

# Distributed Operating Systems

## Project 3

### Group Members-

Name: Akash Jajoo                      UFID: 61326882

Name: Niraj Chowdhary              UFID: 15013123

### Implementation (Bonus part)-

The project implements Tapestry algorithm with the failure of nodes as an input parameter. In the project, random nodes are killed permanently according to the number entered by the user. Each node stores up to 3 values in each slot of their routing table. The nodes in the slots are sorted according to their distance from the root node. Hence, the first value in the slot will always be the closest to the root node. If the first value of the slot dies then it is replaced by the second value. If the second node is also killed then the third value in the slot is selected.

When a node is deleted, it is removed from the routing table of all other alive processes.

### Input-

`mix run project3.exs numNodes numRequests numKill`

Here, numNodes – Number of nodes created in the peer-to-peer system.

numRequests – Number of requests each node has to make.

numKill – Number of nodes to be killed.

**Example-** `mix run project3.exs 100 2 4`

Here, 100 is the number of nodes.

2 requests are sent by each node.

4 nodes are killed.

### Output-

Maximum hops is 2 for 100 node(s) with 2 request(s) after 4 node(s) die permanently.

### Observations-

We observe that if less than 50% of the nodes fail, each node can still reach the destination with **log N** number of hops. Hence the system is resilient.

### Sample execution snippet-

```
C:\Users\Akash\Desktop\tapestry>mix run project3.exs 1000 3 100
Compiling 1 file (.ex)
Maximum hops is 4 for 1000 nodes with 3 request(s) after 100 node(s) die permanently
```