INTERNET RELAY CHAT

Table of Contents

1. Introduction

This is a simple Internet Relay chat (IRC) application used by clients to do a text based conferencing i.e. client will communicate among each other by passing the text messages. This application follows client server model. The set up of the application involves central server to which the clients will connect to and it will be responsible for all the communication. Clients can join the conference (one to many) or do the one to one chat, in a conference there will be group of clients which are associated with the same message stream. Any message sent to that conference is forwarded to all clients currently joined to that conference. The IRC provides a reliable way of communicating among the clients by following a connection oriented approach and thus it follows the TCP protocol.

1.1 Server

The server being the central part of the application, can connect or link rest of the components together. It provides all the basic functionalities described in the protocol of the application like- storing the information about the clients participating in the communication, delivering the messages reliably. For the IRC to perform properly the server should always be available and running and that's why it is considered as the central part of the application.

1.2 Client

A client is one who wants to communicate with other people using the text messages. Each client is identified by its unique name. They may communicate with other participating clients via Conferences or private chats. The Client must sign on the server by providing its details which server can use to pass its messages to other clients and then passing back the replies.

1.3 Conference

A Conference is a group of more than one clients connected to the server with the same message stream. Multiple Clients can communicate in the Conference. The Conference is created by one client and then other clients can view it and join it. A client broadcasts message to all the clients participating in the Conference. The Conference must contain at least one client. The Conference ceases to exit when the last client leaves it. The Conference has a unique name and it can be referenced by that name. The Conference will receive all the messages which are addressed to it.

2. IRC Specifications

These specify the protocol details to be followed.

2.1 Character Code

The clients must follow the command syntax while communicating in the application as there is no character code for it.

2.2 Message

In this client server based application all the communication is done via text messages which are sent in a particular format. When a message is sent a response should be expected but the sender should not wait forever for the response as sometimes connection may be lost or the receiver is busy.

2.3 Message Format

Messages are of <Command> <Argument> format.

‘Command’ specifies the service name the client is asking from the server.

Eg: join/leave Conference1

Command - Conference

Argument - create/join/disconnect, Conference1

In this example join/leave is the command and the client is requesting server for Conference related service.

‘Argument’ allows the server to provide a very specific service for the command specified. A command may have one or more arguments.

In above example client can create or can join or can disconnect the conference.

3. IRC Concepts

3.1 One-to-one communication

In this type of communication a host talks to only one other host. A client may talk to another client in private or talk to the server to request a service. So in this case client will directly send message to the server which in turn will send it directly to the client intended for that message.

3.1.1 Client-to-client

A client can privately talk to another client. Even if the clients are participating in the same Conference, their private conversation won’t be seen by other clients in the Conference.

3.1.2 Client-to-server

A client can talk to server to request a service like connect/join to Conference, create a Conference, disconnect etc.

3.2 One-to-many communication

A client can talk to multiple other clients in one-to-many communication. This is possible via Conference. In a Conference when a client posts a message, it can be read by other clients in the Conference.

3.3 One-To-All

This a basically a broadcast message to all the clients and the server.

4. Message details

The following format for command messages must be followed:

4.1 Server connection format

server listening at port

this message will be displayed once server is up after launching the application server.

4.2 Server reply format

Connected/Disconnected/listening <user name>.

<username> it is a unique name given to each client by the server, when it connects to the server.

4.3 Server shutdown format

Shutdown or closing the server

This will shutdown the server. Before shutting down, it will notify and close all the clients and Conferences.

4.4 Client message format

<personal> <user name>

This command allows a client to send message to another client specified by '<user name>'.

<user name> is name for different client and <message> is messages to be sent to other client.

4.5 Conference connection format

<join> <Conference name> :This command allows user to create or join a Conference.

<list>: It allows the client to see the list to Conferences currently active.

<leave> <conference name> : leave a Conference specified by '<Conference name>'.

<switch> <confernece name> : to switch between the conferences.

<manual>: to see all the comands available

<quit>:to exit completely.

5. Other details

5.1 Error handling

1. A client cannot join a conference which does not exist.

2. A client cannot communicate with a client which does not exist.

3. A client cannot message or join a Conference which does not exist.

4. A client should not be able to connect to the Conference again after it is already a part of that Conference.

5. If a client is not part of a Conference, then it cannot send message to that Conference.

5.2 Terminating Connection

1. A Conference ceases to exit when all its clients leave the Conference.

2. If the server abruptly shuts down, then all the clients get disconnected and the message will be displayed "Server is down".

3. If a client is part of one or more Conferences, and sends a disconnect request to server, then it is first disconnected from all the Conferences, and then disconnected from the server.

6. Conclusion & Future Work

This specification provides a generic message passing framework for multiple clients to communicate with each other via a central forwarding server. Without any modifications to this specification, it is possible for clients to devise their own protocols that rely on the text-passing system described here. For example,

transfer of arbitrary binary data can be achieved through transcoding to base64. Such infrastructure could be used to transfer arbitrarily large files, or to establish secure connections using cryptographic transport protocols such as Transport Layer Security (TLS).

7. Security Considerations

Messages sent using this system have no protection against inspection, tampering or outright forgery. The server sees all messages that are sent through the use of this service. 'Private' messaging may be easily intercepted by a 3rd party that is able to capture network traffic. Users wishing to use this system for secure communication should use/implement their own user-to-user encryption protocol.

8. IANA Considerations

None

9. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

https://tools.ietf.org/html/rfc2813

10. Acknowledgments

This document was prepared using word 2016(2-Word-v2.0.template.dot)

Authors' Addresses

FNU Akanksha
Portland State University Computer Science
1612 SE 168 AVE Vancouver WA 98683
Email: akanksha@pdx.edu