Project 2 - Gossip Simulator

Group Members

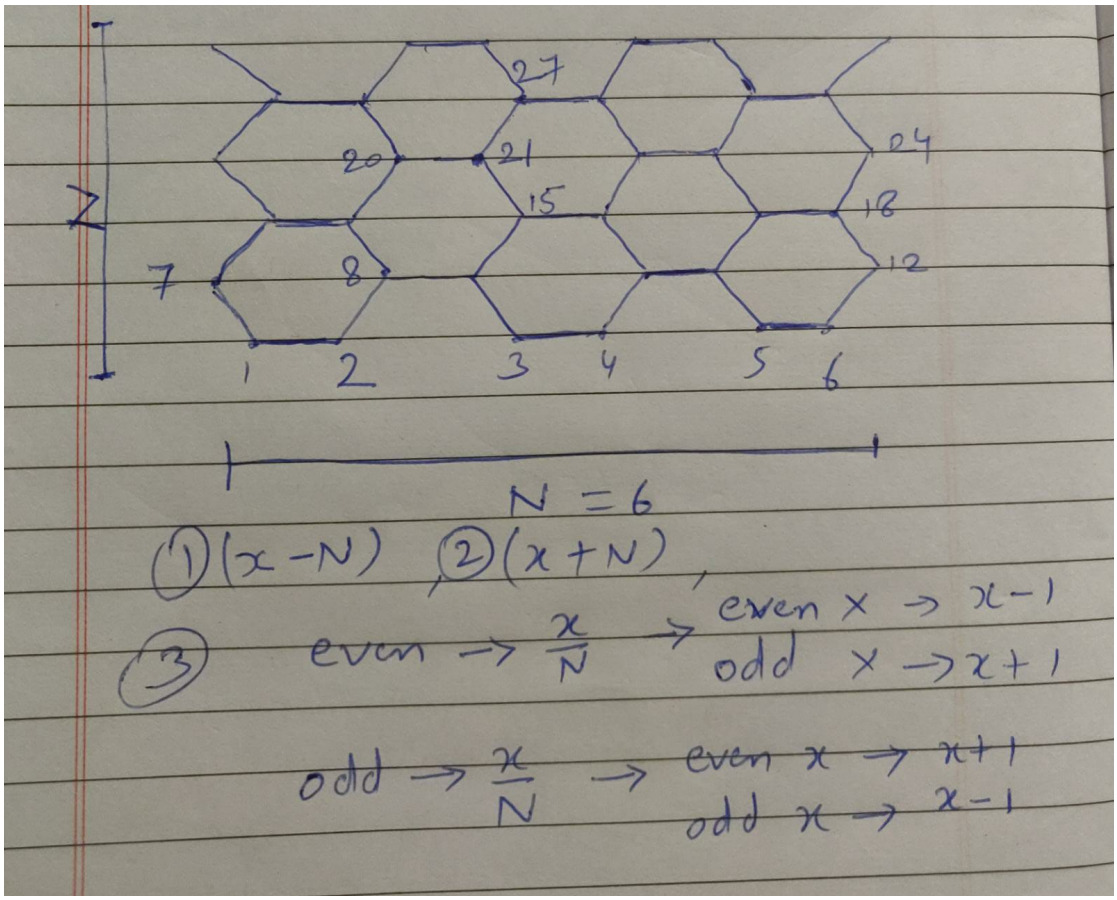
Abhishek Jaimini - 7365-2858

Yash Kandalam - 3515-4584

Topologies

The steps to run can be found in Readme file in the same folder.
All 6 topologies have been implemented for both Gossip and PushSum algorithm. Here, we are describing our approach for Honeycomb and Rand2d topologies as others are self explanatory .

1. Honeycomb -
   The Algorithm that we have used for finding neighbors in honeycomb is that first draw a grid closest to given input dimensions. Number of nodes in each row/ column is N (6 in given case).
   If the row in which element resides is even or odd, based on that the third neighbor is found which also depends
   Two remaining neighbors are simply found by adding and subtracting N from element.
   In given case 21 resides in row 3, (rows starting from 0). The quotient of 21/N is also odd which gives formula for finding third neighbor as x-1.

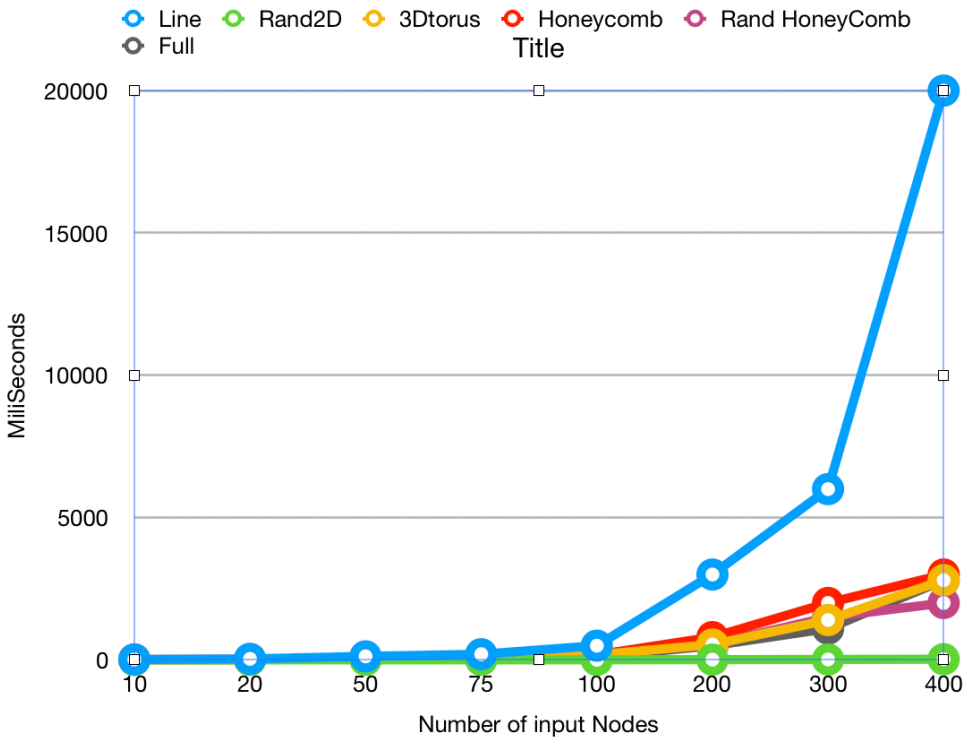


HONEYCOMB - NEIGHBOR SEARCH ALSO

2. Random 2D grid -
   For random2d, We have placed the nodes on a 2d grid. For any given node, we find its distance to other nodes. If distance of a node comes to be less than (ElementsInEachRow/10) then we can consider it to be the neighbors. As we have to find the nodes within 0.1 distance of a given node on a 1x1 square, we take the 0.1 -> N/10 as respective ratio for an NxN square. Distance is calculated using formula-
   D = sqrt ((x1 - x2)^2 + (y1-y2)^2).

Gossip Algorithm

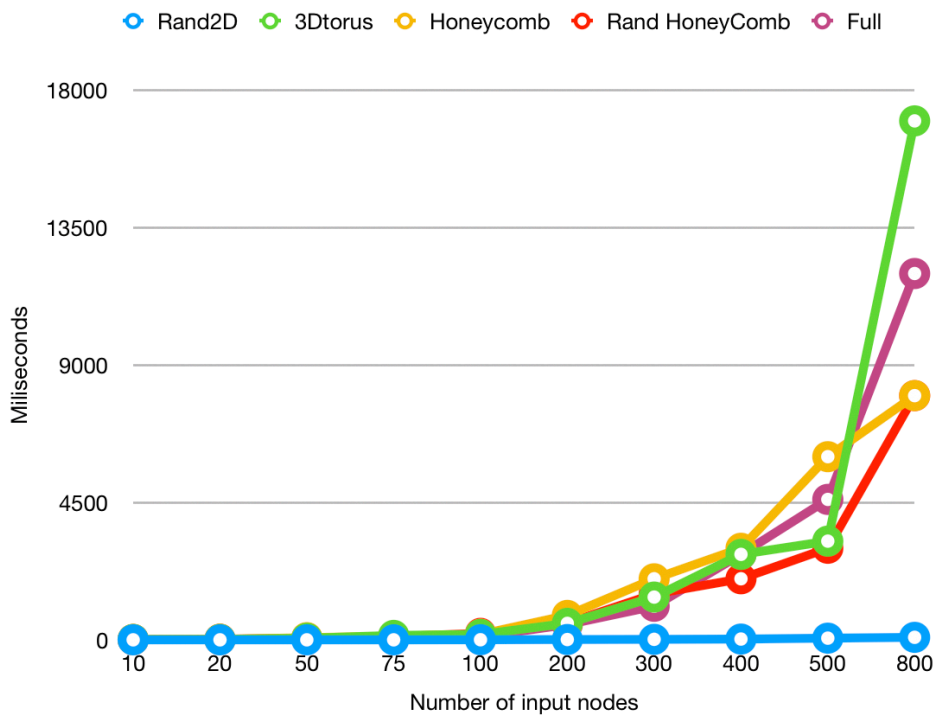
For gossip, we collected the convergence time data for all the topologies for different inputs and plotted a graph.



GOSSIP - ALL TOPOLOGIES

Finding - Line topology seems to be giving convergence with highest time compared to other topologies. One reason for this can be that in Line topology, nodes have just two neighbors to gossip to on a linear scale, which will take longer time for message to converge.

If we remove line topology from graph, we got the following result-



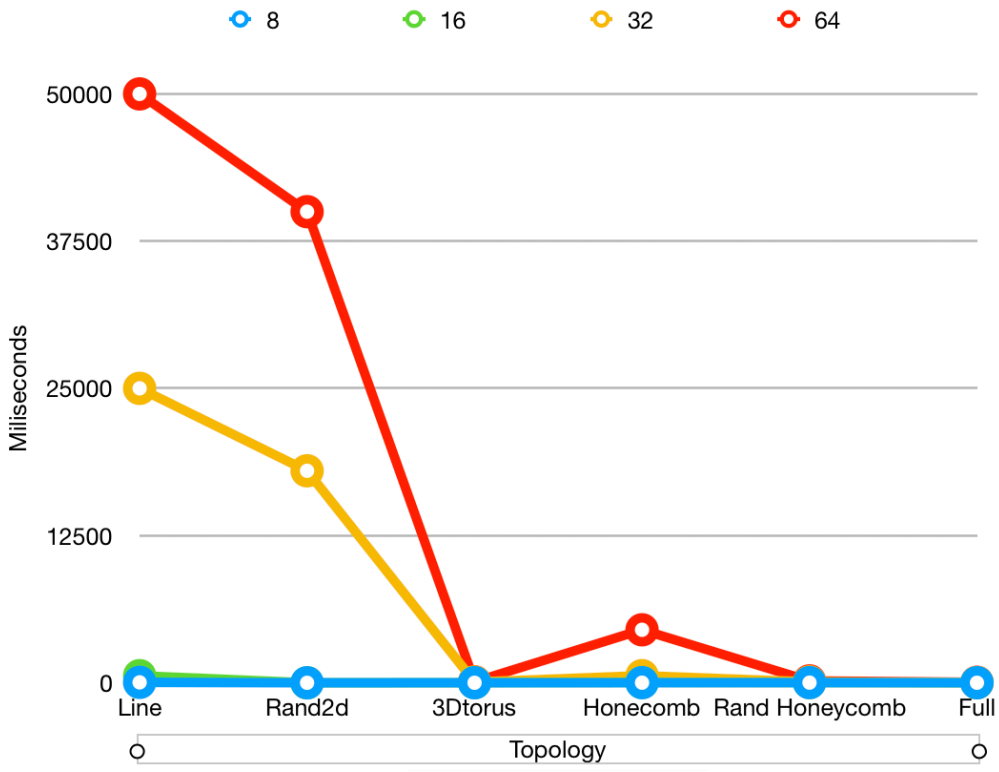
GOSSIP - ALL TOPOLOGIES EXCEPT LINE

By looking at our data and graph, it appears that for larger input sizes, Rand2d works better than others. One possible reason might be, when placed on a grid, the neighbors in 1/10 of side distance might be considerably higher than other topologies.

One interesting thing we observed about Rand2d was that for small inputs (say less than 100), convergence does not occur. Because less than 10 nodes are placed on 1 length square, it is hard to get neighbors within 0.1 distance.

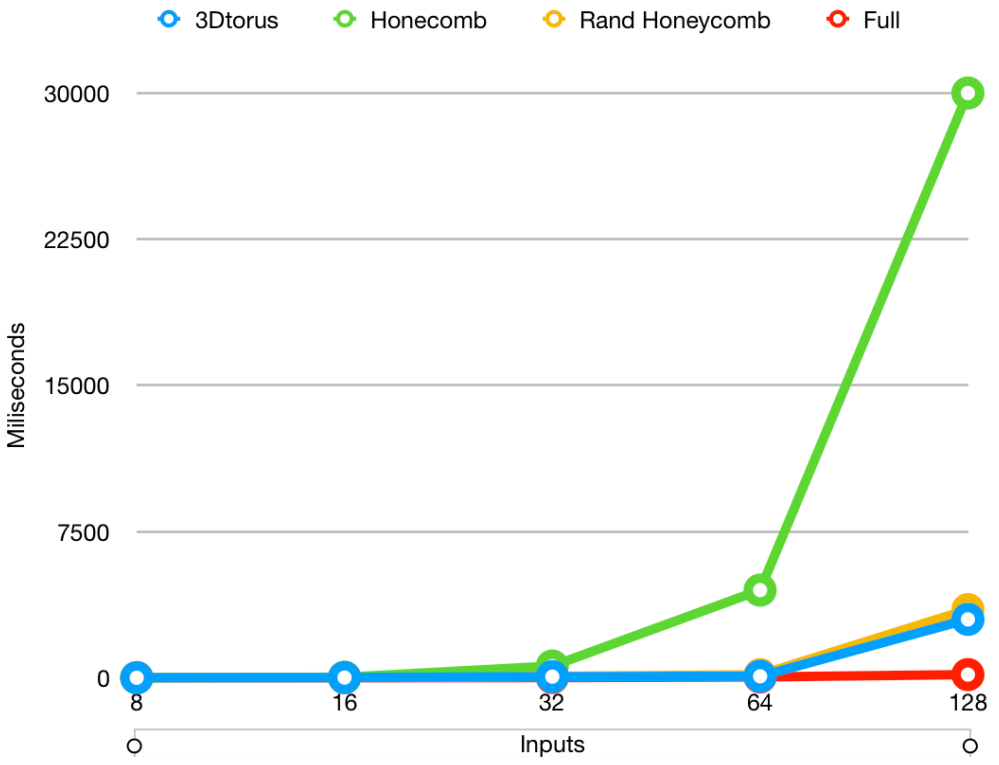
Push Sum Algorithm

We collected convergence time for all the topologies for push sum algorithm and plotted on a graph. As expected, line had the worst convergence time.



PUSHSUM - ALL TOPOLOGIES

Pushsum performance was worse than gossip in our analysis as weight ratios took longer time



TYPE TO ENTER A CAPTION.

to converge.
Then we plotted topologies except line. Honeycomb showed comparatively higher convergence time than others except line. Lowest convergence time was shown by Full topology in our analysis.