COP5615: DOSP Project 2 Bonus

Name: Pranath Reddy Kumbam
UFID: 8512-0977

Problem Statement

Implement node and failure models to existing work.

language: Erlang

Implementation

A node failure model has been added the existing implementation of Gossip and Push Sum. The model is parametrised by the percentage of node failures (rate) which determines the number of randomly selected nodes that should fail in the network. We use this model to test the spread of rumour on a fixed network for a set amount of time. For this report, I have tested the failure model with the Gossip algorithm on a “full” network with 6000 nodes and recorded the number of nodes converged in a set time period of 8000 milliseconds.

Execution Details

The program takes four parameters, number of nodes, the algorithm, topology, and lastly the rate of node failure.

```
-> c(project2_bonus).
-> project2_bonus:main(6000, "full", "gossip", 0.5).
```

Submission Details

The submission zip file contains the code, times (all the recorded values used for the plots), results, and plots.

Results

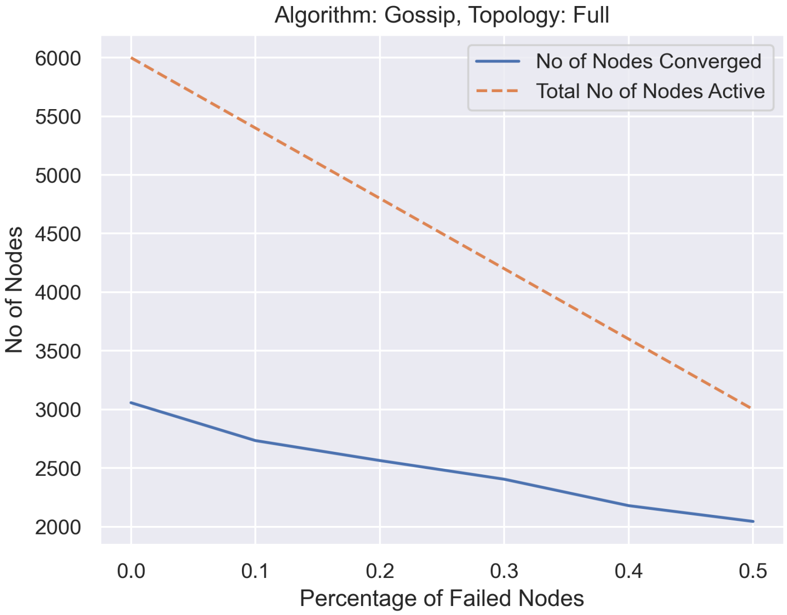

Figure 1: Number of nodes converged in T=8000ms

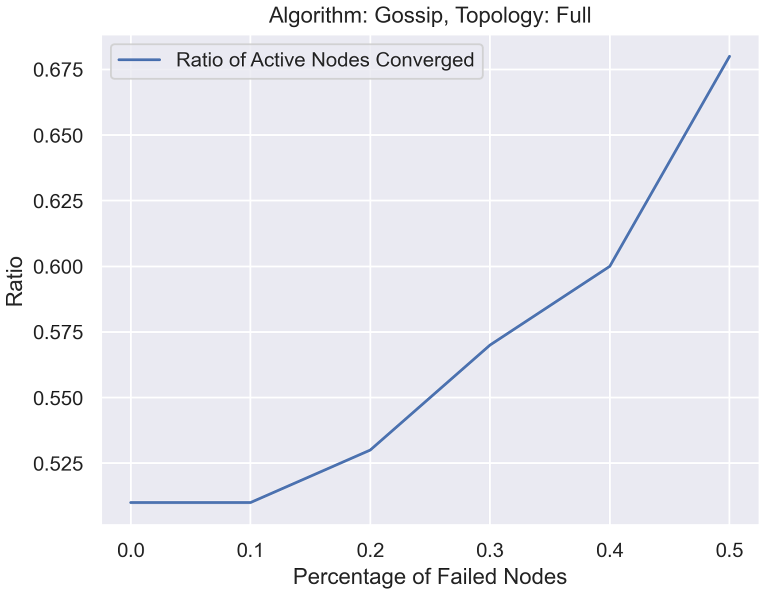

Figure 2: The percentage of active nodes converged

Observations and Notes:

1. As seen in the results, the spread of rumour decreases as we increase the rate of failure, which makes sense since increasing the rate of failure decreases the connections in the network.
2. Although the spread of rumour is inversely proportional to the rate of failure, from the second plot, we can observe that the ratio of the available nodes that converge increases as we increase the rate if failure and constrain the network, which makes the algorithm more efficient.