Gossip Simulator Bonus

Distributed Operating Systems

Project 2 Bonus - October 1, 2018

Madhukar Allu UFID : 79621171

Bhuvan Malik UFID : 70104341

Bonus Introduction

In this Gossip Simulator assignment, there is no failure at all. We are now implementing node and failure models, where a node dies, a connection dies permanently. The gossip and push sum algorithms are then run on the remaining nodes and the convergence times for various topologies are noted and compared.

Parameter used : “**Percentage**”. The percentage parameter denotes the percentage of the total number of nodes which we are killing at the beginning, when the gossip is starting.

For failure findings we tested all the topologies with both the algorithms. We incrementally increased the percentage of failure nodes and tested the converge.

Implementation Instructions:

Build the file using “**mix escript.build**”

Navigate to the “**project2-bonus**” folder in the command line and enter,

“**escript Project2_Bonus numNodes topology algorithm percentage**”

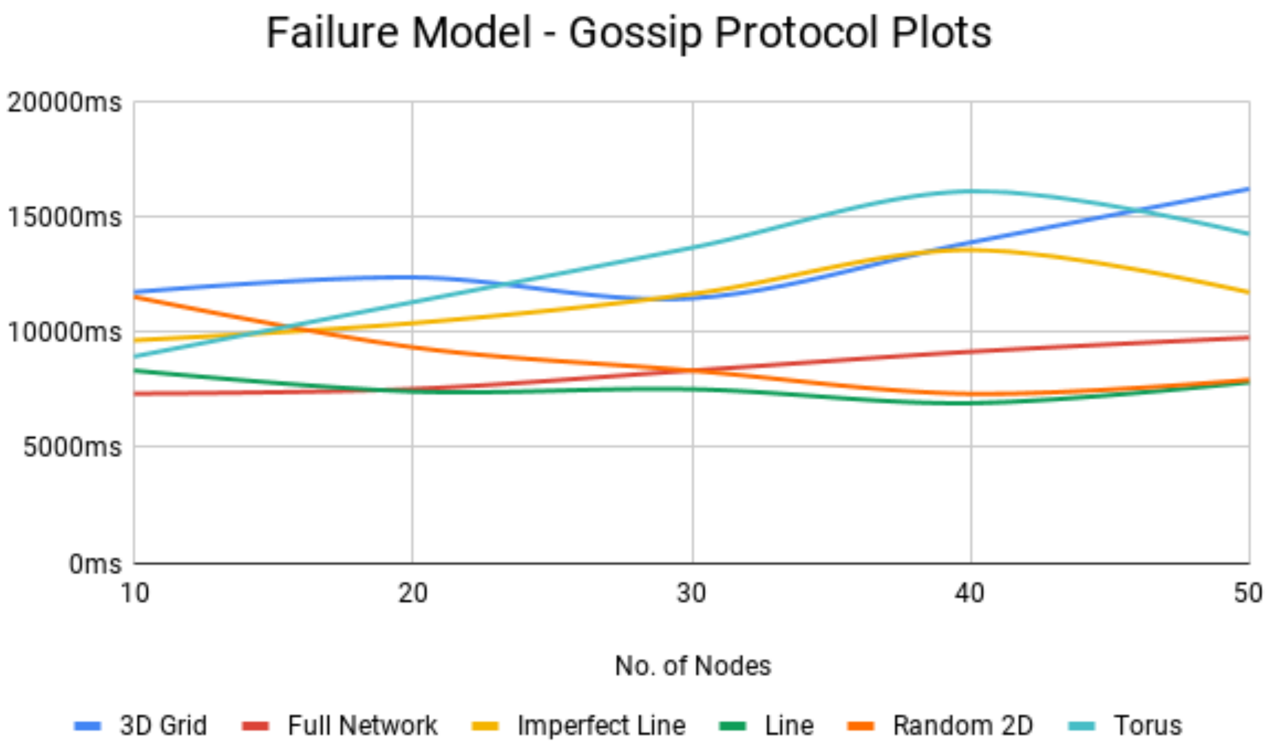
Make sure to enter the arguments in the mentioned order.

Output:

The output gives us the convergence time which is the time taken by the given set of arguments to form a network, run the algorithm and converge, which we have decided as (**Current system time** (end time) **- Start time**(time at which the first node receives the gossip)).

As we know, certain percentage of nodes will not be converging here, hence we have to wait for some time to make sure that there is no more convergence happening. Therefore, the output time includes a wait time of 5000 milliseconds. Due to this, in certain topologies where randomization plays an important role, we see some significant variations in the plots. We will look into that further when we plot the graphs below and make observations.

Gossip Protocol :

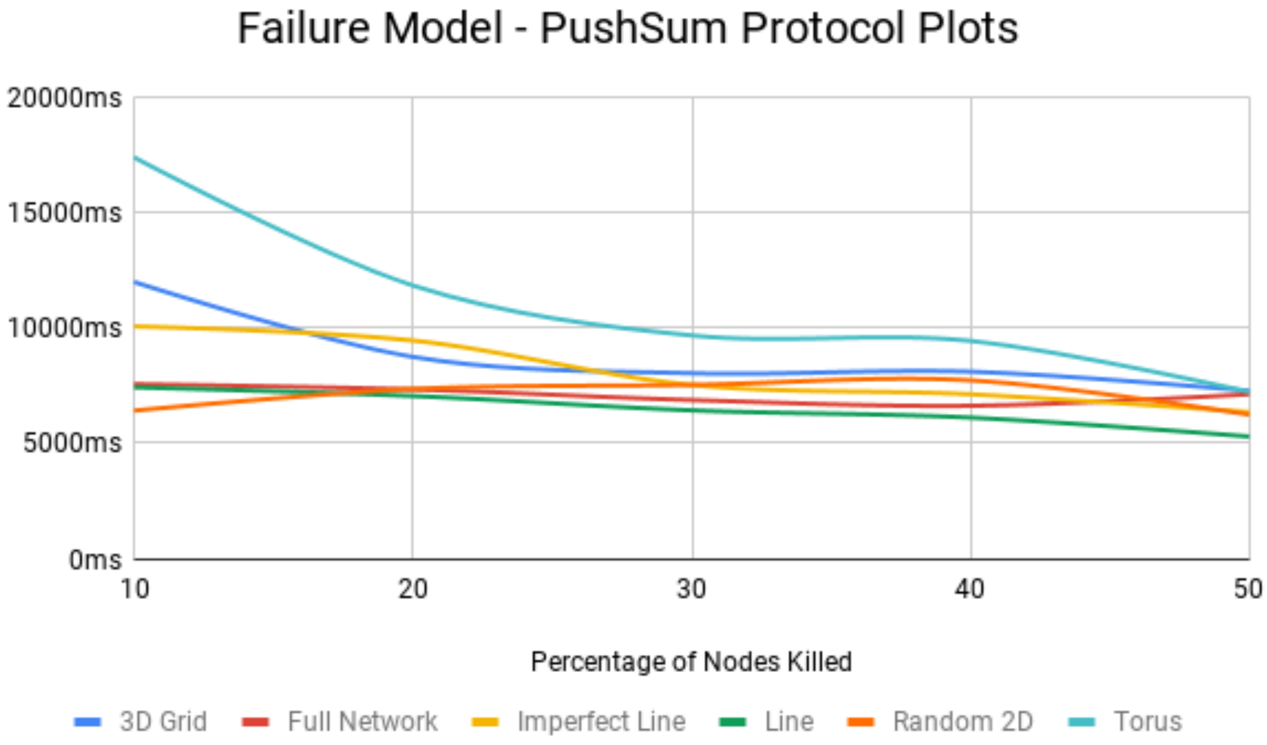


The above graph plots the convergence time of different topologies vs percentage of Nodes that we killed randomly. The number of nodes were kept constant (**200** in these observations) Following observations were made:

The convergence time for **Line** topology for 200 nodes using Gossip protocol was 15000ms when implemented without failure model. When we are killing some random nodes, the flow of messages is obstructed as the message is passed to only 2 neighbors by each actor. Hence as less actors are converging, the convergence time drops. Hence the convergence time here is between 7000-8000ms.

In **Random 2D** topology, all the actors are packed in a fixed are, hence the lower the number of nodes, lesser the neighbors fro each actor and as a result many nodes do not converge. Therefore, as the percentage of nodes being killed increases, the convergence time decreases.

PushSum Protocol :



The above graph plots the convergence time of different topologies vs percentage of Nodes that we killed randomly. The number of nodes were kept constant (**200** in these observations) Following observations were made:

The convergence time for **Line** topology for 200 nodes using Gossip protocol was 2559ms where as it is between 5500-7500 when we are deleting certain nodes. This is due to the flow of messages being obstructed because of the random deleting of actors.

The convergence time for **3D** and **Torus** topology decreases as we increase the percentage of nodes being killed as the neighbors for certain actors might have been deleted when we randomly killed the actors and hence the 3D structure is incapable of reaching all the alive actors, hence not all the actors which are alive converge, resulting in a lower convergence time.

In general, the more the number of neighbors, the greater the fault tolerance a network has.