COP 5615

Project 3

Submitted By-:

Rajat Koujalagi

rkouj@ufl.edu

6138-2944

Largest Network:

100,000 Peers with an average hop of 4.68

Implementation:

Each peer will have a routing table and a leaf set from which it can choose an appropriate peer to route information. I have chosen the value of b=4 and L=16. Each peer will first look in its leaf set. If key is present in the leaf set then the search stops. If not, the algorithm looks for the closest peer in the routing table (closest in terms of common prefix) and route the key to it.

This process continues until destination is reached. As we are moving 16 bits closer each time on an average, the search will terminate in $\log_{16}N$ steps where N is the total number of peers.

Failure Model:

How to run

scalac project3bonus.scala

scala project3bonus numNodes numRequests numFailures

This will kill numFailures amount of nodes after the network is created. Following are the observations after implementing the failure model

How failure nodes are handled

I have created a random generator that will generate the failure peer ids and then store them. While routing, each peer will determine a path to the target. If any of the peers on the path are idle then the source will find a closest peer and route the information to that peer.

When failures occur, there are largely two chunks that need to be fixed to bring the network back, the idle node's id that lies in the leafsets of other nodes and the idle node's id that lies in the routing table of other nodes.

To update the routing tables, the peers that have the idle ids in their routing table will first remove it. Now they will send a message to peers closer to the idle peer. These close nodes typically lie in the positions that are near to the idle nodes' id in the routing table.

To update the leafsets, the nodes in the leafsets of every dead node will remove the idle nodes' ID. Now the peers that need to refill their leafset will send a message to the last node that lies on the same side of the idle node.

The average number of hops will increase first because some of the nodes will not get an accurate replacement of the idle node's ID.

The average hops will also decrease as idle peers lead to fewer peers in the network.

The system with failures will have some hops increased, but attainment of destination is guaranteed.