Network Working Group

PIERRE-JEAN BESNARD
LOUIS BILLAUT
Master 1 Computer Science - IGM
March 2020

THE CHATHACK PROTOCOL

Summary

CHATHACK is a very simple protocol used to create a chat between differents users. It allows
to create a private chat where two users can exchange messages and files.
This document describes the protocol and its types of packets.

Purpose

CHATHACK is a protocol to send messages and transfer files which based on TCP protocol. First of all clients connect to a server. Each connected client is identified by a login. The protocol must allow two forms of access:

- authentification by password.
- authentification without password.

Once connected and identified by a login, customers can:

- send messages which will be transmitted to all connected clients.
- send private messages and files to another client.

Compared to a standard chat server (IRC type), the special feature of this protocol is that all private messages and files are sent by a direct connection between the two clients.

The protocol allows clients to make requests for private communication and to accept / reject the request for private communication.

Acknowlegements

The protocol was originally imagined by Arnaud Carayol, and has been designed by Pierre-Jean Besnard and Louis Billaut.
Special thanks to IGM which allows us to carry out this project.

CHATHACK Packets

CHATHACK supports seven types of packets, all of which have been mentioned above:

Opcode	Operation
1	Existing Login Connection With Password

2	New Login Connection
3	Global Message
4	Private Message Asking
5	Private Message Validation
6	Private Message Denied
7	Private Message Sending (String)
8	Private Message Sending (File)
9	Private Message Connection establishment

The CHATHACK header of a packet contains the opcode associated with that packet.

CHATHACK supports eight types of server response packets, all of which have been mentioned above:

ACK	Operation
1	Packet Error
2	Password Or Login Error
3	Connected
4	Login Already Exist
5	Global Message Received
6	Private Message Asking
7	Private Message Validation
8	Private Message Denied

CHATHACK supports two types of adress during to establish a private connection, all of which have been mentioned above:

AdCode	Adress Type
1	IPv4 (4 bytes)
2	IPv6 (16 bytes)

The CHATHACK header of a server response packet contains the ACK associated with that packet.
By default the CHATHACK server considers that all text messages has been encoded in UTF-8 encoding.

Integer are signed on 4 bytes in BigEndian.
Long are signed on 8 bytes in BigEndian.
Opcode, ACK, AdCode are signed bytes.

All message, private message or file which size is upper than 1024 bytes will be automatically rejects.

Initial Connection Protocol

The connection between server and client is established by client who sends a request which specifies if the client connects with an already existing login or a new one.

The connection packet contain an opcode, a login size and a login.
Connection with existing login specifie a password size and a password.

- Existing login connection packet:

```
    1 byte       int      string        int        string
    --------------------------------------------------------
   | Opcode | login size | login | password size | password |
    --------------------------------------------------------
```

- New Login connection packet:

```
    1 byte       int      string
    -----------------------------
   | Opcode | login size | login |
    -----------------------------
```

Server answers a packet which contains an ACK which describes the client's connection state.

- Server response:

```
    1 byte
    -------
   |  ACK  |
    -------
```

Global Messages Protocol

When a client is connected to the server, he can send a message to all other connected users.

The Global Message packet contains an opcode, a message size and a message.

- Global Message Packet:

```
       1 byte        int      string       int        string
    ----------------------------------------------------------
   |   OpCode   | login size | login | message size | message |
    ----------------------------------------------------------
```

Server can send Global Message packet which correspond to other users messages.
The Global Message packet from server contains an opcode, a message size and a message.

- Global Message Packet from server:

```
     1 byte       int      string      int        string
    -------------------------------------------------------
   |   ACK   | login size | login | message size | message |
    -------------------------------------------------------
```

Private Connection Etablishement Protocol

When a client is connected to the server, he can ask to another user to create a private connection which allows them to exchange private messages and files.

The private connection establishement packet contains an opcode, a receiver login size, and a receiver login.

- Private Connection Establishement packet:

```
     1 byte           int             string            int
    -----------------------------------------------------------------
   | Opcode | sender login size | sender login |receiver login size|
    -----------------------------------------------------------------

        string
   ---------------
   receiver login |
   ---------------
```

Server can send Private Connection asking from another user.
This Private Connection asking contains an opcode, a sender login size and a sender login.

- Private Connection asking from server:

```
     1 byte          int             string
    --------------------------------------------
   |   ACK   | sender login size | sender login |
    --------------------------------------------
```

When an user receives a Private Connection asking from the server he can accept or decline the connection by sending packet.
If the user accept the connection, he has to specifie his adress, a port number where the other user can connect and an id which correspond to a securised communication key which will be only know by the other user.

Acceptation Connection packet contain an opcode, a login size, a port number, an adress size and an id. (Please notice that the adress size is specified by the adCode which is the adress type. All adCode are specified at the beghining of this RFC).

- Acceptation Connection packet:

```
     1 byte           int            string            int
    -----------------------------------------------------------------
   | Opcode | sender login size | sender login |receiver login size|
    -----------------------------------------------------------------

         string          int       1 byte   bytes   long
   -----------------------------------------------------
    receiver login | port number | adCode | adress | id |
   -----------------------------------------------------
```

When a connection is accepted by a user and he has sent his packet, server send a Validation Private Connection packet to the asker user for telling him that the receiver user has accepted the connection.

The Validation Private Connection packet contain an opcode, the port number where the user can communicate with the user, an adress size, an adress and an id which correspond to a securised key which will be the id of the private communication.

- Validation Private Connection packet :

```
   1 byte       int      string      int      1 byte    bytes
  --------------------------------------------------------------
 |   ACK  | login size | login | port number | adCode | adress |
  --------------------------------------------------------------

 long
 ----
  id |
 ----
```

If the user decline the connection, he has to send a Refusal Private Connection packet.

Refusal Connection packet contain an opcode, a login size and the login of the user which is refused.

- Refusal Private Connection packet:

```
   1 byte         int                string            int
  ----------------------------------------------------------------
 | Opcode | sender login size | sender login |receiver login size|
  ----------------------------------------------------------------

      string
 -----------------
   receiver login |
 -----------------
```

Server send an ACK to the user which is refused.

- Server Refusal Response:

```
  1 byte      int    string
  --------------------------
 | ACK | login size | login |
  --------------------------
```

Private Messages Protocol

Users who have established a private connection can send private messages or files to each other. They have to start the connection by sending a frame like :

```
    1 byte        int      string  long
 --------------------------------------
|   Opcode   | login size | login | id |
 --------------------------------------
```

The Private Message packet contains an opcode, an id which correspond to a private communication key a message/file size and a message/file.

- Private Message Packet :

```
      1 byte          int     string      int        string
    ----------------------------------------------------------
   |   Opcode   | login size | login | message size | message |
    ----------------------------------------------------------
```

- Private File Packet:

```
     1 byte          int     string      int            string
   ---------------------------------------------------------------
  |   Opcode   | login size | login | file name size | file name |
   ---------------------------------------------------------------

     int     bytes
   ------------------
    file size | file |
   ------------------
```