

## I. Summary of New Commands

### i. User Node Commands

1. FIND - This command is sent from a user node to its server requesting that the user node with the specified alias be found.
2. PREFERRED - This command is used to connect a user node and server. Before a server will process commands from a user, it must issue this command telling the server who its preferred server is. A server first checks itself against the IP address and port information and if it is not preferred notes the alias of the preferred server. This semantic allows a user to connect to its preferred server without knowing its alias and if its preferred server comes online after failure at a different location but with the same alias, to find it.

### ii. Server Commands

1. DISCOVER - This command is used by a server only when it first connects to the network to both detect name collision as well as find neighbors.
2. KNOWNUSERS - This command is sent between servers to inform each other about the users who are connected to them.
3. KNOWNSERVERS - This command is sent between servers to inform each other about who their neighbors are.
4. PREFERRED - This command is sent to a server to get a list of user nodes that prefer the requesting server over the receiving server.
5. JOINCHANNEL - This command is used so a server may request another server join a different channel with it. This allows any network topology to be formed.

## II. Protocol Description

### i. Server Discovery Protocol

When a server starts up, it binds itself to a listener socket to receive communiqués from user nodes. It should also begin listening on 230.0.0.1:8450; the multicast channel that is used for server communication. It then begins receiving broadcasts from that channel and sends its own broadcast of the DISCOVER Command until it has found its neighbors or changed its alias. If a server receives a DISCOVER command and is either in conflict or knows a server with a conflict of that name, a ServerReply error is returned. A ServerReply of success is returned if the server wishes to neighbor with the new server. There may be additional channels used so that a server can be discoverable by a subset of servers and not all live servers. A channel is uniquely determined by a multicast address and port, and servers may belong to one or more channel.

Server communication is now a combination of multicast and unicast. After the DISCOVER command is sent and a neighbor server replies, the ServerReply

includes the unicast listener information so future commands that are not meant for all neighbors, such as JOINCHANNEL, may be sent directly from one server to another. Other commands such as KNOWNUSERS, KNOWNSERVERS and PREFERRED which are meant for all neighbors of a server may be multicast to all channels that a server belongs to.

ii. Create New Server Group

At some point, a Server may wish to create a "private" communication channel between servers. This Server would send a JOINCHANNEL command to its neighbors who it wishes to invite. These neighbors would then begin listening on the new multicast channel. They would have the option to continue listening on the original channel or disconnecting from it.

iii. Fault Discovery Protocol

After N occurrences of a user node sending a request to its current server and receiving no reply within a pre-specified timeout period, it assumes the server has failed and proceeds to attempt to connect to a new server based successively on its current list of the failed server's neighbors. Here N may be arbitrarily set, greater than zero, based on a process's tolerance for failure. Once connected, it sends the PREFERRED command with the information about the previous server so that the system knows to move it back to its preferred server in the event the failed server comes back online. If there is no available backup server, the user node exits.

iv. User node discovery of neighbor servers

When a server wishes to tell its attached nodes about its neighboring servers, it sends a KNOWNSERVERS Command to the UserNode with the list of known Servers.

v. User node reacquisition after failure recovery

Since the server saves the current neighbor list to non-volatile storage, once it comes online after a fault it will contact its old neighbors with the PREFERRED command. The servers will return a ServerReply listing all UserNodes that prefer the recovered server who are currently under their jurisdiction. Once the recovered server receives the ServerReply it contacts each UserNode with the same preferred Command to let them know their preferred server is online again. The UserNode then disconnects, by way of an EXIT command, from its backup Server and reconnects with its preferred Server.

vi. Finding a User node

A User node queries its server with a Find command. If a server can fulfill the request and has not previously responded to this request, it replies directly to the UserNode, otherwise it forwards the request to all neighboring servers. The ServerReply includes both the Server information as well as the found UserNode.

If a user receives multiple replies from different servers regarding its FIND request, it is left to the user to decide which reply to favor.

### III. Message Definitions

For this part of the project, we have defined a grammar for all messages passed from server to server, as well as all messages between the user agent and a server. At the top level of the grammar are two entities: Command and ServerReply, which together comprise all messages sent through the system.

**Command**      A command is either from a server or a user. The first field denotes the sender's reply address and the second field denotes the command they are sending. The recipient is not included in the command; rather it is determined by packet destination.

Server This is the information describing a server, specifically its globally unique alias and the location of the receiver thread.

SCommand This is used for commands initiated from servers.

- A DISCOVER command was added for a new server to let others discover it. If a server receives a DISCOVER command with an alias it already knows, a ServerReply error is sent back to the server to inform it about the name collision. Otherwise a ServerReply is sent with the receiving servers Server information for consideration as a neighbor.
- A KNOWNUSERS command lets a server tell others information about the users nodes that are currently attached to the server.
- A KNOWNSERVERS command lets a server tell other information about its neighboring servers.
- A PREFERRED command is used to either ask another server if it has any user nodes that prefer the requesting server over the receiving server or to let a user node know that its preferred server is available
- A JOINCHANNEL command is issued when a server wishes to form a private channel with other servers. Any server may issues this command to any other server but the recipient must ServerReply either +OK or -ERR to let the requesting server know its intentions.

UserNode This describes a user node's reply address, i.e. talk receiver thread

UCommand This is used for commands initiated from a user node.

- A FIND command is used to ask for the location of a specific alias.
- A PREFERRED command tells the backup Server which server is the user node's preferred server

ServerReply This is for a server to reply to a request. It begins with the server replying and the status of the reply. Next is the command it is replying to (usually verbatim from the request for simplicity). Last is a variable list of server information, user node information and messages to include with the reply.

Status +OK denotes success and -ERR failure

Alias Allow any alphanumeric combination of at least 1 character

IPAddress An IP Address (v4) consists of four bytes separated by periods

Byte Allows for a decimal representation of a single byte

Port Require 4 digit ports to encourage port >= 1000 since leading zeros are allowed

RoomName Allow any alphanumeric combination of at least 1 character

Message Allow a message to be a null terminated string of any non-null characters.

#### IV. Message Grammar

```
Command      => [Server] [SCommand]
              [UserNode] [UCommand]

Server       => SERVER [Alias] [IPAddress] [Port]

SCommand     => DISCOVER [Alias]
              KNOWNUSERS [UserNode]*
              KNOWNSERVERS [Server]*
              PREFERRED [Server]
              JOINCHANNEL [IPAddress] [Port]

UserNode     => USERNODE [Alias] [IPAddress] [Port]

UCommand     => FIND [Alias]
              PREFERRED [Server]

              CREATEROOM [RoomName]
              JOINROOM [RoomName]
              LEAVEROOM [RoomName]
              ULIST [RoomName]
              LISTROOMS
              CSEND [Message]

              EXIT
              ALIAS [UserNode]
              REMOVE [Alias]
              SEND [Alias] [Message]
              LIST
              SETALIAS [UserNode]

ServerReply  => SERVERREPLY [Server] [Status] [Command] [Server]* [UserNode]*
              [Message]*

Status       => +OK
              -ERR

Alias        => [A-Za-z0-9]+

IPAddress    => [Byte].[Byte].[Byte].[Byte]

Byte         => [0-2]?[0-9]?[0-9]

Port         => [0-6]?[0-9][0-9][0-9][0-9]

RoomName     => [A-Za-z0-9]+

Message      => MESSAGE [^\0]*[\0]
```