**Internet Relay Chat**

**Abstract**

This Internet Relay Chat application is communication tool for multiple users to communicate via messages in the chat rooms. It’s a chat system involving set of rules based on IRC Protocols involving client and server python scripts. This document includes the specification of IRC protocols involved in developing this project.

**Table of Contents**

1. Introduction

i) Server

ii) Client

iii) Discussion Room

2. Specification

i) Mode of Communication

ii) Character Codes

iii) Messages

iv) Replies

3. Message Infrastructure

i) Create room

ii) Join room

iii) Leave room

iv) Room List

4. IRC Concepts

5. Error and Crash Handling

6. Conclusion & Future Work

7. References

**1. Introduction**

This specification explains a simple Internet Relay Chat (IRC) protocol where the clients can communicate with each other. This system employs a central server which relays messages that are sent to it to other connected users. Users can join rooms, which are groups of users that are subscribed to the same message stream. Any message sent to that room is forwarded to all users currently joined to that room. Users can also send private messages directly to other users.

i) Server

The traditional IRC protocol, documented in RFC 1459, supports multiple servers in a tree configuration. However, for the purpose of this project I am limiting the number of servers to one. This single server will act as the backbone of the IRC network, providing a unique point for clients to connect to and talk to one another.

ii) Client

A client is anything that connects to the single server. Each client is identified by a unique nickname. Typically, there is a human user at the client’s end, so going forward we will use the terms User & Client interchangeably.

iii) Discussion Room

A room is a named group of one or more clients, which will all receive messages addressed to that room. A room is implicitly created by the first user who joins the channel. It ceases to exist when all users leave the channel. While a room exists, anyone can reference it by its name. A user can join a room by issuing a JOIN command. If the room doesn’t exist, a new room is created, and the user is added to that room. Once created, a room is open to any user who wishes to join it. Since the IRC network only has a single server, in case the server crashes, all rooms cease to exist.

**2. Specification**

i) Mode of Communication

In this version of the IRC app, the only support one to many communications via rooms. The room works as a dynamic multicast group. To communicate, a user sends a message to the server addressed to the room name and the server forwards it to all users that are part of that room, except the sender itself.

ii) Character Codes

The protocol uses the ASCII character set. Each message may be composed of any number of these characters.

iii) Messages

Each IRC message contains three parts: the name of the command, a transaction ID, and the payload. Each part is separated by a single ASCII SPACE character (the delimiter). The transaction ID is an unsigned integer, starting at 0 and incremented by 1 for each message sent by the client. We envision that this ID could be used for debugging purposes on the server, by tracing a client’s steps. Server to client messages will always use 0 instead of an incrementing number.

iv) Replies

Most of the messages sent to the server generate a reply of some sort. The most common reply is the numeric reply, used for both errors and normal replies. The numeric reply consists of the transaction ID of the request followed by the status of the request, 0 for a successful request and non-zero for errors. A numeric reply is not allowed to originate from a client; any such messages received by a server will generate an error response. In all other respects, a numeric reply is just like a normal message. A list of different error codes is provided in the next section.

**3. Message Infrastructure**

i) Create room

When a user starts this application and there is no discussion room open to join in, it has the option instead to create a new discussion room and wait for others to join in.

ii) Join room

When a user starts the application, and if there is a discussion room available, it can join into that room and start sending messages to the members of that room.

iii) Leave room

At any moment of time, a user can leave a room it wants to end the discussion for itself. However, the group will still exist as there are other members still joined into it.

**4. IRC Concepts**

i) One-to-one communication

When In this type of communication, a host talks to one and only one other host. A client may talk to another client in private or talk to the server to request a service. Therefore, client will directly send a message to the server, which in turn will send it directly to the client for whom the message was directed.

a) A client can talk to another client in private. This chat will not be visible to anybody else even if the sender and receiver are a part of a room.

b) A client can talk to the server to request any kind of service like to create, join or leave a discussion room.

ii) One-to-many communication

A client can talk to many other multiple clients and send messages to them. This is possible when a user joins or creates a room and post a message on that discussion room. Then that message will be viewed by all the other members of that room.

iii) One-to-all communication

A client can send a broadcast message that will be delivered to each and every client and server.

**5. Error and Crash Handling**

i) A client cannot communicate with a client which does not exist.

ii) If a client is not a part of a chat room, then it cannot send message to that chat room.

iii) When the server crashes, the clients are logged out of the service and will get the message, “Sorry the server is down.”

iv) If a client gives a join command on a room that does not exist, a room with that name will be created and the user will be added to that room.

v) When a client crashes, the server as well as the room and the clients present in that room will be notified.

vi) A client should not be able to connect to the chat room again after it is already a part of that chat room. If he tries, ‘You are already a member’ message is displayed.

vii) If a user tries to leave a room which he has not joined, ‘You are not a member of the group’ message will be displayed to the user.

**6. Conclusion & Future Work**

The framework for interactions and message passing between single server and client has been extended for multiple clients to communicate with each other via broadcast messages or private messages. The specification has been designed for interactions within the virtual rooms which are implemented. The interactions include protocols for text-passing and can be extended to images. It can be used to transfer large files of data in a secure connection using cryptographic transport protocols. Further, Transport Layer Security (TLS) protocols and Secure socket connection are exploited. There are also some problems associated with this protocol, that will probably be resolved in future.

**7. References:**

i) <https://en.wikipedia.org/wiki/Internet_Relay_Chat>

ii) <https://pythonspot.com/building-an-irc-bot/>

iii) <http://www.networksorcery.com/enp/rfc/rfc1459.txt>

iv) <https://modern.ircdocs.horse/>

v) https://github.com/xysun/pychat/

vi) <https://github.com/bl4de/irc-client>/