Apprenticeship Network Programming Group
Request for Comments: 9912
Category: Experimental

Y. Rameau
M. Bouazza
UGE

02/28/2021

ChatOS Protocol : Anonymized direct TCP chat service

Abstract

This note describes the ChatOs Protocol (COP), a protocol used in ChatOS Application to route messages in TCP. COP can provides private connexion between two users or a public connexion to send messages for all users.

Status of this Memo

Copyright Notice

Summary

The ChatOs protocol is a very simple protocol used to communicate with other users. This document describes the protocol and its types of packets. The document also explains the reasons behind some of the design decisions.

Table of Contents

1. Purpose

The ChatOs Protocol is intended to be used for the transmission of messages between clients. It may also be used for the local message system of a network or host.

The focus here is on the internal mechanisms to transmit messages, rather than the external interface to users. These will be both new programs designed to work with this system and old programs designed to work with earlier systems.

2. Overview of the protocol

2.1. Clients

A client is defined in the eyes of the server by a nickname and an IP address. The nickname is chosen by the user during his first connection, it must not be larger than the maximum size imposed by the server and must not be offensive or rude. Each user has his own nickname and must therefore be unique for better authentication.

2.2. Client authentication

Clients are subject to a level of authentication. An nickname lookup (and reverse check on this) is performed for all connections made to the server in order to avoid collisions with other nicknames which can lead to authentication difficulties. Users are currently not subjected to a password check. Even if these checks are possible on all connections although the password check is currently not being set by the protocol.

2.3. Message

Server and clients send each other messages wich may or may not generate a reply.

When a server receive a message, it MUST identify its source using (eventually assumed) nickname. If the nickname cannot be found in the server’s internal database, it then is registered if it is valid [See section 2.2]

ChatOs messages are always lines of characters and these messages SHALL NOT exceed 512 characters in length. There is no provision for continuation message lines.

2.4. One-to-one communication

Communication on a one-to-one basis is usually only performed by clients. To send a private message you have to choose to whom to send the message. In order to provide a secure means for clients to talk to each other, it is required that the server is able to send a message in exactly one direction in order to reach any client.

Example : A message between clients 1 and 2 is only seen by server A, which sends it straight to client 2.

With this way, this provide a secure manner of exchanging messages with another client.

2.5. One-to-all

The one-to-all type of message is better described as a broadcast message, sent to all clients. On a large network of users, a single message can result in a lot of traffic being sent over the network in an effort to reach all of the desired destinations.

2.6. Private connection

The establishment of a private connection between 2 users is achieved by creating a TCP bridge relayed by the server between 2 clients to guarantee permanent anonymity. Through this feature it is possible to exchange all types of information (messages, files, etc...).

2.7. Client commands

all the above-mentioned functions are selectable by the customer by means of numerical controls. At each new step the user has the choice to enter several codes allowing the selection of the different functionalities.

Command	Description
0	New connection with the server
1	Sign out of the server
2	Request of a private connection
3	Send a public message
4	Send a private message
5	Get a list of all connected users

3. Implementation details

3.1. Introduction

To ensure the anonymity of IP adress, the server redirects the messages provide by the sender with it’s own connection to the receiver.

3.2. OP packet

OP Code	Description
0	Connection request with the server
1	Connection acceptance
2	Connection refusal
3	Connection request with a specific user
4	Public message
5	Private message

3.3. Identification

In ChatOS the first step for an user is to start a connection with the server. The user send a packet with the OPCode 0:

```
  Byte          Int               String
+-------+-------------------+--------------+
|   0   |   nickname_size   |   nickname   |
```

If the nickname is already use by an other user or it’s does not respect the rules (too long, prohibited characters), the server does not allowed the connection and send a connection refusal packet.

```
  Byte
+-------+
|   2   |
```

However, if the server accept then it send a connection acceptance packet indicating the client that he is free to come and now connected.

```
                                   Byte
                                +-------+
                                |   1   |
```

3.4. Private connection with an other client

The ChatOS specificity is to have establish a private connection between two users.

For this, the client will first ask the other client and send a private connection request packet:

```
    Byte      Int           String         Int           String
   +---+-------------+--------------+-------------+----------------+
   | 3 | S_nick_size | sender login | R_nick_size | receiver login |
```

If it’s a negative response the server close properly these two connections.

```
                                    Byte
                                 +-------+
                                 |   2   |
```

However if the response is positive then the receiver client send a connection acceptance packet to the sever who transmit this packet to the sender client.

```
                                    Byte
                                 +-------+
                                 |   1   |
```

3.5. Send a public message

When a client is connected into ChatOs, he can send a message for all connected users.

The client send a packet with the OP code 6 and his message coded in UTF8 charset:

```
 Byte      Int           String         Int         String
+---+-------------+--------------+-------------+---------+
| 4 | S_nick_size | sender login | message size | message |
```

Then the server broadcast it for each connected users.

Finally the clients catch and decode it to have the message and the sender’s nickname.

3.6. Send a private message

To send a private message, an user need to be connected on the server and be aware of the other user's nickname. If it have these two conditions, the client can send a private message on the private connection with the code 4. (For more detailed description, refer to the section 2.4 One-to-One communication)

```
 Byte      Int    String     Int       String     Int     String
+---+------------+-----+------------+----------+--------+---------+
| 5 | exp_l_size | exp | rcv_l_size | receiver | m_size | message |
```

The server at the reception of the packet checks if the recipient's login is correct, if this is the case it sends the packet to the recipient afterwards.

4. Known concerns

4.1. Authentication

Servers only have two means of authenticating incoming connections:
plain text password, and IP lookups. While these methods are weak and widely recognized as unsafe, their combination has proven to be sufficient in the past:

* public networks typically allow user connections with only few restrictions, without requiring accurate authentication.

* private networks which operate in a controlled environment often
use home-grown authentication mechanisms not available on the internet.

The current protocol offers enough to be able to easily plug-in authentication methods based on the information that a client can submit to the server upon connection.

4.2. Charset

Unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems. UTF-8 is a so called "implementation of Unicode". That’s why we choose UTF8 as the standard charset for this protocol, however We cannot guarantee complete compatibility with regard to character decoding, this depends very much on the operating system and platform used for the server and clients.