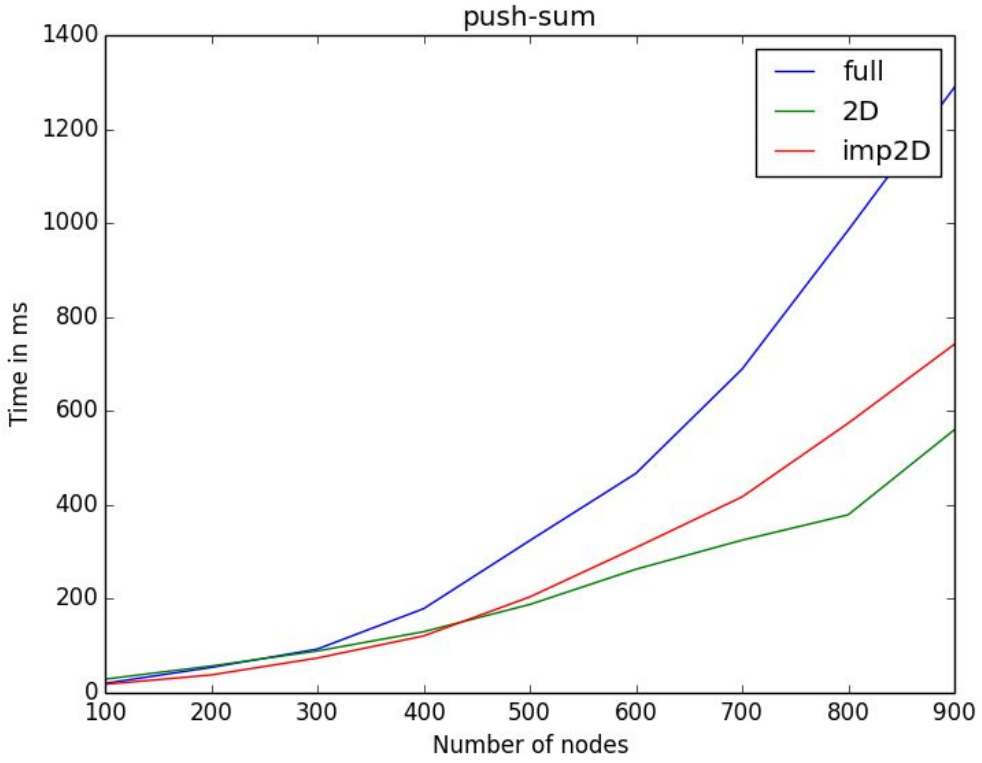
Gossip protocol :





Push sum protocol :





Important Findings :

1. The Line topology was not initially converge for Gossip algorithm since each node in line topology will have only 2 neighbours (except for 1st and last node which will have only one neighbour) and there came a situation where a node gets a rumour but does not have alive neighbours who can transmit the rumour. In this case, the line topology will not converge and keeps on running. To avoid this scenario, we have added a special case for line topology that when it gets stuck this way, a random node from the network is chosen to transmit the rumour and the process continues.

2. Line topology is found to give the worst performance in both algorithms and from the graph we can see that it shows an exponential increase in convergence time as the number of nodes increases.

3. From the graph we can see that Line topology is the slowest to converge for both the algorithms. Full topology is the fastest to converge for smaller values of number of nodes and 2D works fastest when number of nodes are higher. For smaller values of number of nodes, full works faster than 2D. 2D topology works faster than full topology for higher values of number of nodes because 2D has only 4 neighbours irrespective of the number of nodes but for full topology, as the number of nodes increases the number of neighbours also increases for each node. So here we see a tradeoff that when the number of nodes increases, full topology will become slower and 2D topology will work faster.

4. By doing a comparison of both the graphs, we can conclude that time taken to converge the algorithm for all topologies is more for push sum than gossip.