

CS594 - INTERNETWORKING PROTOCOLS

Name : HARINI G T

Portland State University

Intended status: IRC Class Project Specification

Status of this Memo

This memo describes a feasible implementation of Internet Relay Chat Protocol for a client/server system. The protocol is implemented from scratch. This document may be modified in future. Discussions and suggestions for welcomed.

Abstract

This memo describes the communication protocol for an IRC-style client/server system for the Internetworking Protocols class at Portland State University.

TABLE OF CONTENTS:

- 1. Introduction**
- 2. IRC Concepts**
- 3. Components**
 - 2.1 Server**
 - 2.2 Client**
 - 2.3 Chatroom**
- 4. Functionalities and usage**
- 5. Error handling**
- 6. Extra features**
- 7. Future Enhancements**
- 8. Security considerations**
- 9. IANA considerations**

1.Introduction:

The Internet Relay chat (IRC) is a simple application used by clients for text based chatting. Clients on different workstations will communicate through Text messages. The application is implemented using a Client-server model. Clients shall connect to a single central server and all the communication take place through the central server. Clients can join one or more chat rooms simultaneously. Messages sent by a client in a chat room would be sent to all the clients in that chat room unless it a personal chat with another client. The reliable communication in IRC is implemented using TCP/IP protocol.

2. IRC concept:

This section describes the actual concepts behind the organization of the IRC protocol and how the current implementations deliver different classes of messages.

2.1 One to One Communication:

Communication on a one-to-one basis is usually only performed by clients. A client can also send message to the server and server also sends data to the client.

To provide a secure means for clients to talk to each other, it is required that all servers be able to send a message in exactly one direction in order to reach any client.

2.2 One-to-many:

The main goal of IRC is to provide a forum which allows easy and efficient conferencing (one to many conversations). IRC offers several means to achieve this, each serving its own purpose.

2.2.1 To a group (room)

In IRC the room has a role equivalent to that of the multicast group; their existence is dynamic (coming and going as people join and leave room) and the actual conversation carried out on a room is only sent to servers which are supporting users on a given room. If there

are multiple users on a server in the same room, the message text is sent only once to that server and then sent to each client in the room.

3. Components

3.1 Server

Server in the IRC application provides all other components with the necessary connection links that allow them to work together (Send/Receive messages). Server provides all the functionalities that are necessary for application to work properly such as storing the information about the clients, delivering the messages to clients efficiently and reliably. The important requirement from the server is for it to be available at all the times.

3.2 Client

A client is the one who wants to send the message to group of people or an individual. Each client has Name which is unique and used to identify the client. Clients have an option to have a personal chat with another client or a group chat with a bunch of clients. The Clients have to register on the server before initiating any kind of chat. The server will use these details to send and receive the message to and from the client.

3.3 Chat Room:

The chat room is the component of the system where all the clients connect to the system and communicate with each other. The chatroom is a message channel shared by client to exchange messages. All clients in a chat room communicate with each other. They are not allowed to send messages to a different room unless they are part of it. A client can be part of multiple chat rooms. A client of the application can use all the functionalities offered by the applications.

4. Functionalities

The IRC application would provide the clients with various functionalities such as:

4.1 Entering a chat room: [enter] room_name

A client adds himself to the application by using the command [enter] room_name where room_name specifies the name of the room client wants to join. The above command also creates the chat room when the application starts. This command can be used by the clients of the application only.

4.2: List all the rooms in the application: [List Rooms]

The above command is used to list all the chat rooms in the IRC application. The clients use this command when they want to view the rooms of the IRC application

4.3: List clients in the application : [List Clients]

To view all the clients in the application the above command is used. The list has all the clients in the application along with the chat room the client is a member of. This command helps the clients to view the other clients and other chat rooms in the application.

4.4: Send message to a specific room : [Message] Room_name msg

A client can send messages to a specific chat room in the application. The room_name specifies the room the client wants to send message to. This command enables the client to send messages to multiple chat rooms and toggle between the rooms. Thus the client is able to communicate in multiple chat rooms.

4.5: Exit from the chat room : [Exit] room_name

The client can exit from a chat room specified by room_name using the command [exit] room_name.

4.6: Exit the IRC application : [Quit]

To exit from the IRC application the command [quit] is used. This command removes the client from all the chat rooms the client is part of and enables for a smooth exit from the application.

5. Error Handling :

The various errors and abnormalities that would appear in the application would be handled with the error handling mechanisms such as :

The client would be intimated with appropriate messages during the time client/server clashes and there would a smooth exit from the application.

All the commands in the application follow the pattern
[command] arguments.

The client can have any name/nickname that starts with anything except "[".

The chat rooms in the application start with the word "room".

A client would be able to send messages in the room he/she is a member of.

The presence of the client and the room is always checked at the time of communication initiation.

At every stage appropriate error messages is displayed and a reliable and smooth chat experience is provided to all the clients in the system.

6 Extra features:

The client is allowed to exchange messages privately with another client in the application using the command `[private] client_name`.

7 Future Enhancements:

This Implementation provides a message passing model for multiple clients to communicate with each other via a centrally located 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 in this document. This could be extended to transfer arbitrarily large files, or to establish secure connections using cryptographic transport protocols such as Transport Layer Security(TLS).The current implementation doesn't have a well designed GUI.

8. Security Considerations:

Messages sent using this system have no protection from tampering or outright forgery. The server sees all messages that are sent through the use of this service. Users wishing to use this system for secure communication should use/implement their own user-to-user encryption protocol.

9. IANA Considerations

None

