Project 3: P2P

Team members

- Vishisht Khilariwal
- Siddhant Jain

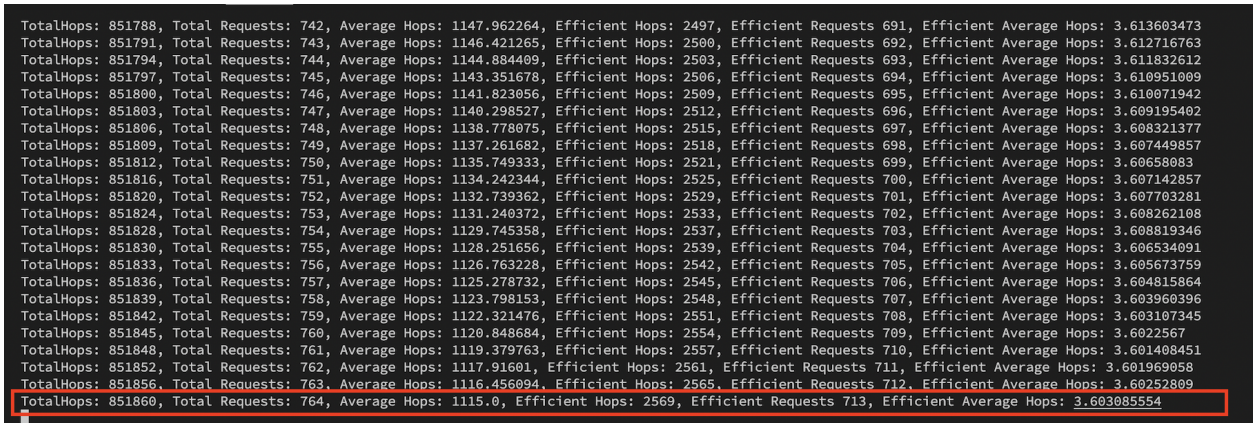
Command to Run : **dotnet fsi Chord.fsx 100 100**

What is working

- Actor Nodes are created and then joined to each other using the Join API defined in the research paper
- Every actor stabilises itself periodically (every 1 sec) to fix its neighbours i.e the predecessors and the successors with the logic defined in the paper
- Every actor fixes it's finger table to enable efficient routing in log(N) steps in worst case by running fix finger API defined in the paper every second
- Nodes send requests to find a key in the system

When the system is not stable the finger tables are still being fixed. The messages sent at that time take longer to route and sometimes may not be delivered since all the nodes might not have been discovered.
After the System stabilises and the finger tables are also stable the routing becomes efficient

Below is the screenshot for 80 nodes equally distributed in the network represented in 8 bits(Max keys = 256)

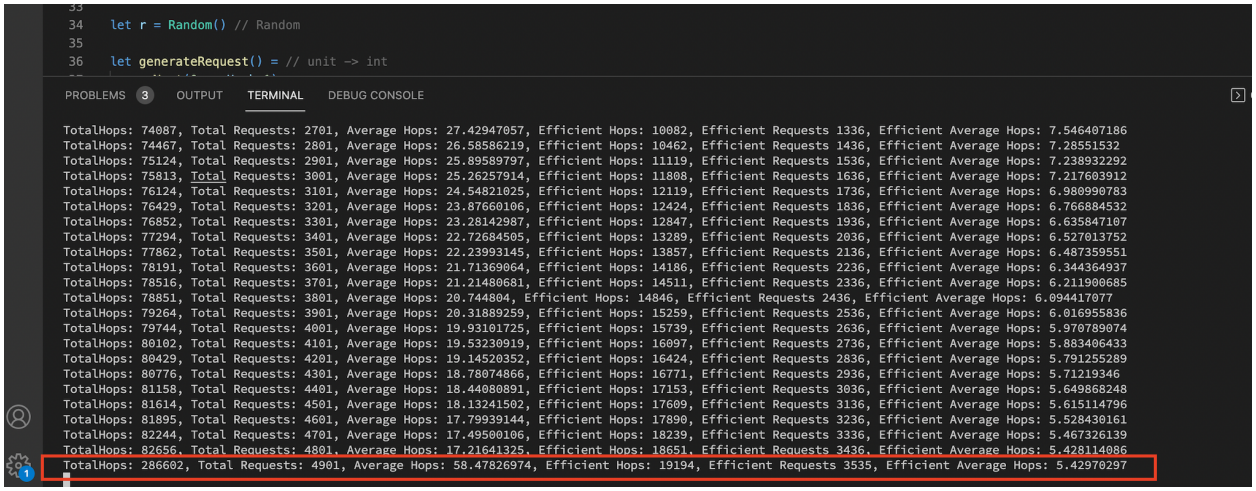


Here efficient requests are defined as the requests which took less than 8 (no. of bits) hops to reach the target.

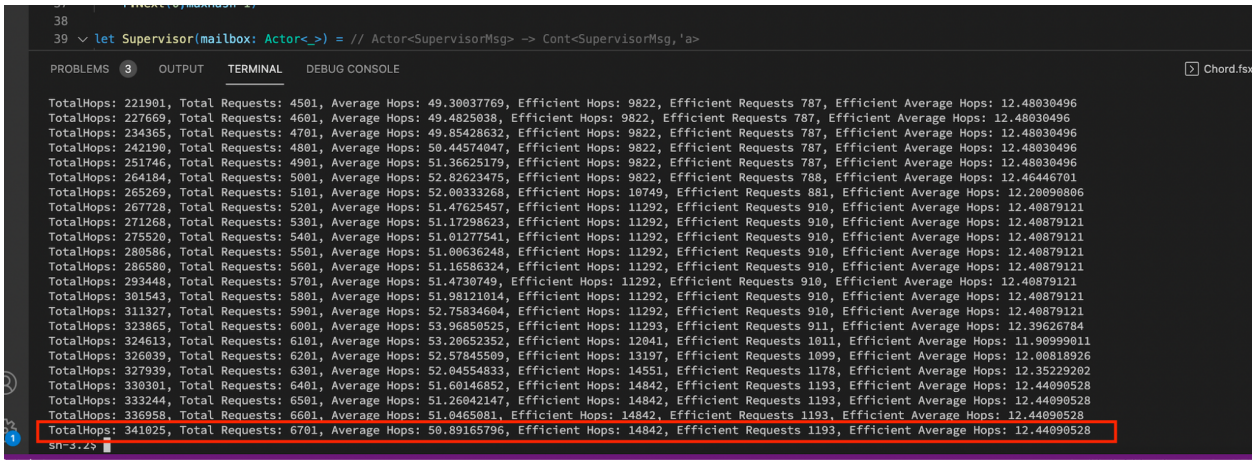
As we can see for this network 713 requests were efficient out of 764 tota requests. The average hops to reach the target for the efficient requests were 3.6

What is the largest network you managed to deal with

For 500 Nodes and 15 bits network avg hops are 5.4



For 1000 Nodes and 20 bit network



Running for 10,000 Nodes gave stack overflow error