Portland State University | Andrew Capatina
Intended Status: IRC Project Specifications | CS594
Expires: June 2016 | May 2018

Internet Relay Chat Protocol

Abstract

This document defines the protocol for the Internet Relay Chat program. The Internet Relay Chat program will use a client server model to communicate. The transport protocol layer interface used will be the Transport Control Protocol. The purpose of Internet Relay Chat is to give users an interface to chat among each other in real time. Users will have the ability to create a room and join any available room where messages sent are shown. Since the users services will be based off of the server, a protocol is needed to ensure pleasant and efficient user experiences. Rooms and members of rooms will need to be tracked to ensure clients receive the correct message. Along with a protocol, metadata needs to be taken into consideration for how the server handles incoming packets. The metadata needed to ensure each user is properly serviced shall be discussed. Without metadata, the server will not know how to interpret the payload of each message.

Table of Contents

1. Introduction………………………………………………………………………………2
   1.1 Servers…………………………………………………………………………...2
   1.2 Clients……………………………………………………………………………..2
   1.3 Chat rooms………………………………………………………………………..2
   1.4 Transport service……………………………………………………………...2
2. IRC Concepts……………………………………………………………………………3
   2.1 One-to-one communication……………………………………………………..3
   2.2 One-to-many communication…………………………………………………...3
   2.3 One-to-all communication………………………………………………………3
3. Message Details…………………………………………………………………………3
   3.1 Username message……………………………………………………………...3
   3.2 User commands…...……………………………………………………………3
   3.3 Server message………………………………………………………………4
   3.4 Chat room operations…………………………………………………………....4
   3.5 Client quit……………………………………………………………………….4
   3.6 Sending messages………………………………………………………………5
   3.7 Server shutdown………………………………………………………………5
   3.8 Ping message……………………………………………………………………5
4. Conclusion……………………………………………………………………………..5
5. Security Considerations…………………………………………………………….....5
6. IANA Considerations……………………………………………………………………6

1. Introduction

1.1 Servers

The server will be the point where clients are able to connect together. Clients will send their message to the server then it will be the servers job to direct the message. The server must be able to track peers so that messages may be directed to the correct peer. The server will keep track of the active rooms for the irc application. Chat room and client data will be stored in seperate data structures. The server will maintain connections for the duration of the user's session.

1.2 Clients

Clients are the users in the IRC application attempting to send messages. Clients are distinguished by socket number and user name and are allowed to enter a username for their session. IP address will be stored as well to ensure there is an absolute way of distinguishing a user. The client will be allowed to have any kind of username. As stated in the prior paragraph, a number of information is needed to ensure messages are sent to the correct locations. The details will be discussed in the client specifications. All client information is deallocated when the user logs off.

1.3 Chat rooms

Chat rooms give clients the ability to connect with each other via the server. Once a client joins a room, they have the ability to send messages amongst others joined in the group. The server will be responsible for allocating a chat room and keeping track of its members.

Chat rooms are created by users. The server will cap the number of chat rooms that may be opened at 25 users. Users are allowed to join and switch chat rooms. To switch rooms, the user must be a member of the requested room. There will be no history of chat room conversations, therefore new users will not see what was previously typed when joining a room. The maximum number of chat rooms will be 10.

1.4 Transport service

Prior to running the application, a network connection needs to be established. Since the connection between client and server remains for the duration of the users session, TCP is being used. After handshaking, the TCP connections stays open as needed. To have the server handle multiple clients, the server will essentially allocate multiple sockets to a port. A descriptor will be used to track a peer to a socket. This will allow the server to distinguish peers and ensure the peers receives the intended information.

2. IRC Concepts

This sections describes the basic concepts/components that make up the IRC application. The reason for implementing the various concepts will be briefly discussed.

2.1 One-to-one communication

One-to-one communication refers to exactly two devices communicating. The server and each client will maintain its own TCP connection with the server. One to one communication is needed for sending acknowledgements to clients making requests. It is also essential for the client, who needs to send messages to the server.

On the IRC application level, the ability to send private messages will be available to users. The users will have to be members of the same room to send private messages amongst each other before sending a private message.

2.2 One-to-many communication

One-to-many communication refers to the ability to communicate with a subset of all peers joined on the server. The point of RFC is to essentially provide one-to-many communication because of the chat room concept. Keeping track of unique peer information is key to ensuring this communication method is possible. A linear linked list will be used to keep track of individual chat rooms as well.

2.3 One-to-all communication

One-to-all communication refers to the ability to communicate with all the clients connected to the server. This communication mean will be needed when the server needs to shut down and close all connections.

3. Message Details

3.1 Username message

Upon connecting, the client must enter a username that is 16 characters or less. Once a name is entered, the server will send a code number back as an acknowledgement. The client can not use any of the operations until they have registered.

3.2 User Commands

The client will use a format with two distinct fields of information, each segment containing the delimiter "/". The format will allow the client to issue commands to the server, and messages to be displayed in the chat. The purpose of these commands allows the user to register a username, create a room, list all rooms, list currently joined rooms, send messages publicly and privately, leave a room, and switch rooms. The client can also disconnect from the server, and request menu options. The message buffer will have a maximum size which the user will not be able to exceed. The goal however, is to keep user message length limited. The chat is intended for quick messages and statements. Upon selecting a command, the application will guide the user through entering the appropriate information for the selected command.

Client General Operation Message Format
<command operation/parameter 1/ parameter 2/ parameter 3/ parameter N/>

Client ACK message Format
<Acknowledge code/>

3.3 Server message

A message format is required such that users understand who sent a message. A format is needed for public and private messages sent on the server. The format for both will be the same, however the only difference is the operation code. The user does not see this change when creating the message.

Server Chat Messages:
<operation/room name/sender username/message>

Lastly, another message format is needed which indicates failure or success for requests between the client and server. Actions that require an acknowledgement from the server are: failure to send a message, create, join, or switch rooms;

Server ACK messages:
<success or fail code/>

3.4 Chat room operations

When a user is connected to a chat room, the user may switch rooms, leave the room, and send a message to the entire chat room, or privately to a member of the chat room. To disconnect, the client must leave the room first. Once a peer leaves, the server broadcast who left to all chat room members.

3.5 Client quit

Upon disconnecting from the server, the clients information must be removed from the server. Their ID must not be present in any of the chat rooms from that point on as well.

3.6 Sending messages

As mentioned, the user will be able to send public and private messages in a chat room. The user needs to specify the appropriate option for sending a public or private message. Once the option is provided, the client application will guide the user in creating a message for the server.

3.7 Server shutdown

When the server needs to shutdown, a message is broadcasted to all clients stating that the server needs to shutdown. After a specified time interval, the server will shutdown. The client application will close the connection as well.

3.8 ACK message

To ensure application robustness by both the client and server, acknowledgements must be given by the server.. On any send function call, an acknowledgement will be returned by the server.

4. Conclusion

In essence, the protocol presented is similar to what we have been learning so far in class. A system sends a request, then the receiving system acknowledges and fulfills the request. That is essentially what is being implemented. For every user action, there has to be some result that lets the user know the application is working; if not, an error code is given.

5. Security Considerations

Server information in regard to its clients, shall be allocated and deallocated based upon active connections. When a peer is assigned a name, the name is only associated with that peer for the duration of the session. Upon leaving the session, the peers information is no longer stored.

Client messages are not altered in any form when being transmitted over the network. This leaves the risk of intruders being able to read messages being sent in chat rooms.

The method used to delimit functions may be dangerous as well, therefore there may be some limitations to user input.

6. IANA Considerations

None.