Project-2

Includes both regular and bonus portions

Submission contains a sbt folder containing the source code and all the files required to build the code locally.

folders---Gossip, Gossip-failure

To compile ---- sbt compile

To run ---- sbt "run <num nodes> <topology> <algorithm>"

The following arguments are allowed--

topology--- line,full,3D,imp3D

algorithm--- gossip,push-sum

In my implementation as per the documents as soon as the the actor satisfies the termination condition the system is terminated, this happens as there is only one message floating around the network, as soon as it hits an actor, which has achieved the termination condition the packet is not forwarded as per specifications and is consumed by the network, resulting in termination of the system. Its like a ball being passed around and one someone has had the ball for x number of time he runs away with it.

Team Members-- Saumitra Aditya UFID # 51840391

What is working-- I have tested all combinations of topologies and algorithms, they seemed to work.

Topology-- for 3D topology I start from (1,1,1) and each plane contains square(cubeRoot(numNodes)) sensors. I have verified the neighbors and print them on console for clarity.-- for example in a network with 27 sensors plane 1 will contain the below --

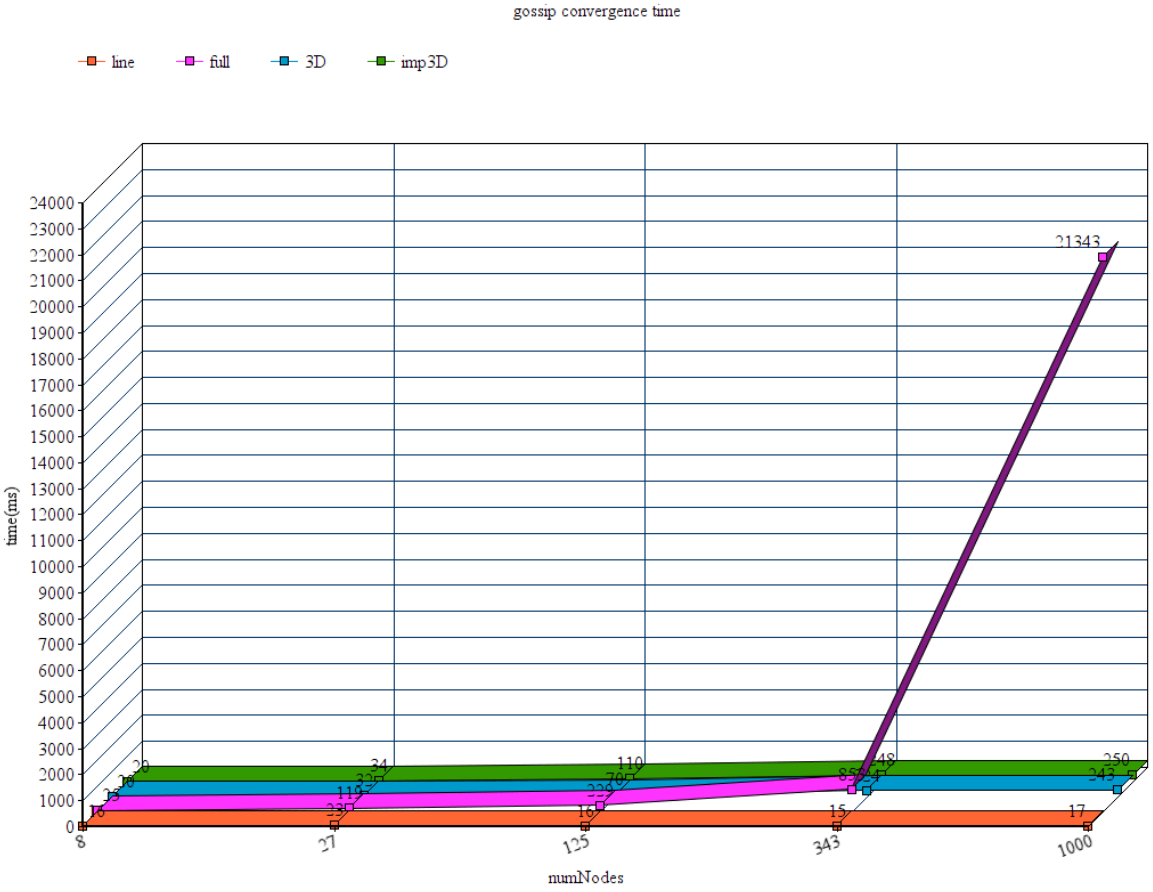
7	**8**	**9**
4	5	6
1	2	3

plane 2 will have

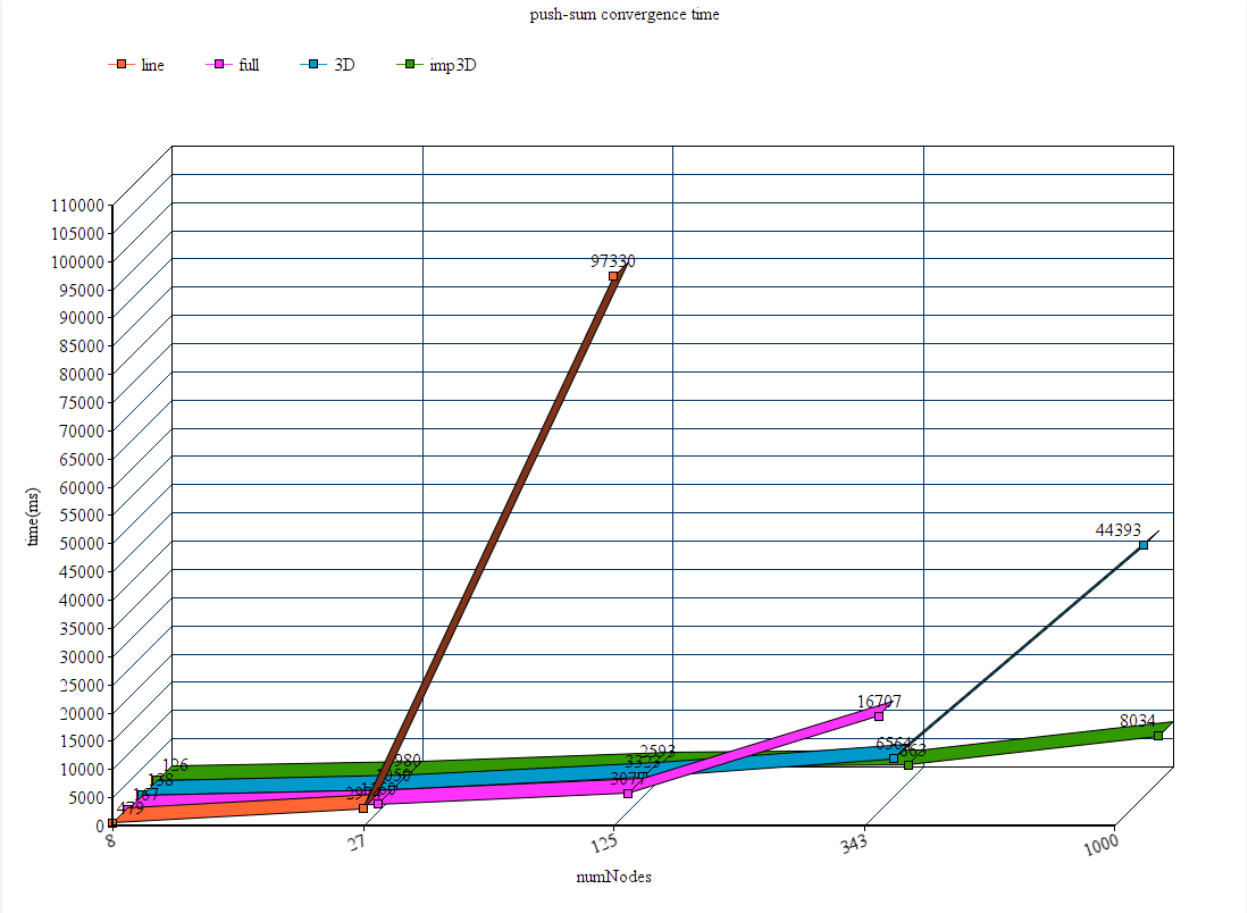
16	**17**	**18**
13	14	15
10	11	12

and node 10 will be neighbors with (13,11,9,19).

The graphs below capture the details about the experiment--with gossip I was able to go up to 1000 nodes with all topologies , but with push-sum only 3D and imp3D were able to converge for 1000 nodes.



I will include the images for the graphs in the submission zip file for clarity. As far as gossip is concerned Line topology emerges as the winner with -- This is because the termination condition is not when all nodes have had the message 10 times but any one of them has had it for 10 times. In this topology the message passing is extremely localized -- as every node has at-most two neighbors--as you can observe irrespective of numNodes line topology has almost constant performance. The same logic can be used to explain the behavior of 3D topology, where neighbor set is fairly constrained, notice that performance of imp3D is marginally poor than 3D which can be attributed to that extra random neighbor which might lead the message away. Full topology has the poorest performance as the message is tossed around the whole network.



for push-sum I noticed irrespective of the topology the estimate-sum is always around the same value, only the time taken to converge varies. Here the performance of line topology is abysmal because the message is always trapped between a small subset of nodes in the network. Imp3D has the best convergence times-with its combination of order and randomness it seems to match push-sum better, 3D comes in second. I think it has to do more with how the values for (s,w) were assigned to the nodes.

Failure Modelling

My implementation to capture node failures follow the below algorithm--

After every interval of <parameterized> milliseconds a message is sent to the Monitor who keeps an eye on the sensor networks to introduce chaos in the system i.e.

Upon receipt of the chaos message -- following steps are followed

/*Simulate failures in the network

* 1. Identify a random sensor from the network.

* 2. identify the neighbours of this sensor

* 3. Inform the neighbours to remove this node from their neighbour_list

* 4. wait for acknowledgement from the neighbour_nodes

* 5. kill the node by sending it a poison pill

*/

I try to keep some sanity in the simulation by at least informing the the nodes which have the marked nodes in their neighbor list to remove it . In spite of this not all simulations are successful some result in exceptions and some go ahead.

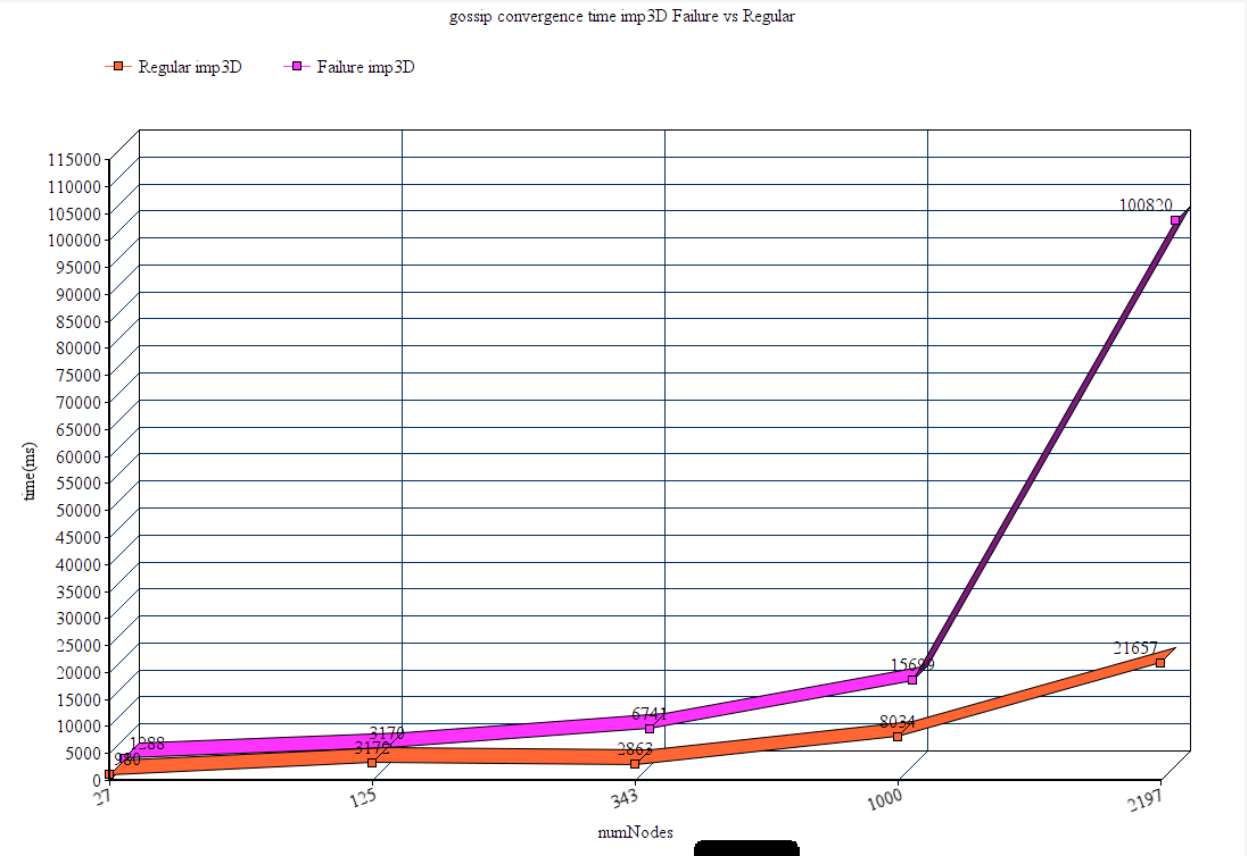
I carried out an experiment to contrast the convergence time with a simulaton with the below parameters-- (log file included in zip)

--failure mode---1000 nodes---imp3D---push-sum---10 initialDelay---100 interval---

vs

1000 nodes---imp3D---push-sum

The results are captured in the below graph.--



I thought it would be interesting to study how imp3D which performed best with push-sum fare when encountering failures--It can be seen that as the number of nodes in the sensor network increase the performance gap between the regular and failure model widens-this could be attributed in part to more number of failures in a large topology, especially if the random nodes which I believe gave an edge to imp3D over 3D are the one's that fail.

To run failure-mode from Gossip-failure

sbt "run <numNodes> <topology> <algorithm> <initial delay> <interval>"

The experiments were carried out on eclipse in a Ubuntu VM.