

CS594 Armaan Roshani

Internet-Draft - work in progress

Portland State University

Intended status: IRC Class Project Specification

May 17, 2018

Expires: November 17, 2018

Internet Relay Chat Class Project
RFC_rev_1.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on November 17, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 |
| 2. Conventions used in this document..... | 2 |
| 3. Basic Information..... | 3 |
| 4. Message Infrastructure..... | 3 |
| 4.1. Generic Message Format..... | 3 |
| 4.2. Error Messages..... | 3 |
| 4.3. Keepalive Messages..... | 3 |
| 5. Label Semantics..... | 3 |
| 6. Client Messages..... | 4 |
| 7. Server Messages..... | 4 |
| 8. Error Handling..... | 4 |
| 9. "Extra" Features Supported..... | 4 |
| 10. Conclusion & Future Work..... | 4 |
| 11. Security Considerations..... | 4 |
| 12. IANA Considerations..... | 5 |
| 12.1. Normative References..... | 5 |
| 13. Acknowledgments..... | 5 |

1. Introduction

This specification describes a simple Internet Relay Chat (IRC) protocol by which 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.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

In this document, the characters ">>" preceding an indented line(s) indicates a statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the portions of this RFC covered by these keywords.

3. Basic Information

All communication described in this protocol takes place over TCP/IP, with the server listening for connections on port 7734. Clients connect to this port and maintain this persistent connection to the server. The client can send messages and requests to the server over this open channel, and the server can reply via the same. This messaging protocol is inherently asynchronous - the client is free to send messages to the server at any time, and the server may asynchronously send messages back to the client.

As is described in [4.2], both the server and client may terminate the connection at any time for any reason. They MAY choose to send an error message to the other party informing them of the reason for connection termination.

The server MAY choose to allow only a finite number of users and rooms, depending on the implementation and resources of the host system. Error codes are available to notify connecting clients that there is currently a high volume of users or groups accessing the server.

4. Message Infrastructure

4.1. Generic Message Format

The format of Generic Messages has yet to be determined.

4.2. Error Messages

The format of Error Messages has yet to be determined, except to stipulate that both the server and client may terminate their connection at any time for any reason. Both must be "robust" in that they must gracefully handle the other party terminating the connection.

4.3. Keepalive Messages

The format of Keepalive Messages has yet to be determined.

5. Label Semantics

Identifying both users and rooms involves sending and receiving labels. All label rules are the same, and MUST be validated as follows:

- o Field size for transmission is yet to be determined.
- o Must consist entirely of readable ASCII character values, between 0x20 and 0x7E.

- o Must be at least 1 character, and at most a yet to be determined number of characters.
- o Cannot start or end with a space.
- o If any of these rules are broken, the receiver MUST terminate the connection and MAY provide a yet to be determined error.

6. Client Messages

The format of Client Messages has yet to be determined.

7. Server Messages

The format of Server Messages has yet to be determined.

8. Error Handling

Both server and client MUST detect when the socket connection linking them is terminated, either when actively sending traffic or by keeping track of the heartbeat messages. If the server detects that the client connection has been lost, the server MUST remove the client from all rooms to which they are joined. If the client detects that the connection to the server has been lost, it MUST consider itself disconnected and MAY choose to reconnect.

As stated previously, it is optional for one party to notify the other in the event of an error.

9. "Extra" Features Supported

Attempts will be made to implement private messages.

10. 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 the (yet to be determined) format that messages will use. 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).

11. 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.

12. IANA Considerations

None

12.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

13. Acknowledgments

This document was prepared using CS594SampleRFC.pdf as a template.