DESIGN

Swathi Malavika Madhamsetty
Sushmitha Vadlakonda

The main objective of this lab assignment is to develop a simple unstructured P2P network that allows a set of nodes to share contents among each other. Initially, every node needs to register with the Bootstrap server (BS) with their socket address (IP and port number).

The nodes need to specify the user name with which they want to register at bootstrap server. For this, as soon as we run the node, it will ask for a user name to register with which it sends REG message to the Bootstrap server, receives a REGOK response from the Bootstrap server and immediately sends the JOIN messages to the nodes whose socket addresses are in their routing tables (These socket addresses at each node are received from the bootstrap server when the node sends a register message).

The node registering first will receive an acknowledgement from the BS, the second node gets details of the first node, the third gets details of the first two nodes and from the fourth node it gets details of any three previously registered nodes. The nodes will communicate between each other using UDP protocol. The node trying to join other nodes will get a JOINOK response from its peers in the routing table. This forms the network.

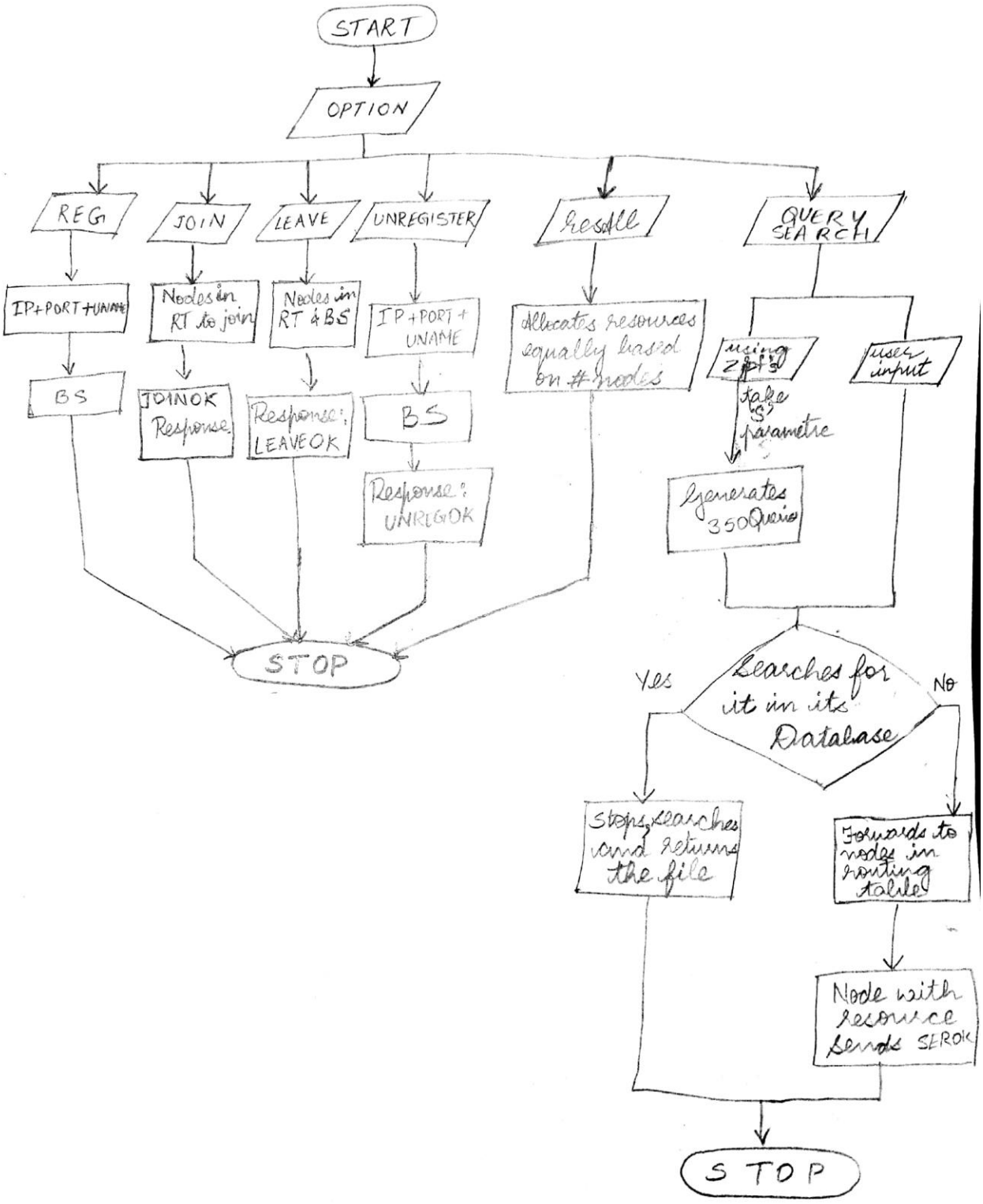
The routing table will increase with the growth in the network. After which we allocate resources based on the number of nodes in the network (as given). This purpose is served by resourceAllocation() method and it happens when we give resAll as a user input. Once the resource allocation is done, the search method is used to find the queries generated.

This program design does query search in two ways. One in which it generates queries using zipf's law and searches accordingly. The other in which we give a filename as a user input and search for it in the resources that a node contains. The first is done using the query() method and the latter is done using findFile() method. The zipf's formula takes s parameter from the user input and generates different queries based on their rank.

When a node receives or sends a query, it first looks for the query search word in its own set of resources and if does not find in its database, it will forward it to the nodes it is joined to. The node which has the resource will send a SEROK message back to the node that initiated the query. The queries are forwarded in the search method using flooding method. A hop count and timetolive is send along with every search message packet. Once the timetolive reaches 0, the packet is killed and not forwarded anymore. The timestamp at each node gives latency and delay.

The leave protocol is done using the LEAVE message as user input. It sends a leave message to all the nodes in its routing table and also to BS. If it happens properly, it receives LEAVEOK from all the nodes in its routing table and BS. The nodes which receives leave message removes that entry in the routing table. The unregister protocol is implemented using UNREGISTER. It sends an unregister message to BS and unregisters itself from the user name. It uses DEL IPLIST message for that. This can either unregister itself or unregister the entire network using the username. exit as a user input stops the program on the node.

The structure is as shown below:


START
OPTION
REG
JOIN
LEAVE
UNREGISTER
Resall
QUERY SEARCH
IP+PORT+UNAME
Nodes in RT to join
Nodes in RT & BS
IP+PORT+ UNAME
Allocates resources equally based on #nodes
using zipf's
user input
BS
JOINOK Response
Response: LEAVEOK
BS
take S parameter
Response: UNREGOK
generates 350 Queries
STOP
Yes
Searches for it in its Database
No
stops, searches and returns the file
Forwards to nodes in routing table
Node with resource sends SEROK
STOP