Projects 3: Protocol Design

Joey Hwang
Yan Yue

**For Discover Protocol:**

Server ___“Discover”___ + ___ServerIP__ + ___ServerPort_______ → ______ Neighbor
← ___”ResponseDiscover”+neighborIP+neighborPort+ user list___
← ___”FailDiscover”____________________________

When a server goes online, it sends out a “Discover” broadcast message, along with its IP address and port. Any online server will receive this message, but will randomly choose whether it will accept this server as a neighbor. If it accepts this server as a neighbor, then it will send a “ResponseDiscover” message back, along with its user list. The user list is in the format of “<alias> <talk receiver address> <talk receiver port> <alias> etc...”.

If it chooses not to be a neighbor, then it will send a “FailDiscover” message.

If all servers send “FailDiscover”, then wait a few minutes and try again.

If this server is the first server online, then no responses will be received. After a timeout period, we can determine that this is the only server online, and therefore, will not have any neighbors.

**For Server Fail Protocol:**

Node ___”Detect”___ + ___NodeIP__ + ___NodePort___________ → ______ Server

A user agent will send heartbeat messages every minute to the server it is connected to. If the agent does not receive a response from its server within thirty seconds, then we can deem the server as being down. The agent will then choose a neighbor from its list of neighbors received from the server and will switch servers.

Node ___“Reattach” + alias + userIP + userPort___________ → ___Another Server
← ___”ResponseReattach”
← ___”FailReattach”____________________________

When a server fails, the user agent will send a “Reattach” message to another server, asking to join that server. Its alias and talk receiver address and port will be send in that message.
**For Server Rejoin Protocol:**

Server ___“Rejoin” + serverIP + serverPort ________ → _______ Neighbor

← __”ResponseRejoin” + alias + userIP + userPort

← __”FailRejoin”

When a server is rejoining the network, it will send a “Rejoin” message to its neighbors. The neighbors will acknowledge that rejoin, and will send a list of the users that was originally on that server. The neighboring servers will then tell those user agents to rejoin their original servers.

**For Node Search another Node:**

Node ___”Search” + User to be searched for + IP + Port ________ → _______ Server

← __”Found” + User IP UserPort + etc....

Neighbor←”ForwardSearch” + agentIP + agentPort + IP + Port + User + Servers

________”Found” + UserIP + UserPort ________ → ________

When a user agent wants to search for an alias, it sends a “Search” message to the server, with the alias to search for, and the talk receiver address and port from the originating user agent.

The server will then send a “ForwardSearch” message to all its neighbors. It will include the alias to search for and the list of servers that have already been visited. This is to prevent infinite loops in searching already searched servers. Also included in the message is its server IP and port, to allow for a response. It will also include the originating user agent’s talk receiver address and port.

When a server receives a “ForwardSearch” message, it will check to see if it exists in the “Servers” part of the message. If it doesn’t, it will proceed to check its user list and see if the alias exists. If it does, it will send a “Found” message, along with the talk receiver address and port for the alias.

When an agent receives a “Found” message, then it will extract the list of IP addresses and ports that were found.