COP5615: Distributed Operating Systems
Project -2

Submitted by:
Sai Swetha Kondubhatla UFID: 1175 – 9282
Nikhil Reddy Kortha UFID: 7193 - 8560

Implementation details:
Gossip Protocol:
In our implementation of the gossip protocol algorithm, a node will stop transmitting once it has heard the rumor 10 times. It can still receive messages, but it stops transmitting the messages. Convergence occurs if all the nodes heard the rumor at least once. We maintain a table to note the number of nodes converged. When a node hears the rumor for the first time, we mark it converged in the table. If the table count is equal to the number of nodes, then convergence is achieved, and the program terminates after time for convergence is printed.

Push Sum Algorithm:
In our implementation of Push sum algorithm, a node exits if the s/w ratio did not change more than 10^{-10} in 3 consecutive rounds. Convergence is achieved if all nodes satisfy the above exit condition. We tried various other convergence techniques such as, converging when only one node satisfies the condition. But in large networks, this condition would be a wrong one because there might be a possibility that the other node values don’t converge but the exit condition would satisfy. This leads to the large variance of values.

Glimpse of Topologies:

- **Full Network** Every actor is a neighbour of all other actors. That is, every actor can talk directly to any other actor.
- **3D Grid:** Actors form a 3D grid. The actors can only talk to the grid neighbours.
- **Random 2D Grid:** Actors are randomly position at x, y coordinates on a [0-1.0]X[0-1.0] square. Two actors are connected if they are within .1 distance to other actors.
- **Sphere/Torus:** Actors are arranged in a sphere. That is, each actor has 4 neighbours (similar to the 2D grid) but both directions are closed to form circles.
- **Line:** Actors are arranged in a line. Each actor has only 2 neighbours (one left and one right, unless you are the first or last actor).

- **Imperfect Line:** Line arrangement but one random other neighbour is selected from the list of all actors.

Also, for random 2D, the minimum number of nodes for which the algorithm runs is 340. Since, if all the points are spread in 1 X 1 grid, for less than 300 points, there might be a possibility that for any centre there is no neighbour.

Interesting findings:
For gossip:
As we can see from the below tables for Imperfect Line, the convergence time is much less because, apart from its neighbors, it also spreads the rumor to one extra randomly picked node. This way the rumor is spread to all the nodes quickly. The convergence condition, i.e. all nodes should hear the rumor at least once, will be achieved faster than other topologies.
The next best convergence time is for 3D network, because, the spread is huge, and it has 6 neighbors across the network. Technically it should be highest for Full network, because the adjacency list has all the nodes except the node that it is transmitting. Since we are picking the node randomly, there is a possibility that the nodes are picked in a line fashion which aren't spread across the network. Hence the convergence time varies.

The convergence times is in the following fashion.
Imperfect line < 3D < 2D < torus < Full < Line

The convergence time for line is the highest because each node has 2 neighbors and it takes time to spread the rumor through those 2 nodes for the entire network.

Also, we noticed that if the stopping condition (i.e. the node stops transmitting once it has heard the rumor 10 times) value is increased, the convergence time also increases because the node which already received the gossip message might receive it again and again thereby reducing the chances of other nodes which haven't heard the rumor at least once

The converge times are as follows.

num of nodes	Full
10	4
100	22
500	366
1000	1800
1500	5000
2000	8500
5000	83198
6000	104565

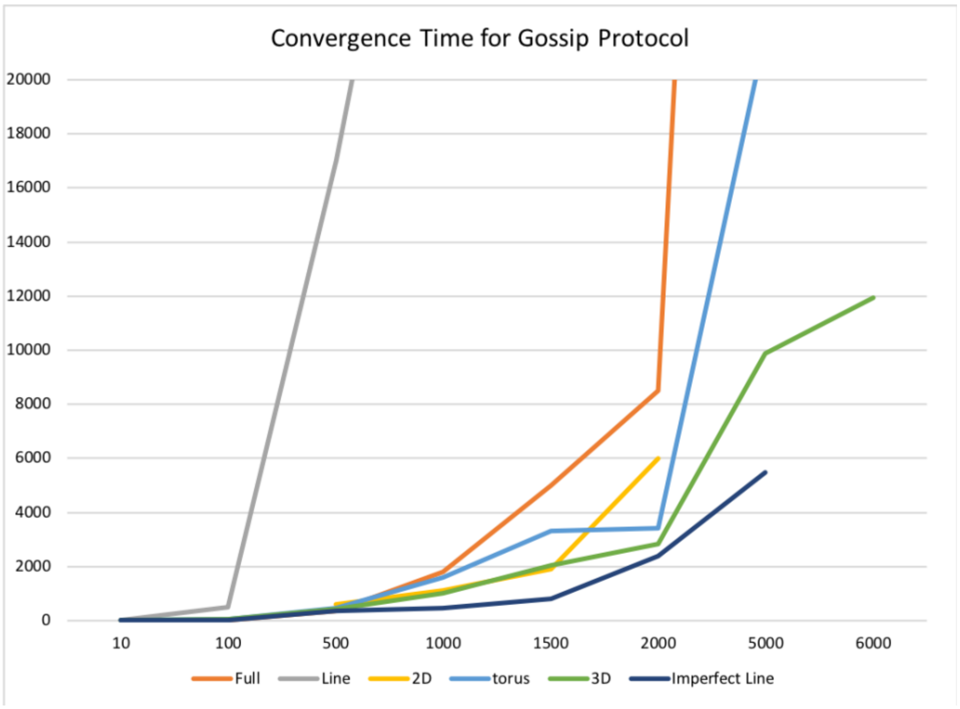
num of nodes	Line
10	8
100	500
500	16964
1000	37065
1500	40000
2000	160813

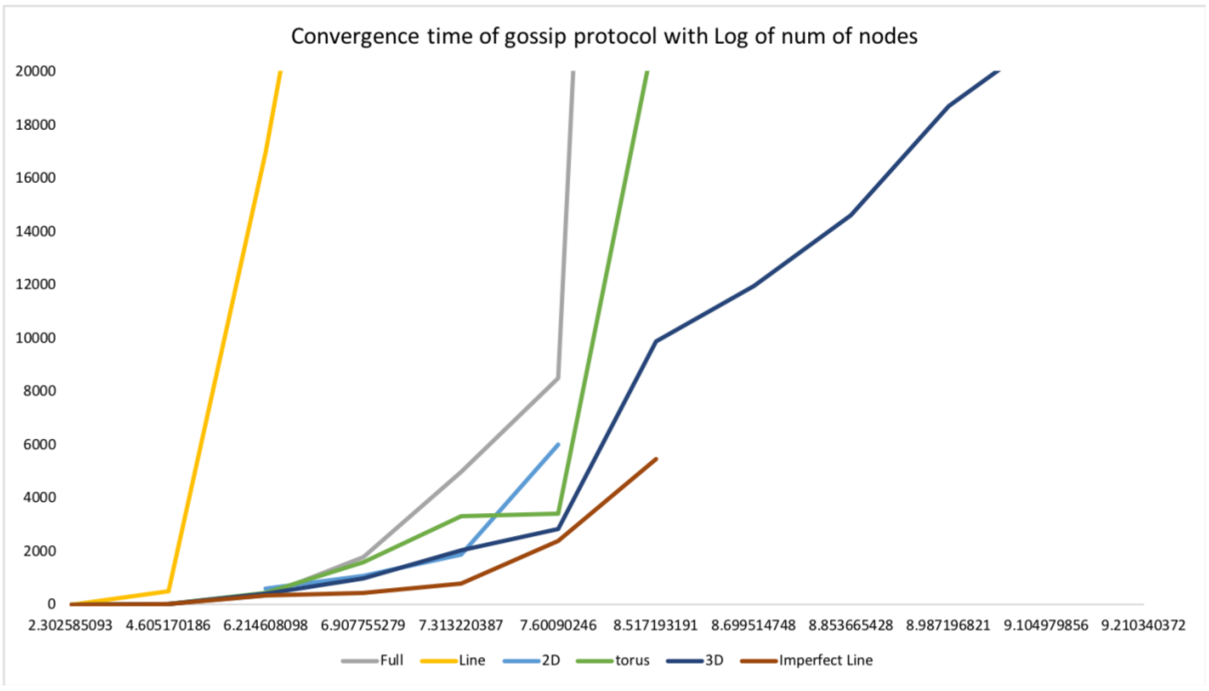
num of nodes	2D
10	
100	
500	600
1000	1100
1500	1895
2000	6000

num of nodes	Imperfect Line
10	8
100	28
500	352
1000	446
1500	800
2000	2388
5000	5469

num of nodes	torus
10	6
100	46
500	454
1000	1596
1500	3318
2000	3427
5000	21669
6000	25403

num of nodes	3D
10	4
100	37
500	429
1000	1001
1500	2044
2000	2841
5000	9884
6000	11950





For Push sum:
Similar to gossip, the convergence time for push sum **is less for Imperfect line and more for line** for the same similar reasons. The notable point in push sum is that, the maximum number of nodes for which the algorithm runs reduces when compared to gossip because, the convergence condition is if the s/w ratio did not change more than 10^{-10} in 3 consecutive rounds. This involves a lot more computations than spreading the rumour. Hence, the total number of nodes for which the push sum works is much less than the total number of nodes for which gossip works.

Also we noticed that if the **w value is increased**, the convergence is achieved faster, since the ratio of s/w reduces faster and thereby converging to 10^{-10} faster than the regular value.

The convergence times is in the following fashion.
Imperfect line < 3D < torus < 2D < Full < Line

num of nodes	2D
10	
100	
500	7309
1000	17518
1500	38442
2000	75558

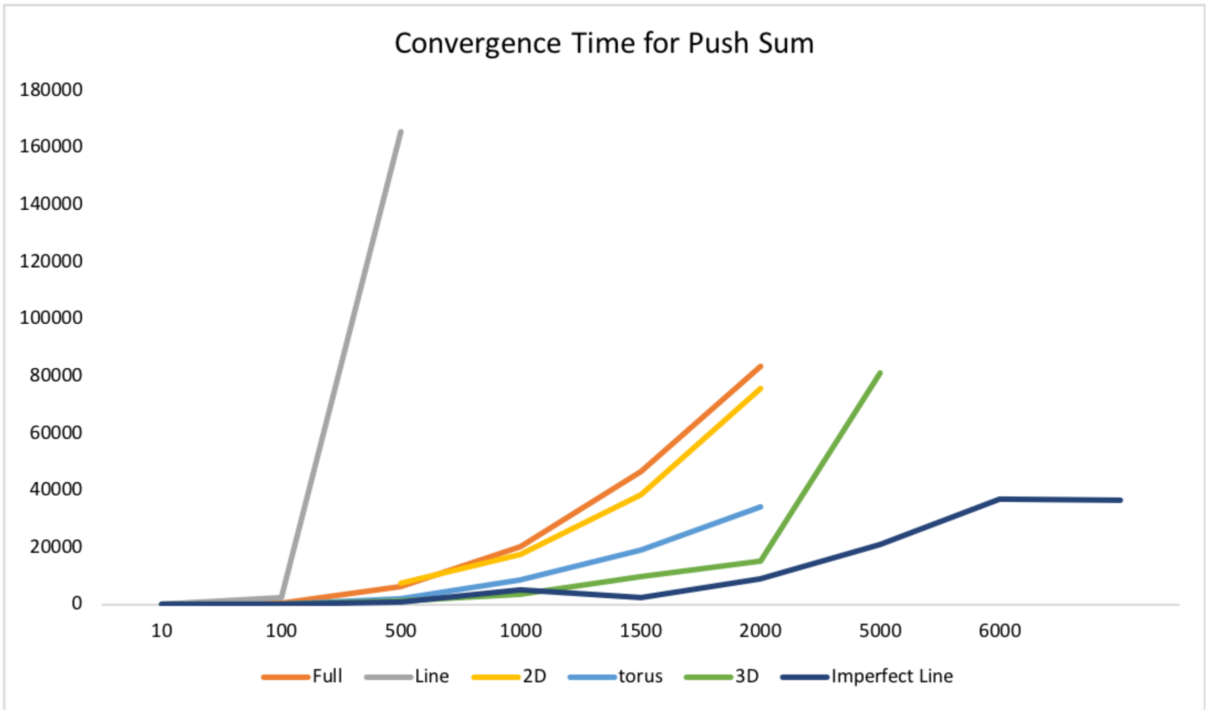
num of nodes	Line
10	7
100	2450
500	165502
1000	
1500	
2000	

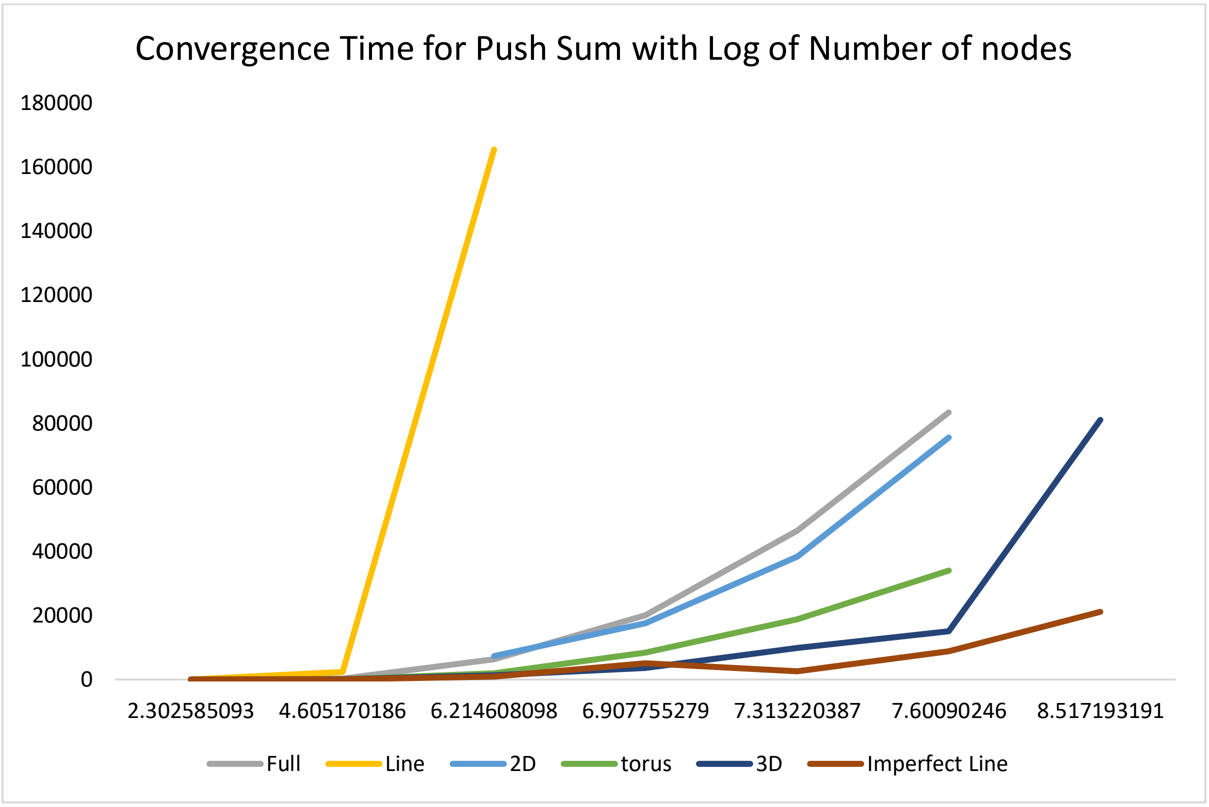
num of nodes	Full
10	6
100	355
500	6283
1000	20083
1500	46494
2000	83412

num of nodes	3D
10	4
100	102
500	1369
1000	3703
1500	9781
2000	15027
5000	81066

num of nodes	torus
10	5
100	88
500	1856
1000	8470
1500	18866
2000	34011

num of nodes	Imperfect Line
10	4
100	89
500	866
1000	4987
1500	2489
2000	8828
5000	21136
6000	36867
7000	36634





In both the algorithms, the nodes just become dormant with exiting. That is, in gossip, the node stops transmitting and in push sum the node will remain until the convergence is achieved. The failure of nodes isn't handled. **The bonus part of the project** handles failure of nodes by checking if the node is alive while picking the random node from the adjacency list.