

# COP5615- Distributed Operating System Principles

## Project 3 Bonus Report

Submitted by:

Naman Arora (UFID: 3979-0439)

Drona Banerjee (UFID: 4662-7749)

### Objective:

This bonus part was aimed at testing the resilience of a tapestry network. This is accomplished by successively failing nodes in the network in bulk. Then, in each attempt, a random alive node is selected as a *publisher* and two others as *requesters*. The *publisher* is tasked with publishing a message in network while the first *requester* is supposed to route to that object. Then, the *publisher* un-publishes that particular object, while the second *requester* tries to find it.

### Methodology:

The experiment was run with two topologies, with 1000 and 2000 nodes each, failing 5%, 10%, 15% and 20% of nodes totaling to 8 experimental runs. Each run has been tested 5 times to provide consistency in output. All these functions are the part of **DOLR** API as specified in [1].

*Publish* and *un-publish* functions are best attempt functions and return no confirmation. While *route to object* function can either return the requested object or fail. Note that *route to object* and *add node* functions have been exhibited in the main part of this project.

### Result:

In our successive attempts to render the topology unusable, we have failed and thus established its high degree of resilience.

The results are tabulated as following:

1000 nodes	0%	5%	10%	15%	20%
<i>F1</i>	2	2	2	2	2/3
<i>F2</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>

Table 1: Test runs for 1000 nodes in the network.

2000 nodes	0%	5%	10%	15%	20%
<i>F1</i>	2	2	2	2	2/3
<i>F2</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>

Table 2: Test runs for 2000 nodes in the network

*F1* represents the *route to object* function call before *un-publish* and *F2* represents the contrast.

Note that, the tables tabulate the *Hops* required to fetch the object. Also, value of “*N*” shows that the object was never found. An uncertain value is represented as “\_/\_” which shows the integral bounds of uncertainty.

### Findings:

The above experiments have proven that tapestry is a very resilient network. The introduction to backup links in the network endows it with such failure resilience.

### Epilogue:

Since the main purpose of a network is to fetch and publish objects, we conclude that the implemented tapestry protocol is very efficient and resilient in doing so.

### References:

[1] [https://pdos.csail.mit.edu/~srib/docs/tapestry/tapestry\\_jsac03.pdf](https://pdos.csail.mit.edu/~srib/docs/tapestry/tapestry_jsac03.pdf)