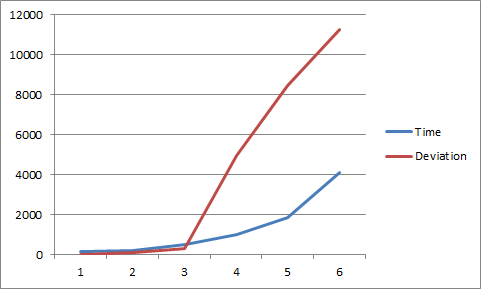
The bonus part was implemented using the following model.

As an input parameter, the “dropCount” variable can be enabled. This basically means that once the convergence starts to happen, you kill a random node which is active at that time after dropCount numbers of nodes have converged. This parameter can be provided as the fourth command line argument to the program.

This model assumes the random node selected permanently kills the node and the node will not start up later. This model was tested with the pushsum model. The results are shown below. A dropCount of 10 is the base case for the tests.

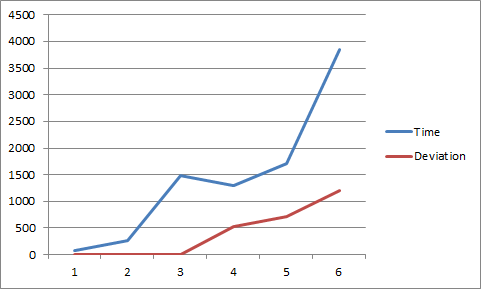
1. Line

We saw that for low number of nodes, before the monitor, which has the information about who has converged and who hasn’t receives 10 messages all of them have converged. As the number of nodes increased, the deviation from the expected sum value greatly increased.



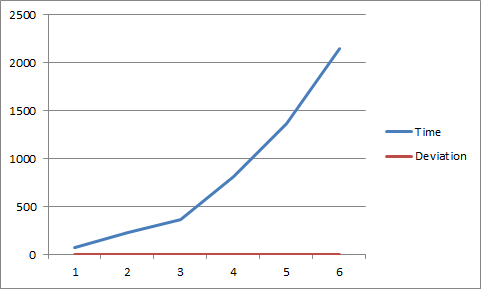
1. 2D

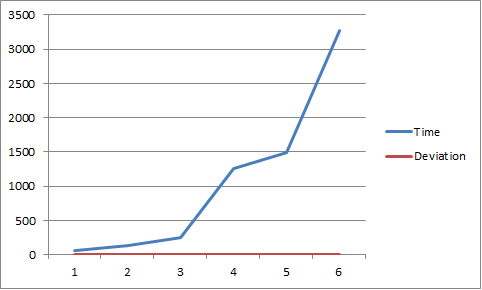
For the 2D topology, the results were comparable to the Line topology, but with the difference that the deviation from the expected result wasn’t as large as the line



1. Imperfect 2D

This topology was where we observed a different behavior. We noticed that all the nodes expect the one that was killed converged (for obvious reasons) and the convergence value was correct to a great extent( a precision ranging from 5 to 11 decimal places)



1. Full

Same results as Imperfect 2D. All nodes expect the one that was killed converged and the convergence value for all were correct to a great extent. ( a precision ranging from 5 to 11 decimal places)

Observations :

The Imperfect 2D and grid are able to converge despite a node dying. Whereas the other two don’t. We observed that full convergence wasn’t happening in the first two topologies anyways and the fact that a node died, just compounded the issue for them.

The code is written in such a way the before a node select one of its connections randomly to send the values, it checks if that node is active or not. This model is present in both the normal and bonus part. Thus, if we force a node to die, that information is propagated through the network and nodes stop sending them the message. Due to the all the messages that have already been passed around and the fact that one node isn’t participating anymore, doesn’t affect the overall outcome.