COP5615 – Distributed Operating Systems

Project 2 Bonus - Report

Team Members

1. Akshay Ganapathy (UFID - 3684-6922)
2. Kamal Sai Raj Kuncha (UFID - 4854-8114)

Input

The input to the program is the total number of nodes in the cluster (which is the total number of actors involved), the number of failure nodes used, the topology to be used (“line” or “full” or “3D” or “imp3D”) and the algorithm to be used (“gossip” or “part-sum”)

Output

The output is the number of nodes that were able to converge during execution of the algorithm with that topology.

Zip File Contents

The zip file consists of the Project_Report_Bonus.pdf file, and the project2_bonus.fsx file which contains the code to be run.

Topologies

The following topologies are used:

Full: Every actor is a neighbor of all other actors. That is, every actor can talk directly to any other actor.

Line: Actors are arranged in a line. Each actor has only 2 neighbors (one left and one right unless you are the first or last actor).

3D Grid: Actors form a 3D grid. The actors can only talk to the grid neighbors which ranges from 3 to 6 depending on the position.

Imperfect 3D Grid: It has the same grid arrangement as 3D Grid, but one extra random neighbor is selected from the list of all actors.

How To Run

Run the project2_bonus.fsx file using the command:
“dotnet fsi project2_bonus.fsx <numNodes> <failureNodes> <topology> <algorithm>”
where ‘numNodes’ is the total number of actors used,
‘failureNodes’ is the number of failure nodes,
‘topology’ is either of “line” or “full” or “3D” or “imp3D”, and
‘algorithm’ is either of “gossip” or “push-sum”

Platforms used for running the code

Visual Studio Code
.NET version 5.0
NuGET Akka.NET v1.4.25

Performed Experiment

The base code is similar to the main project but additionally nodes also implement the condition of node failure. The failure list contains randomly selected node ids for deliberate failure by supervisor. In case of failed nodes, convergence of the whole topology does not take happen for all nodes. Instead, we validate how many numbers of nodes converged successfully for a given number of failure nodes, using a convergence counter.
The experiment was conducted by keeping the total number of nodes in the cluster as constant. By keeping the cluster size constant and increasing the failure node input on each run, we can determine the total number of nodes which were converged and with that, we can analyze how node failures impact convergence of topologies in case of both algorithms.
In case of failure, topology will never converge fully. Therefore, in case the program gets freezes at a certain node because messages can't be transmitted ahead, a timeout occurs. This timeout period is equal to the topology successful convergence time for that cluster size which we know in advance. Using this timer, we are making sure that if the topology has not converged within the known convergence period, some failure has occurred, and we will wait for some extra time than actual convergence time and determine the number of nodes which converged successfully before blind spot occurred.

The following are the observations:

Gossip Failure Model Algorithm

The total number of nodes kept constant in the cluster while running the experiment was **1000** nodes for each topology. The relative order of the number of converged nodes obtained after running the Gossip Failure model algorithm for different topologies is as follows:

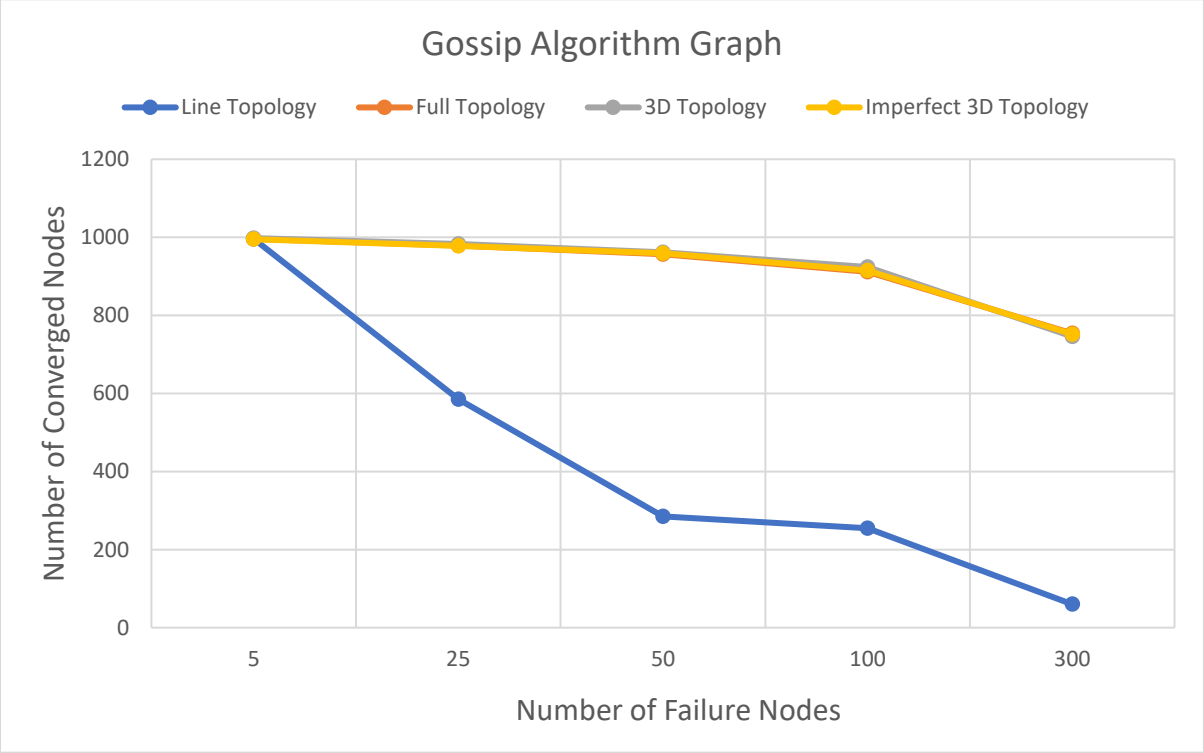
Full Topology > Imperfect 3D Topology > 3D Topology > Line Topology

Interesting Observations: On increasing the failure count of nodes in the cluster, it was observed that the convergence of nodes in line topology drops drastically as the failure nodes increase. This is because line topology is more prone to form blind spots as each node has only two neighbors and the blind spot can be formed in any direction. As a result, the connection to that block of nodes is lost. As the number of failure nodes increase, the convergence is not affected considerably in Full, 3D and Imperfect 3D topologies as compared to line topology.

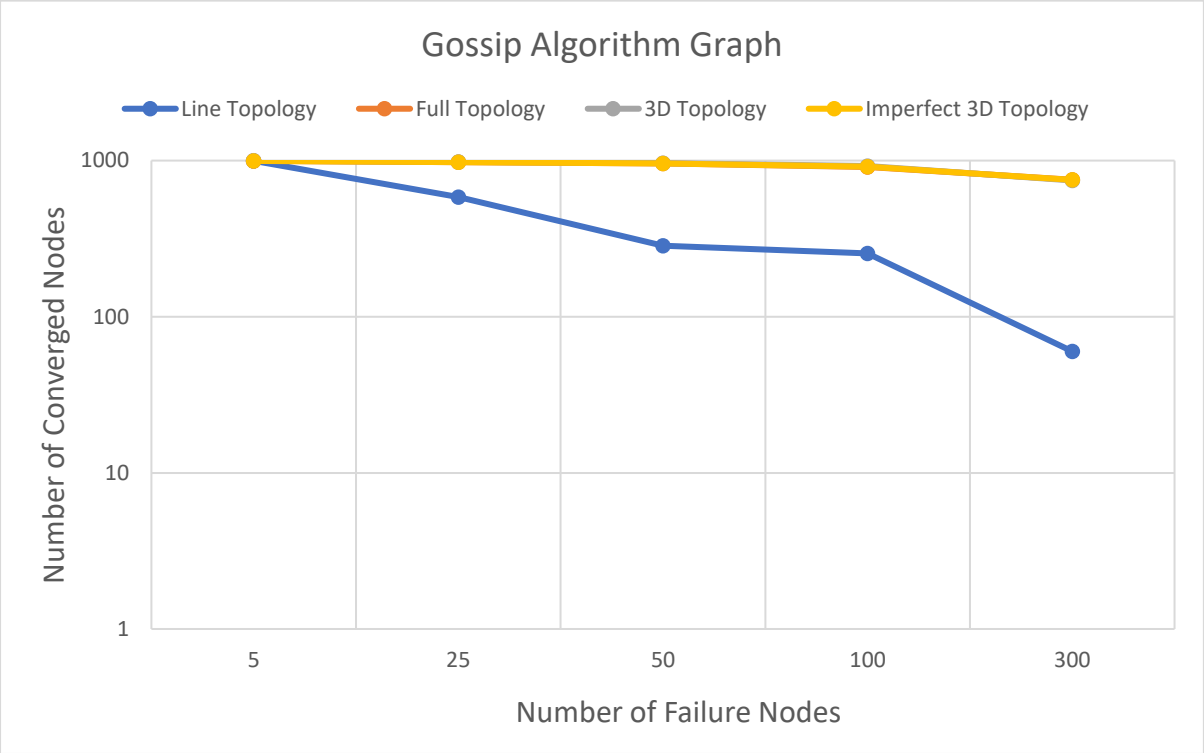
Execution table:

Total Number of nodes = 1000	Line Topology	Full Topology	3D Topology	Imperfect 3D Topology
Number of Failure Nodes	**Number of Converged Nodes**	**Number of Converged Nodes**	**Number of Converged Nodes**	**Number of Converged Nodes**
5	996	996	998	995
25	585	979	983	978
50	285	957	962	959
100	255	912	924	915
300	60	754	746	752

Linear Scale:



Log Scale:



Push-Sum Failure Model Algorithm

The total number of nodes kept constant in the cluster while running the experiment was **200** nodes for each topology. The relative order of the number of converged nodes obtained after running the Gossip Failure model algorithm for different topologies is as follows:

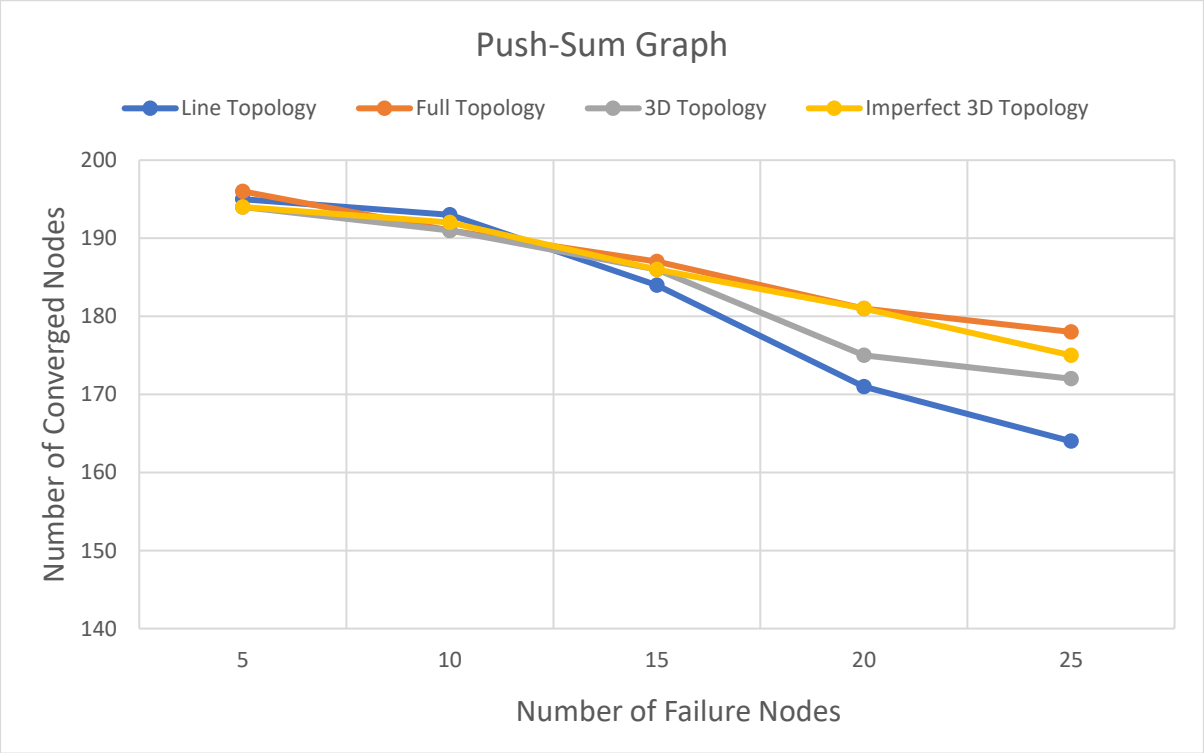
Full Topology > Imperfect 3D Topology > 3D Topology > Line Topology

Interesting Observations: On increasing the failure count of nodes in the cluster, it was observed that the convergence of nodes in line topology drops drastically as the failure nodes increase. This is because line topology is more prone to form blind spots as each node has only two neighbors and the blind spot can be formed in any direction. As a result, the connection to that block of nodes is lost. As the number of failure nodes increase, the convergence is not affected considerably in Full, 3D and Imperfect 3D topologies as compared to line topology.

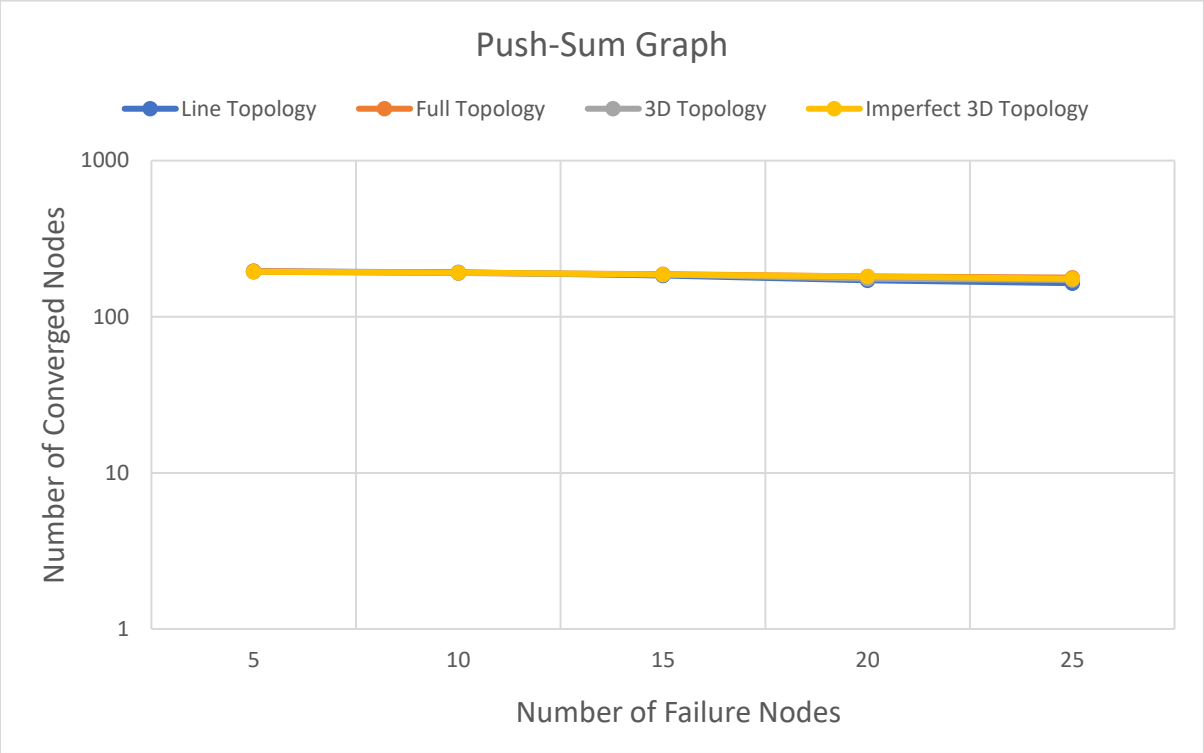
Execution table:

Total Number of nodes = 200	Line Topology	Full Topology	3D Topology	Imperfect 3D Topology
Number of Failure Nodes	**Number of Converged Nodes**	**Number of Converged Nodes**	**Number of Converged Nodes**	**Number of Converged Nodes**
5	195	196	194	194
10	193	191	191	192
15	184	187	186	186
20	171	181	175	181
25	164	178	172	175

Linear Scale:



Log Scale:


Push-Sum Graph
Line Topology
Full Topology
3D Topology
Imperfect 3D Topology
Number of Converged Nodes
1000
100
10
1
5
10
15
20
25
Number of Failure Nodes