COP5615 PROJECT REPORT

(Gossip Algorithm)

Srishti Jaiswal - 80385159

Implementation:

Gossip Algorithm:

According to the project's statement of purpose, a node terminates (reaches convergence) after receiving a rumor 10 times. If a neighboring node is still alive but unable to receive messages, the node will still transmit as long as it has those neighbors. Once all nodes have converged, the algorithm is finished. Here, it is determined how long it will take to reach convergence, including the start time, end time, and overall duration.

Push-Sum Algorithm:

Every node has an index and a weight in the push-sum implementation. Convergence occurs when the node receives three messages in a row without the index-to-weight (S/W) ratio changing by more than 10-10. Every time a message is sent, S and W's values are each reduced by half. Once all of its nodes have converged, the network should be shut off.

Topologies used:

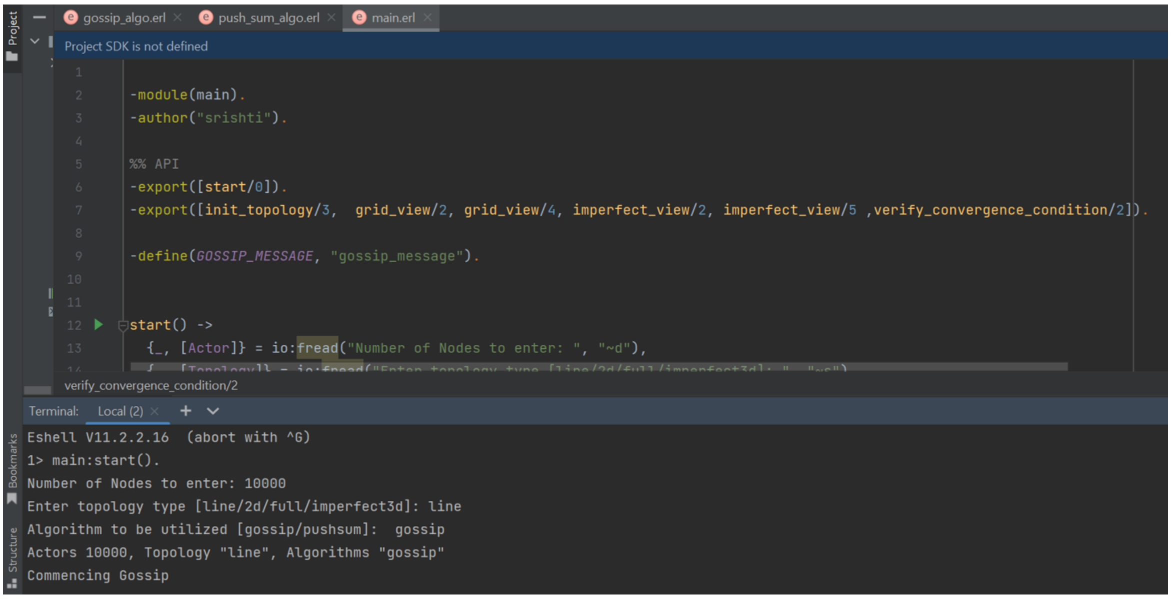
Line: With the exception of the first and last actors, each actor will only have two neighbors when they are organized in a line.

Full Network: All actors have one another as neighbors.

2D Grid: Each actor will have its adjacent and diagonal neighbors in the shape of a grid made up of the actors.

Imperfect 3D: In addition to one random neighbor being added to each actor's list of neighbors, it adheres to the 2D Grid pattern.

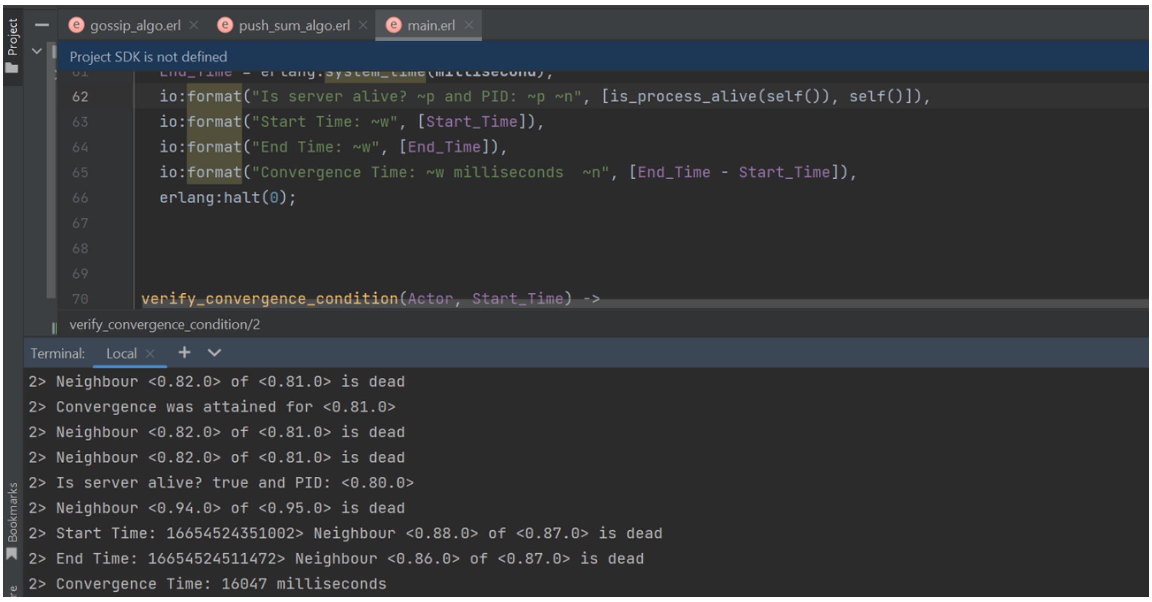
Output Screenshots:



```erlang

-module(main).
-author("srishti").

%% API
-export([start/0]).
-export([init_topology/3,  grid_view/2, grid_view/4, imperfect_view/2, imperfect_view/5 ,verify_convergence_condition/2]).

-define(GOSSIP_MESSAGE, "gossip_message").


start() ->
  {_, [Actor]} = io:fread("Number of Nodes to enter: ", "~d"),
```

```
Eshell V11.2.2.16  (abort with ^G)
1> main:start().
Number of Nodes to enter: 10000
Enter topology type [line/2d/full/imperfect3d]: line
Algorithm to be utilized [gossip/pushsum]:  gossip
Actors 10000, Topology "line", Algorithms "gossip"
Commencing Gossip
```

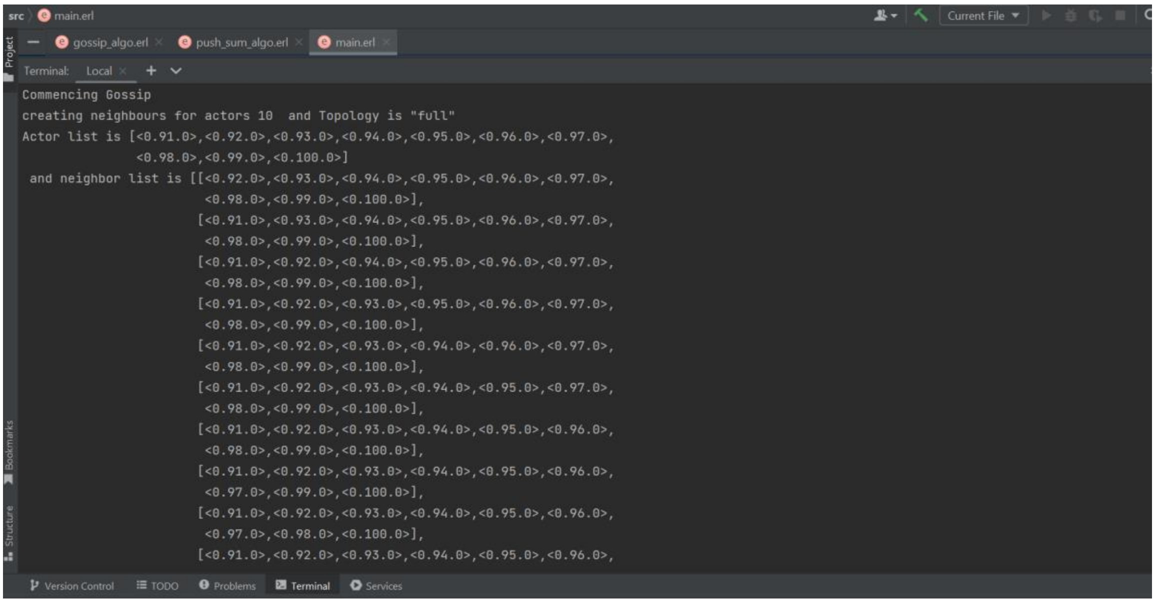
Fig: The input used to create the program is shown in the above figure (node count, topology and algorithm)



```
    End_Time = erlang:system_time(millisecond),
    io:format("Is server alive? ~p and PID: ~p ~n", [is_process_alive(self()), self()]),
    io:format("Start Time: ~w", [Start_Time]),
    io:format("End Time: ~w", [End_Time]),
    io:format("Convergence Time: ~w milliseconds  ~n", [End_Time - Start_Time]),
    erlang:halt(0);



 verify_convergence_condition(Actor, Start_Time) ->
```

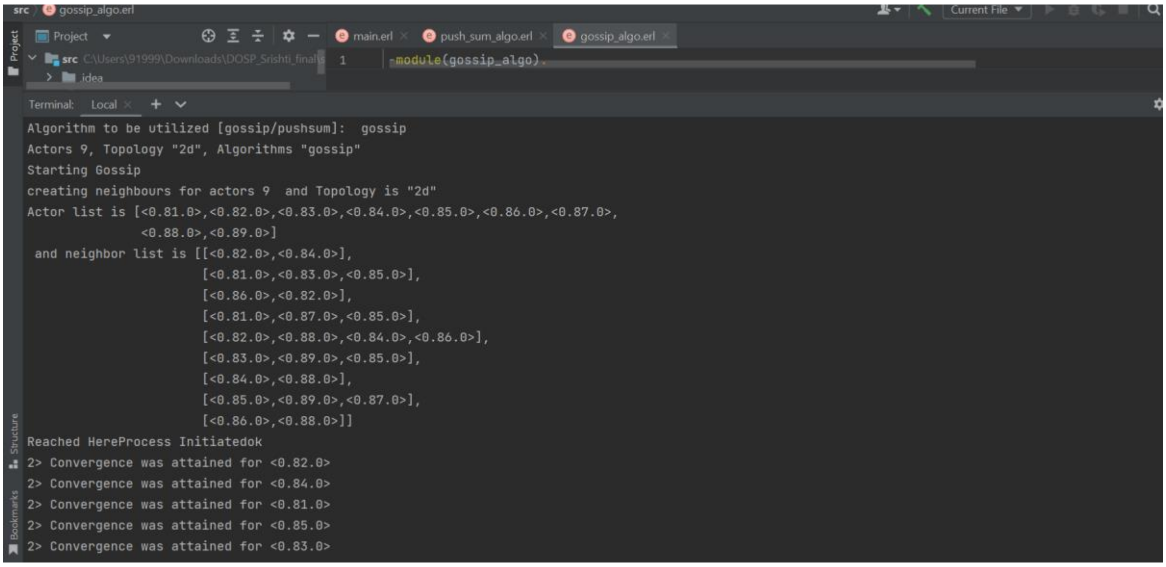
```
2> Neighbour <0.82.0> of <0.81.0> is dead
2> Convergence was attained for <0.81.0>
2> Neighbour <0.82.0> of <0.81.0> is dead
2> Neighbour <0.82.0> of <0.81.0> is dead
2> Is server alive? true and PID: <0.80.0>
2> Neighbour <0.94.0> of <0.95.0> is dead
2> Start Time: 16654524351002> Neighbour <0.88.0> of <0.87.0> is dead
2> End Time: 16654524511472> Neighbour <0.86.0> of <0.87.0> is dead
2> Convergence Time: 16047 milliseconds
```

Fig: The image above displays algorithm convergence



```
Commencing Gossip
creating neighbours for actors 10  and Topology is "full"
Actor list is [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,<0.97.0>,
               <0.98.0>,<0.99.0>,<0.100.0>]
 and neighbor list is [[<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.94.0>,<0.95.0>,<0.96.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.95.0>,<0.96.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.96.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.97.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,
                         <0.98.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,
                         <0.97.0>,<0.99.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,
                         <0.97.0>,<0.98.0>,<0.100.0>],
                        [<0.91.0>,<0.92.0>,<0.93.0>,<0.94.0>,<0.95.0>,<0.96.0>,
```

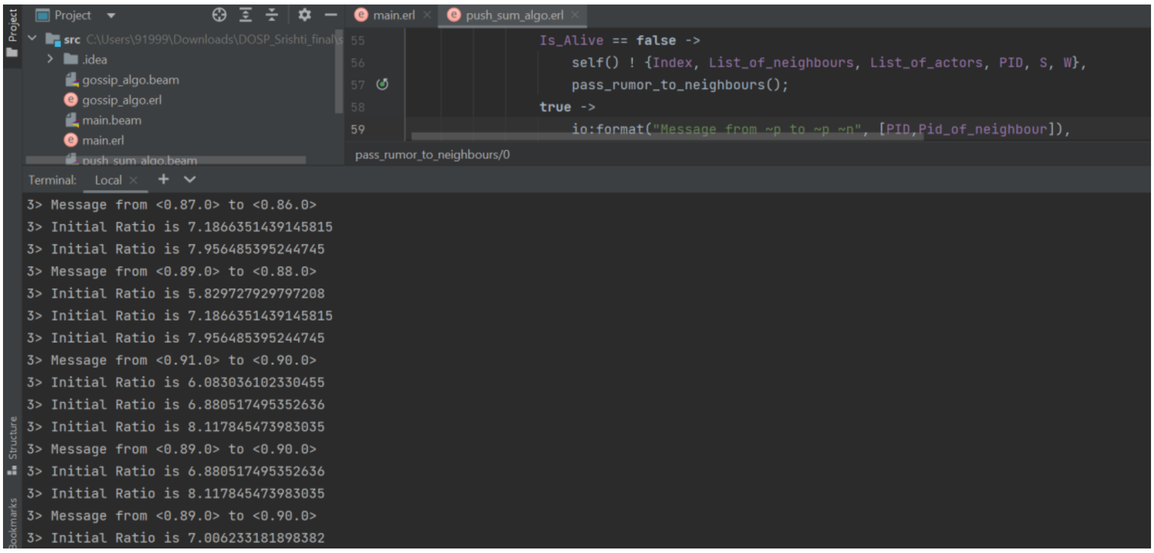
Fig: The neighbor list for Full topology is shown in the above image



```
-module(gossip_algo).
```

```
Algorithm to be utilized [gossip/pushsum]:  gossip
Actors 9, Topology "2d", Algorithms "gossip"
Starting Gossip
creating neighbours for actors 9  and Topology is "2d"
Actor list is [<0.81.0>,<0.82.0>,<0.83.0>,<0.84.0>,<0.85.0>,<0.86.0>,<0.87.0>,
               <0.88.0>,<0.89.0>]
 and neighbor list is [[<0.82.0>,<0.84.0>],
                       [<0.81.0>,<0.83.0>,<0.85.0>],
                       [<0.86.0>,<0.82.0>],
                       [<0.81.0>,<0.87.0>,<0.85.0>],
                       [<0.82.0>,<0.88.0>,<0.84.0>,<0.86.0>],
                       [<0.83.0>,<0.89.0>,<0.85.0>],
                       [<0.84.0>,<0.88.0>],
                       [<0.85.0>,<0.89.0>,<0.87.0>],
                       [<0.86.0>,<0.88.0>]]
Reached HereProcess Initiatedok
2> Convergence was attained for <0.82.0>
2> Convergence was attained for <0.84.0>
2> Convergence was attained for <0.81.0>
2> Convergence was attained for <0.85.0>
2> Convergence was attained for <0.83.0>
```

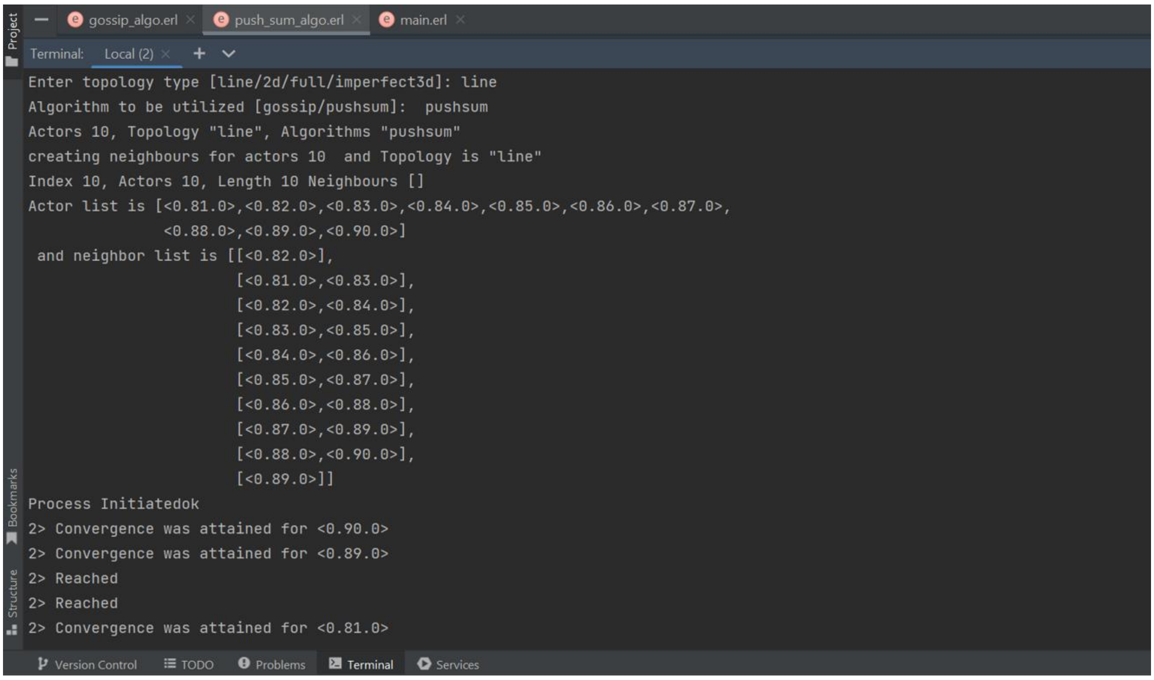
Fig: The neighbor list and convergence for 2d grid topology is shown in the above image



```
            Is_Alive == false ->
                self() ! {Index, List_of_neighbours, List_of_actors, PID, S, W},
                pass_rumor_to_neighbours();
            true ->
                io:format("Message from ~p to ~p ~n", [PID,Pid_of_neighbour]),
```

```
3> Message from <0.87.0> to <0.86.0>
3> Initial Ratio is 7.1866351439145815
3> Initial Ratio is 7.956485395244745
3> Message from <0.89.0> to <0.88.0>
3> Initial Ratio is 5.829727929797208
3> Initial Ratio is 7.1866351439145815
3> Initial Ratio is 7.956485395244745
3> Message from <0.91.0> to <0.90.0>
3> Initial Ratio is 6.083036102330455
3> Initial Ratio is 6.880517495352636
3> Initial Ratio is 8.117845473983035
3> Message from <0.89.0> to <0.90.0>
3> Initial Ratio is 6.880517495352636
3> Initial Ratio is 8.117845473983035
3> Message from <0.89.0> to <0.90.0>
3> Initial Ratio is 7.006233181898382
```

Fig: The revised ratio for the push-sum method is shown in the above figure



```
Enter topology type [line/2d/full/imperfect3d]: line
Algorithm to be utilized [gossip/pushsum]:  pushsum
Actors 10, Topology "line", Algorithms "pushsum"
creating neighbours for actors 10  and Topology is "line"
Index 10, Actors 10, Length 10 Neighbours []
Actor list is [<0.81.0>,<0.82.0>,<0.83.0>,<0.84.0>,<0.85.0>,<0.86.0>,<0.87.0>,
               <0.88.0>,<0.89.0>,<0.90.0>]
 and neighbor list is [[<0.82.0>],
                       [<0.81.0>,<0.83.0>],
                       [<0.82.0>,<0.84.0>],
                       [<0.83.0>,<0.85.0>],
                       [<0.84.0>,<0.86.0>],
                       [<0.85.0>,<0.87.0>],
                       [<0.86.0>,<0.88.0>],
                       [<0.87.0>,<0.89.0>],
                       [<0.88.0>,<0.90.0>],
                       [<0.89.0>]]
Process Initiatedok
2> Convergence was attained for <0.90.0>
2> Convergence was attained for <0.89.0>
2> Reached
2> Reached
2> Convergence was attained for <0.81.0>
```

Fig: The push-sum algorithm's convergence is seen in the above image

Largest Network:

For all topologies and both algorithms, the largest network was achieved using10000 nodes.

Convergence Time vs Size of the network:

Gossip Algorithm:

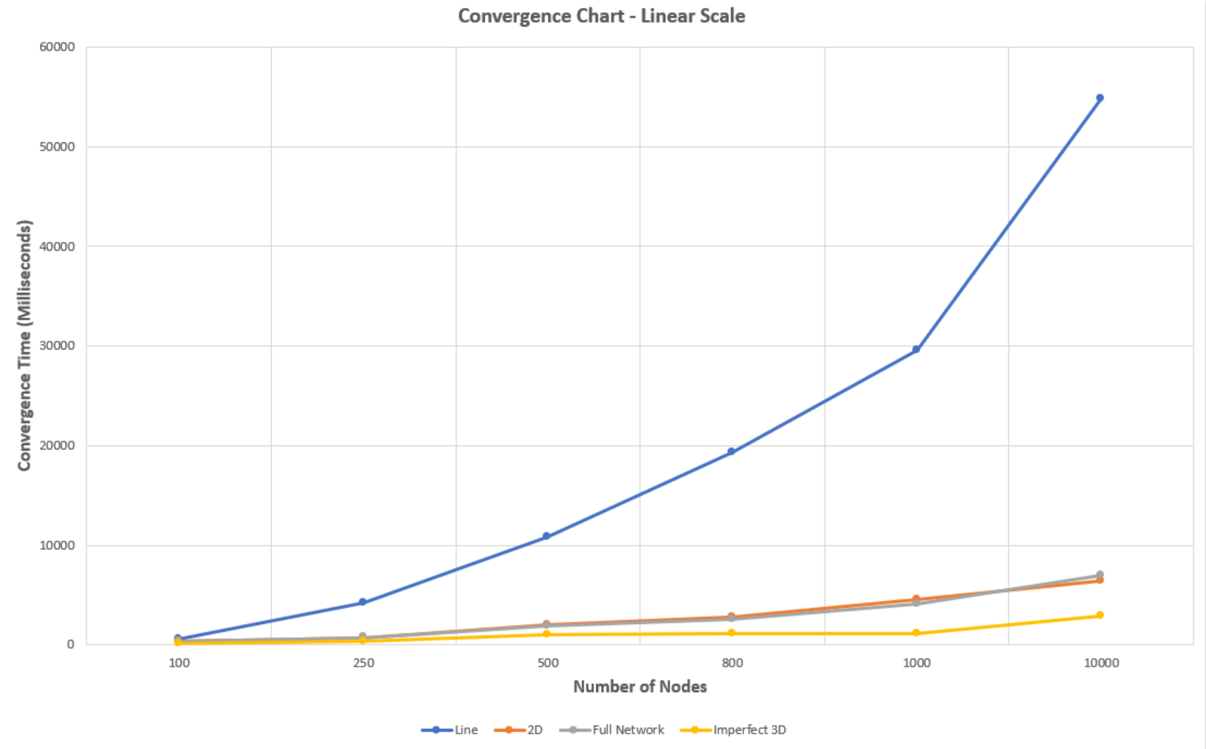


Fig: The graph above demonstrates how, for different topologies, the convergence times of the Gossip Algorithm change with the number of nodes.

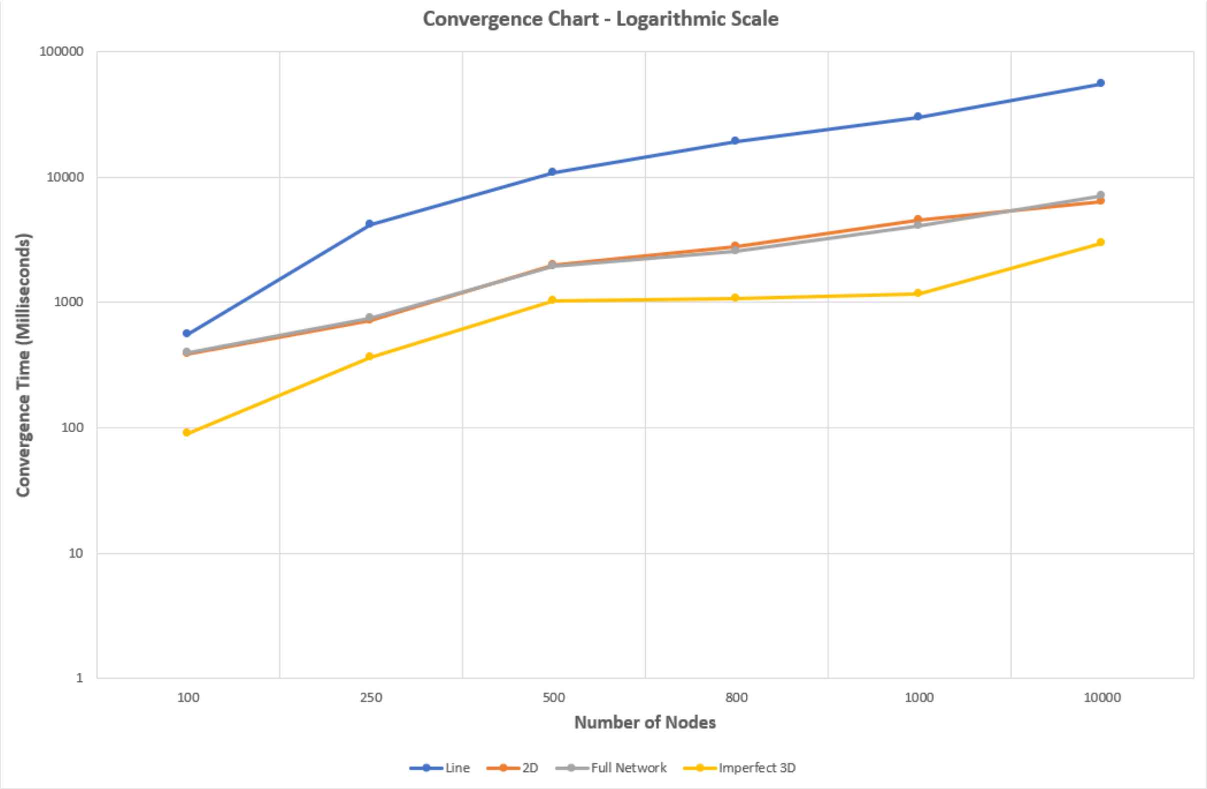


Fig: The graph above illustrates how, for different topologies, the log of the Gossip Algorithm's convergence times varies with the number of nodes.

Push-Sum Algorithm:

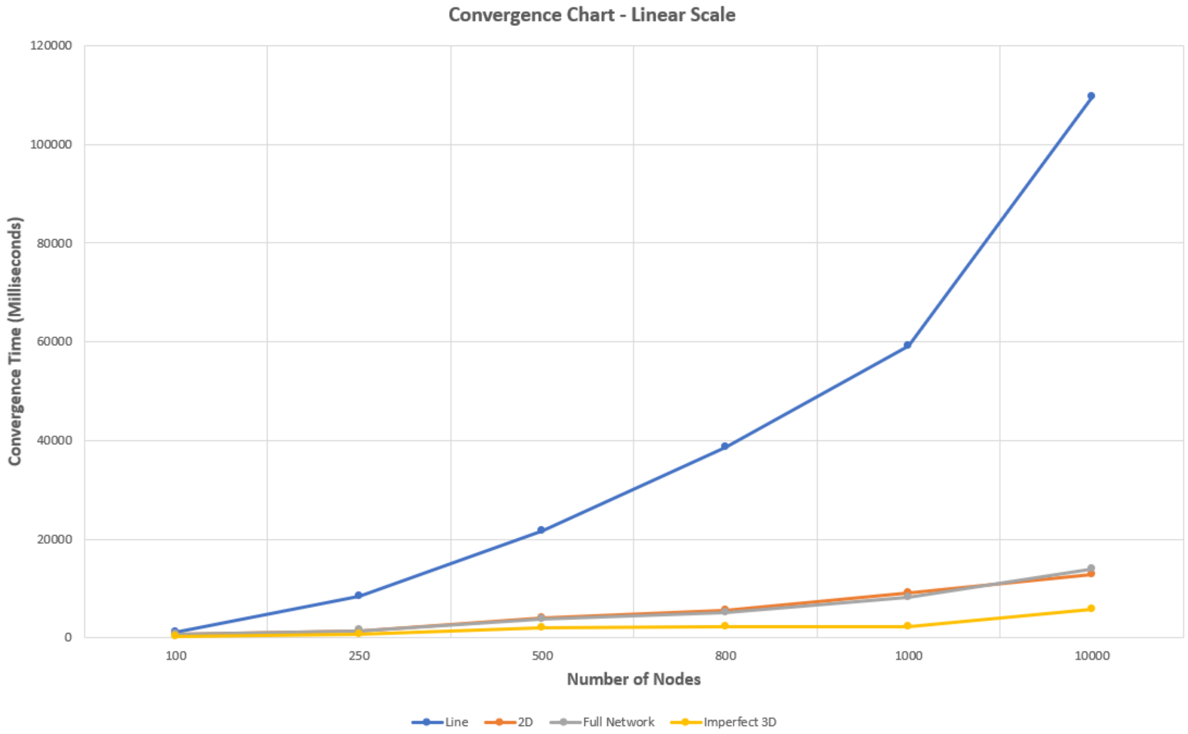


Fig: The aforementioned graph displays how the push-sum algorithm's convergence times change depending on the topologies and the number of nodes.

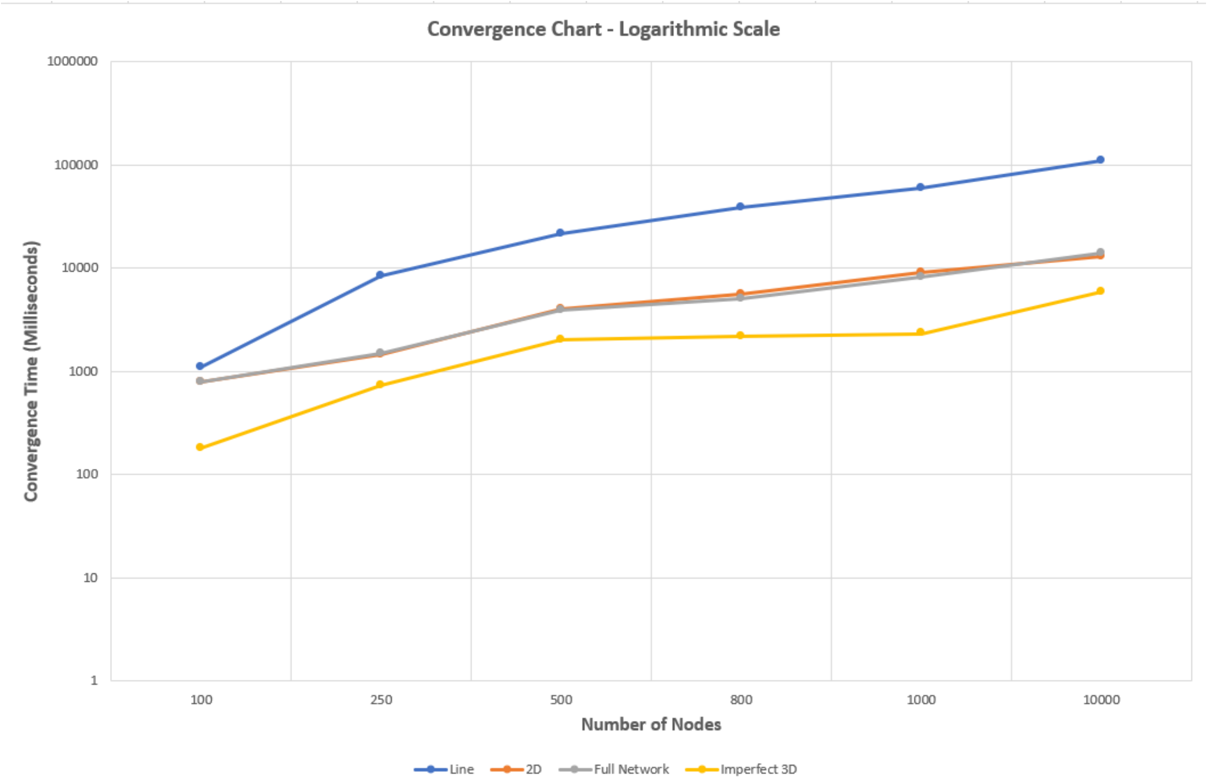


Fig: The aforementioned graph demonstrates how the push-sum algorithm's log of convergence times vary with the quantity of nodes for different topologies.

Reasonings:

1. The convergence occurred most quickly in imperfect 3D. This might be the case since the addition of the random neighbor node increases the number of sources from which messages can be sent, making it faster than the 2D grid.

2. Since each node can only have a maximum of two neighbors, the convergence of the line topology took the longest. Ten messages must be received by each node before they can converge.

3. Being connected to every neighbor, the full network topology is quicker than 2d grid and line topologies. Even with flaws, 3D topology outperformed the whole network in speed.

4. A circumstance where one or more nodes will never reach convergence emerges when there isn't a neighbor accessible to transmit (if all neighbors have attained convergence). This is prevented by making sure that nodes that have already reached convergence keep broadcasting after the procedure is finished.

5. Much more quickly than the push-sum method, gossip achieved convergence. This might be a result of the fact that 10^{-10} is a very small number and it took a long time for the updated weights to get to this value.

6. Despite the fact that nodes in a full network had more neighbors per node, both 2d and full network topologies demonstrated nearly comparable convergence times.