**REPORT**

Initially both of us discussed about how we could implement the project and we came up with a design plan. Then we went about discussing about the possible challenges and Data structures we need to use and how we will be implementing them.

Coming to the high level implementation of the project we broke it down into small tasks. We had to deal with the send and receive functionalities for IOT node, Fog node and the cloud node. The other main functionalities include choosing the best neighbor and forwarding the request to the best neighbor with the least queuing delay. Then we learnt about socket programming and how we could implement it. We used socket programming to create sockets for the nodes to send and listen on outgoing and incoming messages respectively.

At the IOT node, we read the arguments from the command line and use sockets to send the request message to the fog node. And we also listen for any incoming responses from the fog nodes and the cloud at the same time. So in order to do this at the same time we had to use multi-threading. We used multithreading to maintain a thread for the send functionality as well as another thread for the receive functionality.

For the Fog Node, we had to use multiple data structures and we faced some challenges in the fog node implementation. We used queues to maintain the requests in order and process them in order. And we used dictionaries to map socket information for a port. We used dictionaries because they enable fast and immediate lookups at O(1) time complexity. We had to deal with multiple send and receive threads as well as the periodic exchange of best neighbor information among fog nodes in the implementation. We maintain different threads for each of them to make everything run in parallel. We maintain four threads for different functionalities – sending and receiving messages from IOT and sending messages to cloud and periodic exchange of queuing delay information.

We used queues to maintain the requests in order and used the information from the message in the queue to look up the latest processing time and forward limit. Similarly for finding the best neighbor we followed similar process by popping from queue and then forwarding the message to the best neighbor and not sending back to the node from which the request came.

The interesting challenge that we faced was implementing threading and also choosing the best neighbor with periodic exchange of messages. We found it difficult to implement it. We somehow figured out a scheme but could not get the results we desired.

Then at the cloud node, the fog nodes as well as the cloud nodes establish connections and cloud node listens for incoming requests to be processed. The cloud node processes the requests and stores them in a queue. We pop out of the queue and process the request and send it back to the IOT node.

**CONTRIBUTION**

We split the work equally among ourselves. One of the member(Hitesh) - took care of the coding for the IOT node and the Cloud Node and the other member (Chaitanya Eswar) helped code the Fog Node. We together discussed and came up with a scheme to store the necessary information and send it over to the nodes for processing. We worked together on implementing the Threads and integrating all the 3 modules together. The testing and debugging also was done together and necessary corrections were made.