Project Report: Distributed Operating System Principles (COP5615)
Project 2: Gossip and Push – Sum Algorithms Failure Model

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

Group Members: I don't have any group members. I have done this project alone.

Note: Please use following command for program execution if following error occurs (error FS3302: The package management feature requires language version 5.0 use /langversion:preview))

dotnet fsi --langversion:preview failuremodel.fsx totalNumberofNodes failureNodeNumber topology algorithm

- totalNumberofNodes = total number of nodes in cluster
- failureNodeNumber = number of nodes that should fail in cluster while execution of algos
- topology = topology name (line, full, 2D, imp2D)
- algorithm = algorithm name (gossip or push-sum)

Project Code, Design and Implementation:

1) Code is similar as gossip and push algorithms written in the main section the only difference is that in failure model implementation, code is implemented to fail given number of worker nodes on purpose in order to emulate real time failure scenario of distributed systems. Scenario of permanent failure of nodes is implemented.
2) f# code is implements four different methods which define four topologies - line, full, 2D, imperfect 2D
3) It implements a supervisor Actor which supervises the creation of topologies, creation of child actor nodes, and selection of algorithm (gossip, push sum) as per given input. Additionally, supervisor randomly selects certain number of nodes out of all the nodes in cluster (as per given input) for deliberate failure. The selected Node IDs are maintained in a failure node list for failure emulation once all the nodes start processing and exchanging messages
4) Child nodes are implemented to participate in implementation of gossip and push – sum protocol. Additionally nodes also implement condition of node failure which states that when an actor has received 2 messages from neighbor nodes, and if the node id of the node is present in failure list,(failure list contains randomly selected node ids for deliberate failure by supervisor) then the node fails by killing itself. This emulates real time Permanent Node Failure scenario of actual distributed systems.
5) In case of failed nodes, convergence of the whole topology will not happen for all nodes, instead we can check for given number of failure nodes, how many numbers of nodes actually converged successfully using a convergence counter.
6) Here are two possible cases, if node failures do not occur then topologies will converge successfully, but if there are failures then topologies will not completely converge in that case , total number of converged nodes are determined(converged nodes < total number of nodes in cluster , In case of failure).

Performed Experiment: The idea is to observe the effect of failed number of nodes on the convergence of gossip as well as push-sum algorithms for all topologies such as line, full, 2D and imperfect 2D.
These observations are made using following constraints:

1) For the observation of effect of node failures on convergence of topologies, The total number of nodes in cluster are kept as constant, also for that cluster size it is known in advance that the topology converges successfully if there are no node failures, the convergence time is also known in case of both the algorithms.

2) 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 data we can analyze how node failures impact convergence of topologies in case of both algorithms.
3) Ideally on increasing the failure nodes the convergence of nodes should decrease
4) Since in case of failure, topology will never converge fully hence in order to save the execution flow from entering an infinite loop or In case program gets stuck at certain node because messages can't be transmitted ahead, A timeout period is used, 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 , this means some kind of failure occurred, we will wait for some extra time than actual convergence time and determine the number of nodes which were actually converged before blind spot occurred in the program execution.

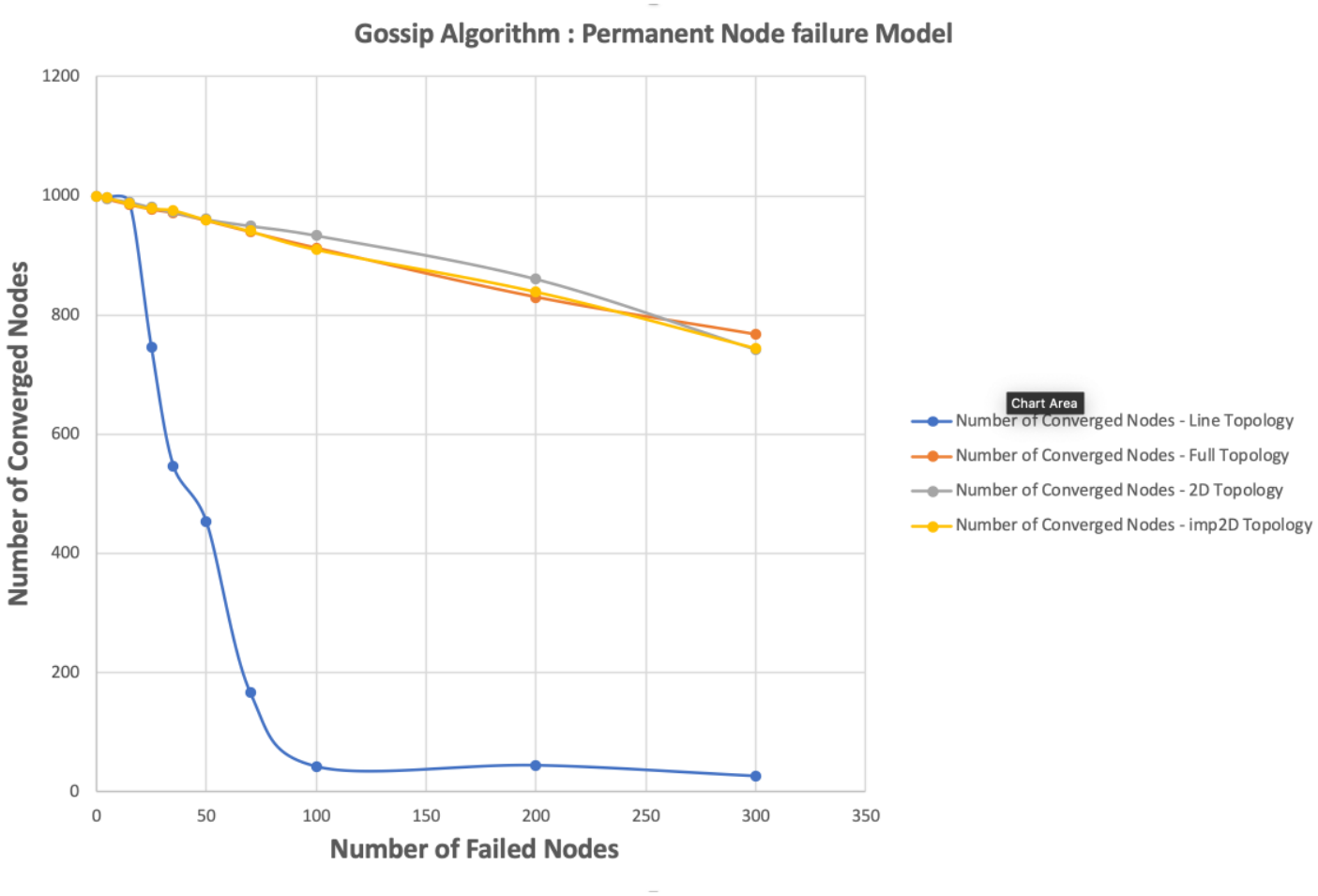
Problem 1: Gossip Algorithm for information propagation

1) **Experiment Parameters Chosen for Gossip Protocol Testing:** Total Number of Nodes = 1000 (for all topologies → line, full, 2D, Imp2D). It is known that for 1000 nodes all topologies converge fully without any error within 200 seconds. Hence the timeout period is used as: 5 minutes. Giving all extra time to algorithm, so that maximum number of converged nodes can be identified and counted. For each run the number of failed nodes is increased and number of converged nodes is determined.
2) **Observations made from the Experiment:** For above mentioned parameters on increasing the failure count of nodes in the cluster, it was observed that convergence of nodes in "Line topology" drops drastically as the failure nodes increase. It is justified because on failure of nodes line topology is more prone to form blind spots since each node in line topology has only two neighbors, if blind spot is formed in any direction the connection to that block of nodes is lost. However, it was observed that in case of full topology, 2D topology and imperfect 2D topology. When the failure nodes increase the convergence is not affected very drastically as compared to line topology. Observed Data – (number of Failed Nodes vs number of converged nodes) for cluster size of 1000:

Gossip Algorithm : Permanent Node failure Model				
Total Number of nodes in cluster : 1000				
Number of Failed Nodes	**Number of Converged Nodes - Line Topology**	**Number of Converged Nodes - Full Topology**	**Number of Converged Nodes - 2D Topology**	**Number of Converged Nodes - imp2D Topology**
0	1000	1000	1000	1000
5	998	995	996	997
15	989	986	990	987
25	746	978	981	979
35	547	972	973	976
50	453	959	961	960
70	166	940	950	941
100	42	913	934	910
200	44	830	861	839
300	26	767	743	744

Graph: Permanent Failure Model- Gossip algorithm:

(**number of Failed Nodes vs number of converged nodes**) for cluster size of 1000:



Problem 2: Push sum Algorithm for sum computation and convergence

1) **Experiment Parameters Chosen for Gossip Protocol Testing:** Total Number of Nodes = 200 (for all topologies → line, full, 2D, Imp2D). It is known that for 200 nodes all topologies converge fully without any error within 300 seconds in case of push sum. Hence the timeout period is used as: 5 minutes. Giving all extra time to algorithm, so that maximum number of converged nodes can be identified and counted. For each run the number of failed nodes is increased and number of converged nodes is determined.
2) **Observations made from the Experiment:** For above mentioned parameters on increasing the failure count of nodes in the cluster, it was observed that convergence of nodes in "Line topology" drops drastically as the failure nodes increase. It is justified because on failure of nodes line topology is more prone to form blind spots since each node in line topology has only two neighbors, if blind spot is formed in any direction the connection to that block of nodes is lost. However, it was observed that in case of full topology, 2D topology and imperfect 2D topology. When the failure nodes increase the convergence is not affected very drastically as compared to line topology. Observed Data – (number of Failed Nodes vs number of converged nodes) for cluster size of 200:

Push-Sum : Permananet Node Failure Model

Total No.of nodes in cluster = 200				
Number of Failed Nodes	**Number of Converged Nodes - Line Topology**	**Number of Converged Nodes - Full Topology**	**Number of Converged Nodes - 2D Topology**	**Number of Converged Nodes - imp2D Topology**
0	200	200	200	200
3	195	197	191	193
5	184	195	190	192
10	144	190	190	189
15	114	184	184	185
20	98	179	178	176
25	63	175	174	173

Graph: Permanent Failure Model – Push -Sum Algorithm

(number of Failed Nodes vs number of converged nodes) for cluster size of 200: