**PROJECT 3 - PASTRY ALGORITHM**

**RESULTS SAMPLE**

1) Number of Requests = 3, b = 3

| Number of Nodes | Expected Hop Count (log n) base = 2^b | Actual Hop Count |
|---|---|---|
| 1000 | 3.32 | 3.00 |
| 3000 | 3.85 | 3.34 |
| 5000 | 4.09 | 3.49 |
| 7000 | 4.25 | 3.61 |
| 10000 | 4.42 | 3.73 |

2) Number of Requests = 4, b = 3

| Number of Nodes | Expected Hop Count (log n) base 2^b | Actual Hop Count |
|---|---|---|
| 1000 | 3.32 | 2.97 |
| 3000 | 3.85 | 3.33 |
| 5000 | 4.09 | 3.49 |
| 7000 | 4.25 | 3.60 |
| 10000 | 4.42 | 3.74 |

**OBSERVATIONS**

- We were able to implement the pastry algorithm for 10000 nodes.
- Average hop count for improves relative to the number of nodes in the network.
- Worst Case Hop Count is bounded by log n (base = 2^b).
- When routing tables are updated keeping in mind locality properties, there is an improvement in the average hop count. We tested this by sending the new node and complete routing tables of the new node to nodes in the route path when a new node joins versus sending only the new node.
- Increasing the size of the Leaf Set also improves the average hop count. It is expected as the probability of finding one of the k nodes increases.

## RESULTS SAMPLE UNDER FAILURE

1) Number of Requests = 3, b = 3

| Number Of Nodes | Expected Hop Count (log n) base = 2^b | Actual Hop Count (NO FAILURE) | Actual Hop Count (5 % FAILURE) | Actual Hop Count (10 % FAILURE) |
|---|---|---|---|---|
| 1000 | 3.32 | 3.00 | 3.1 | 3.2 |
| 3000 | 3.85 | 3.34 | 3.5 | 3.7 |
| 5000 | 4.09 | 3.49 | 3.6 | 3.8 |
| 7000 | 4.25 | 3.61 | 3.8 | 4.1 |
| 10000 | 4.42 | 3.73 | 3.9 | 4.2 |

## OBSERVATIONS UNDER FAILURE

- We simulated random node failure in the network. We do not observe any failure to route messages under nominal failure rates.
- The average hop count increases in proportion to the failure rate but is still bounded by log n (base = 2^b)

Team Members:

Abhinav Rungta UFID: 69517289

Akchay Srivastava UFID: 17991933