COP5615 Project 2 -Gossip Simulator

Byneedi Deepthi
UFID:36871955

Sanjay Reddy Banda
UFID:58782239

Implementation Details:

Gossip:

We assumed that the convergence of gossip algorithms occurs when all the nodes have heard rumor 10 times or if the node has already sent the rumor across to its neighbors 1000 times. In the way we implemented the algorithm, when a node sees that, it has heard the rumor more than 10 times, it announces itself as converged and apart from that it also converges when it has sent the rumor across to more than 1000 nodes. After this, when a node sees that all its neighbors have converged, it declares itself converged as no further flow of rumor can be expected at this point.

Push-sum:

The convergence of push-sum has been implemented by assuming that directly when a node's ratio has not changed more than 10^ -10 in three of its consecutive rounds. Termination of this algorithm happens when all the nodes have achieved convergence.

For the Push-Sum algorithm, If we implement it the way it is defined in the document only one Node will converge and halt the network and all other nodes will not be converged. To address this issue we have implemented in 2 ways:

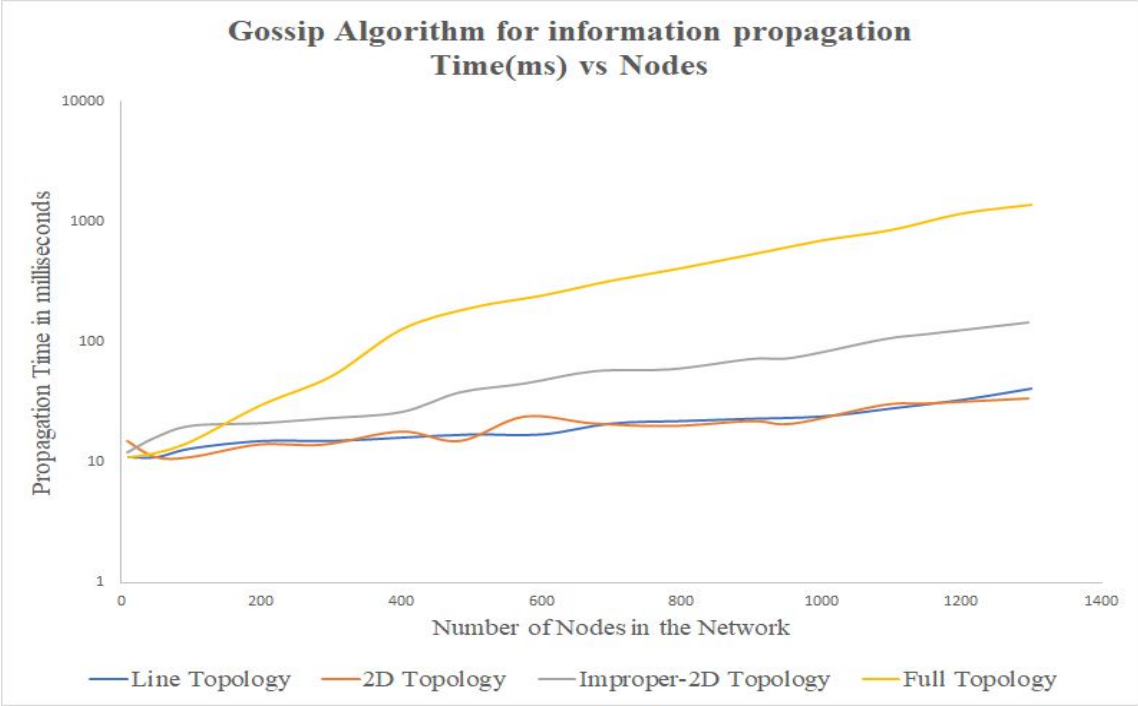
1) Whenever a converged node receives a message it passes the same s and w values to its neighbours but it doesn't accumulate received s and w values to it. If all neighbours of a Node are converged then itself declares as converged.In this way we can make sure all nodes are converged.

2) When a Node is about to converge it will broadcast its s and w to all its neighbours that aren't converged. By this way we can say all the nodes are converged. And we also made sure that a converged node will never get a message.

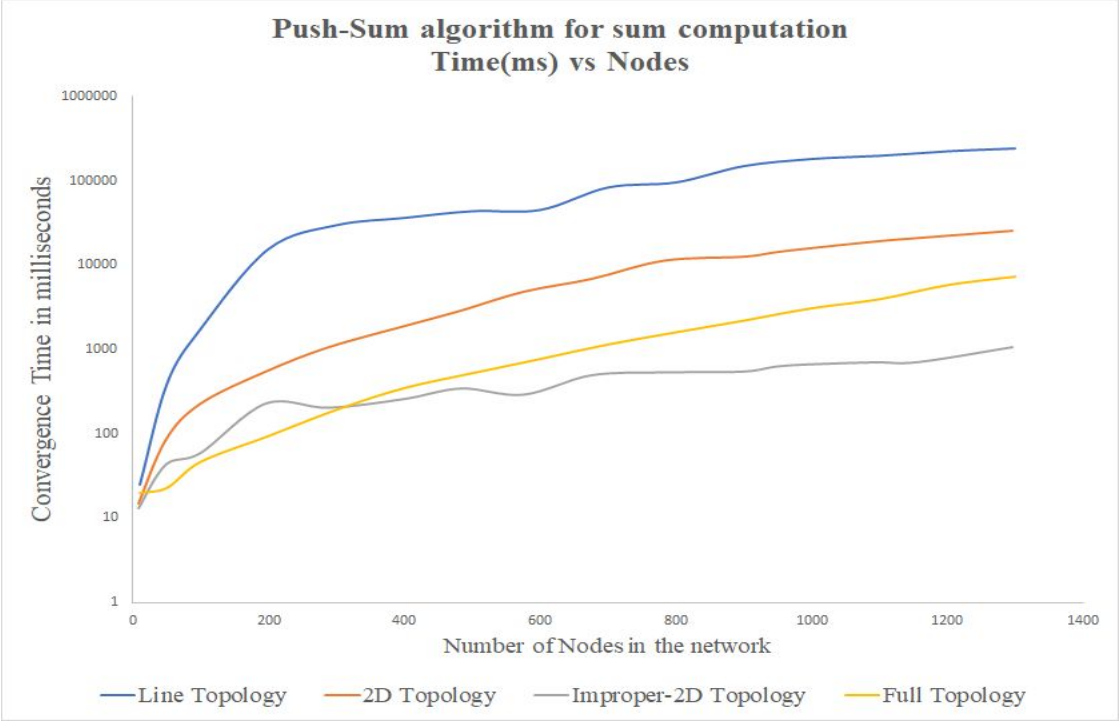
Out of above 2 implementations 1st implementation gave better results in terms of runtime and s/w ratio as well. 1st implementation is highly scalable with respective to convergence time compared to 2nd Implementation.

Results:

Gossip Algorithm:



Push-Sum:



The above graphs depict the convergence time on the Y-axis vs the number of nodes on the X-axis.

Interesting Findings:

Gossip Algorithm:

We were expecting the propagation time as “Full<Imp2D<2D< Line” because after the rumor has started at time t if n nodes had the rumor then there can be utmost n+2 nodes with rumor at time t+1 in case of Line topology, 2n nodes with the rumor at time t+1 in case of Full topology and 2D and Imp2D topology will fall in between and Imp2D has little edge over 2D. But we have achieved the following “Line~2D<Imp2D<Full” but the reason for getting these results are as we are simulating more nodes on a quad core machine where we cannot guarantee that all the actors are getting equal amount of execution time and cannot guarantee the parallelism of all node.
We also observed that as Number of Nodes increases, the propagation time is increasing exponentially.

Push Sum:

Here we got results about what we have expected. The reason why we are getting the expected results is that here we can guarantee that all the nodes are getting an equal amount of computational time because at any point of time only a few nodes perform the actions.