

Nikhil Patil

ECE/CS 658: Internet Engineering

Design for the sharing contents in an unstructured peer to peer network.

The main aim of the assignment is to develop the simple unstructured peer to peer solution that will allow sharing of the content among each other. The flow for the program is according to the mentioned protocols. Each node takes the IP address and port number of the bootstrap server and the port number on which this node will work. These parameters are passed from the arguments to the program from the user. The argument parser is used for the getting these values. Node will ask for the user name, at the start which it will use to register with the bootstrap server. When Node will get all necessary parameters and details from the user, it will start the registration process. As given in the protocol the node will create a UDP socket for communication and it will send the REG protocol message to the bootstrap server. REG protocol message will contain the length of message REG keyword, Ip address of the node, port number of the node on which it is working as well the user name.

The bootstrap server will send the register Ok message along with the three IP address of the existing node (if nodes are registered previously). For first registering node the bootstrap will send only Ok acknowledgement, for second node it will send one IP address of previous node, for third node it will send the IP addresses of first two nodes, and from second onwards the bootstrap will send the random 3 IP addresses which are already registered with the bootstrap. The IP and port from the bootstrap are stored in the table. The table class is designed for the maintaining the routing table. After storing the entries in routing table, the routing table is accessed and the JOIN message containing the self IP address and port number is send to each entry in the routing table. Node will receive the join Ok response from that node. Each node will receive some join request from some new nodes appeared in the network. The one thread is created which will always receive all incoming requests and serve them. When join request will come from the other node, the protocol message will contain the IP and port of the requesting node, The Ok response as per protocol is send to that node as well as that IP address and port number is added to the existing routing table. As network grows and more and more nodes join the size of routing table will increase which will increase the per node cost.

When the search request is placed on any node or received on node, that node first search for the resource on its database and if it do not found the resource, that node will forward that request to all

other node which are in the routing table of that node. It can be implemented with different techniques, which are flooding based search or random walk. Each search request is forwarded with the IP address Port number and time stamp of the originating node, also the hop count field is added in this protocol. The request travel from node to node the hop count is increased by one. In flooding based searching to restrict the number of requests the optimum hops count is used. When the hops count reaches the maximum value the request is dropped and it is not forwarded to other nodes. To restrict the duplication of the request at each node the history table is maintained which will contain the IP, Port, resource and time stamp at which request has placed. Every request is compared with the previous request history. The time stamp for each request will be unique. If all these parameters of request are same as previous, it will confirm duplicate request and that request is dropped.

The resources.txt file is used for the query generation, according to zipf's formula the frequencies are calculated, the value of s parameter is taken from the user input. According to rank calculated frequencies the N_s queries are generated and stored in one text file.

The leave message protocol is used to leave node. When user gives the `exit()` command the node will create the leave message as given in the protocol which contain IP address, port number of the leaving node. This message is given to all the IP addresses which are in the routing table of the leaving node. When these IP address receives the message these nodes will delete the IP address of the leaving node from their routing table and update their routing table. After updating the routing table, the nodes send the LEAVE OK message to the leaving node. When node will receive the Leave OK message from all the nodes, it sends the DEL IPADDRESS message to the bootstrap server. When it receives Ok message from the bootstrap server the node will delete the process and exit from the program.

The following parameters are calculated in program. The number of hops required for the query is calculated and displayed. The latency for each query is calculated, the node process time which is delay is calculated.

The `genquery()` on console is used for searching query file generated with the Zipf's distribution.

The any normal input on the console will search for that keyword.

The `exit()` input on console will leave the node from the network.