DOS PROJECT 2

Karan Goel 81160185 Siddhesh Muley 25901911

Gossip

For the Gossip Algorithm, a node sends a message to any random node connected to it. We set an upper limit on how many times a node can receive a message and stop the node once it has received the said number of messages(10 in this case). Eventually the entire network stops when we there are no more nodes left in the network for the message to be transmitted to. To solve the problem of a single node being left in the network, we check for active neighbors. If no neighbors are found, we stop that node too.

We will keep nodes working till 90% of the network has not converged as it is possible that some of the nodes might not have neighbours as they are being terminated and it might also be possible that they might not converge at all. Though our network converges most of the times, we can never really be certain of this for large number of nodes. The probability of this event occurring increases with the possibility of a failure.

No. of Nodes	line Topology	2D Topology	imp2D Topology	full Topology
5	232	276	242	191
100	1717	320	307	267
200	2860	375	297	311
300	4258	408	362	300
400	4126	507	407	322
500	5383	463	353	302
600	7922	441	364	324
700	7195	485	374	335
800	9912	474	409	321
900		463	398	313
1000		540	388	368

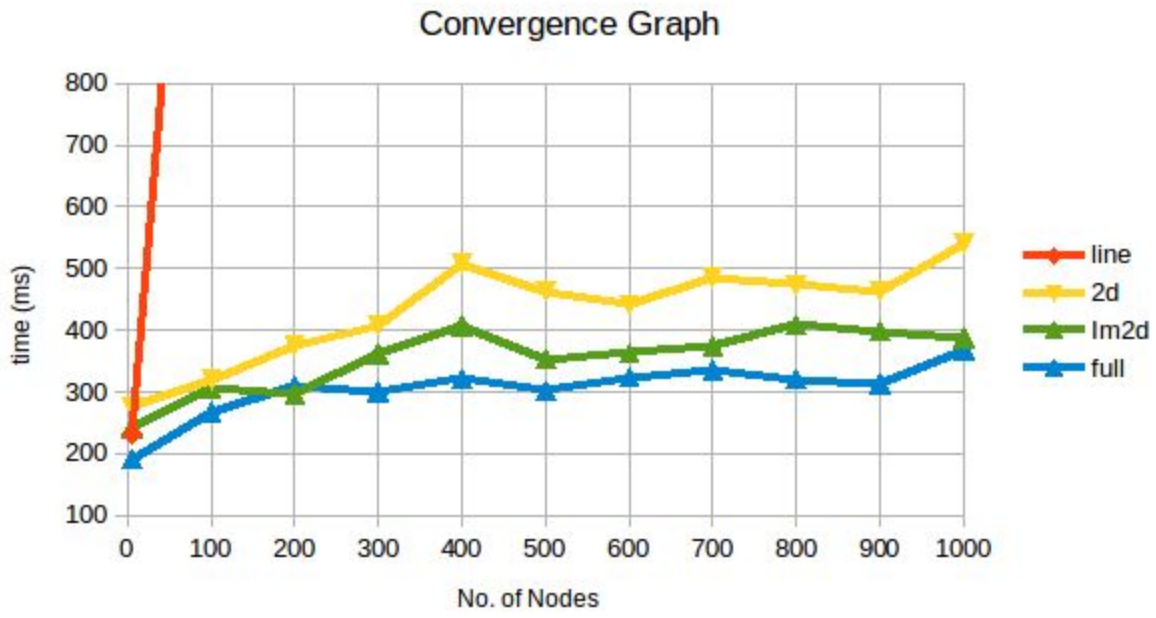


Fig 1: This is the convergence graph in the case of gossip protocol

No. of Nodes	line Topology	2D Topology	imp2D Topology	full Topology
10	298	268	231	244
100	1155	375	275	267
1000		529	395	308
10000		3926	3847	3543

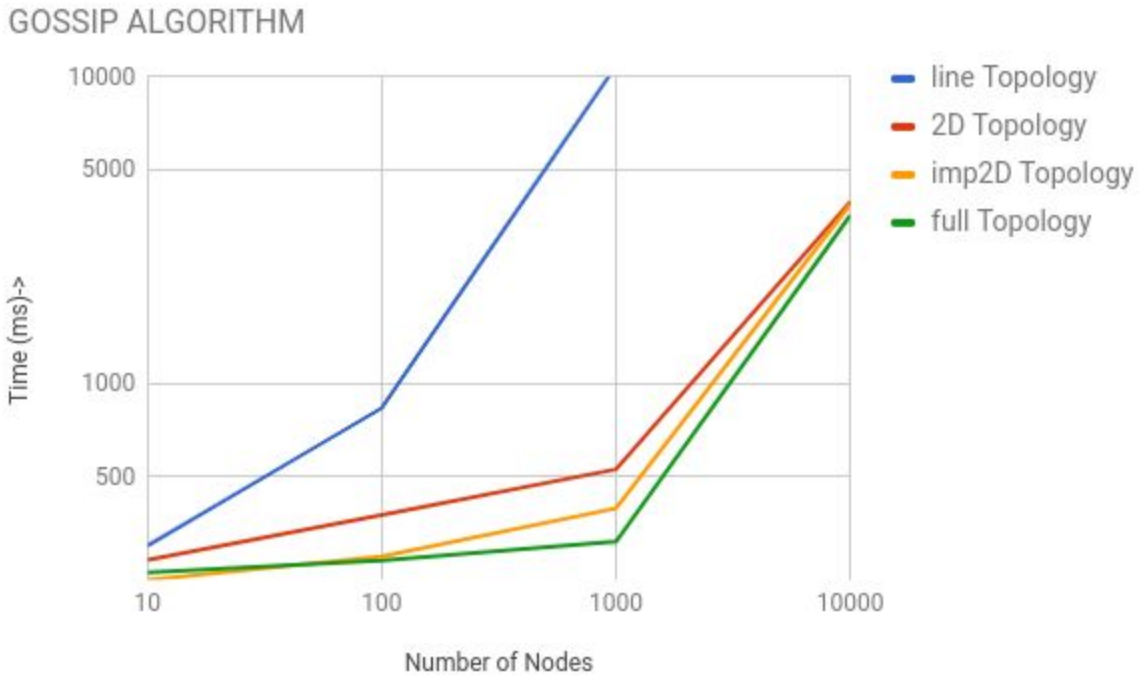


Fig 2: Logarithmic graph between number of nodes and time for different topologies

Push-Sum

1. Initially we will set all the nodes to have value (S,W) where S=i and W=1 , where i is the number of the node
2. Then we select a random node and send half value to the the random node and half to ourselves.
3. We will continue this process till the S to W ratio is greater than 10^-10, 3 consecutive times.[1]

No. of Nodes	line Topology	2D Topology	imp2D Topology	full Topology
5	2	1	1	4
100	37	29	17	9
200	96	32	21	10
300	168	48	25	22
400	504	74	42	28
500	757	127	63	32
600	1260	136	62	38

700		137	73	55
800		175	84	55
900		208	104	70
1000		274	127	81

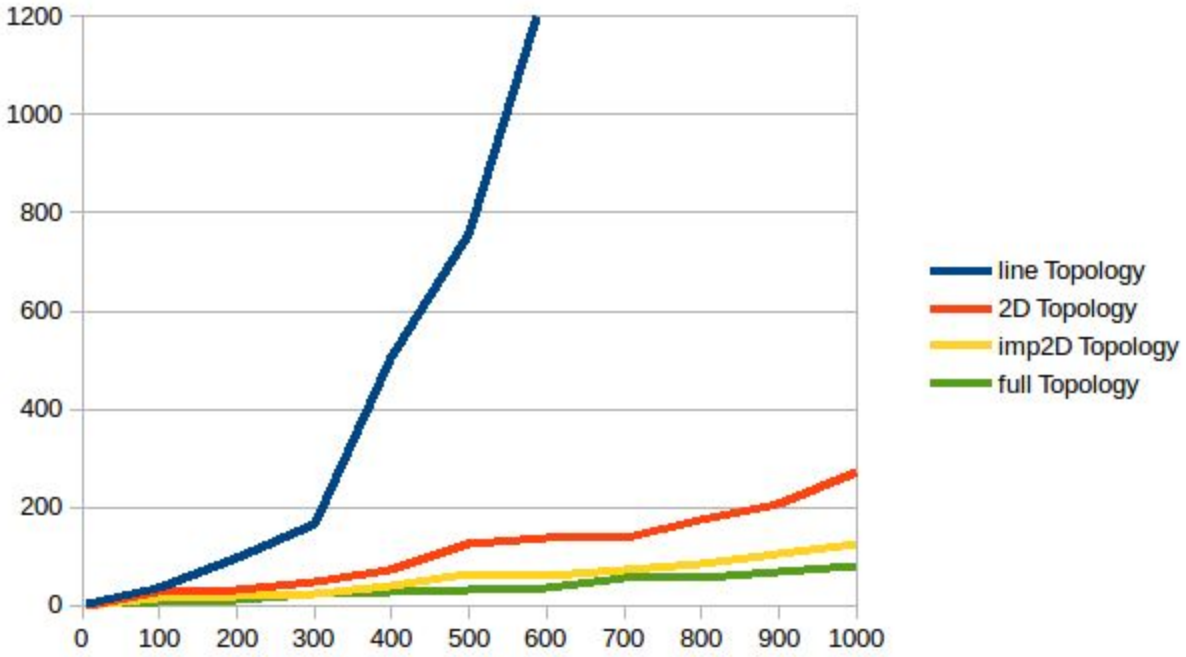
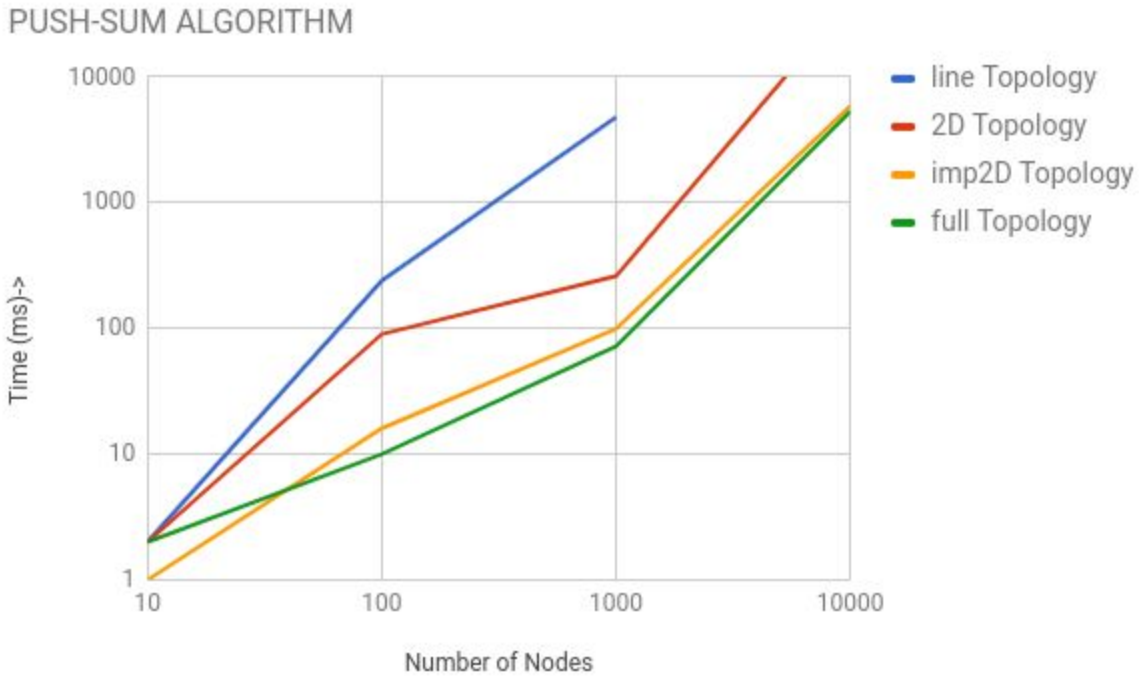


Fig 3: This is convergence time vs number of node graph for push-sum algorithm

	line Topology	2D Topology	imp2D Topology	full Topology
10	2	2	1	2
100	25	29	9	6
1000	2350	87	83	53
10000		5397	4690	4418



Analysis

Line topology in both the push-sum and gossip algorithm appears to be O(n**2). This is because every node can send messages back and forth and really doesn't give good convergence time at all.While the 2D, imp2D and full algorithms appear to be working in log time for small networks, but they go exponential after a point in the network.

We also observed that building the full topology takes a lot of time, but it's convergence time is really fast. The more neighbours you are connected to the faster the algorithm works.

We believe that the size of the network is only limited by the upper limit on the number of atoms that one can generate in Elixir.

References

[1] Kempe, David, Alin Dobra, and Johannes Gehrke. "Gossip-based computation of aggregate information." *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*. IEEE, 2003.