Jie SONG, Chaochen MA
IMT Atlantique
September 26, 2017

Protocol Specification: ChatWhileWatching (Text Version)
ChatWhileWatching-text

Abstract

This protocol specification for the text version of the protocol. The goal of this protocol is to allow a client to log in 'ChatWhileWatching' to find the film list and the user list saved in the server. It also allows different users to chat in the main interface with movie lists and chat in the interface of MovieRoom.

Table of Contents

1. Introduction

'ChatWhileWatching' is an application which allows differents clients to log in, watch the lists of users and movies. Every client can chat in the main interface and choose a movie then chat in the

MovieRoom. All messages sent to the same interface can be seen by others.

Clients can use either UDP or TCP to exchange messages with the server. While sending a reply, the server MUST use the same layer for protocol used by the client for the corresponding request.

1.1. Requirements Language

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].

2. Message Format

For specifying differents kinds of messages, we set various formats of message for reserving useful information. For each kind of format, we put an identifer in the first place to identify them.

2.1. Scenario 1: while loging in

```
                        LOGIN<CRLF>
                        IP<CRLF>
                        PORT<CRLF>
                        USERNAME<CRLF>
```

LOGIN

For requests(i.e., messages sent by a client), this is the identifer to tell that this message is a message of Login. Defautly, we set the value of LOGIN as 01 to identify a login request. For responses(i.e., messages sent from the serveur to the client), if this login request is accepted, LOGIN in the message sent from the server will be 11 ;If not, LOGIN will be 10.

2.2. Scenario 2: Chat Message request

```
                        CHAT<CRLF>
                        IP<CRLF>
                        PORT<CRLF>
                        USERNAME<CRLF>
                        TIME:<space>TEXT<CRLF>
```

CHAT

CHAT is the identifer to tell that this message is a message of chatting. Defautly, we set the value of CHAT as 02 to identify a chatting request.

TIME

For requests(i.e., messages sent by a client), this is the time at which the request was sent, expressed as the number of seconds since Januaey 1st, 1970.For responses(i.e., messages sent from the serveur to the client), TIME field MUST be the same as the TIME field of the corresponding request.

In every message the TIME must be followed immediately by a single colon, followed a single white space.

TEXT

The text of the question, for requests, and the text of the reply for responses.

2.3. Scenario 3: Chat Message reply in main interface and movie room

```
                        TRANSFER<CRLF>
                        IPAddressPresentList<CRLF>
                        TIME:<space>TEXT<CRLF>
```

TRANSFER

TRANSFER is the identifer to tell that this is a transfer message when a server received a chat message and is going to send this message to the others users who are online. Defautly we set the value of TRANSFER as 31 to be a reply in the main interface and 32 to be a reply in the present movie room.

IPAddressPresentList

When the server contribute a chat message in the main interface, IPAddressPresentList is a list of IP address which points to the the users who are online when the message is going to be sent.When the server contribute a chat message in a certain movie room, IPAddressPresentList is a list of IP address which points to the the users who are online in this certain movie room when the message is going to be sent.

2.4. Scenario 4: Enter MovieRoom Request/Reply

```
                        MovieRoom<CRLF>
                        IPAddress<CRLF>
                        PortNumber<CRLF>
                        USERNAME<CRLF>
                        MovieName<CRLF>
```

MovieRoom

MovieRoom is the identifer to tell that this message is a message while a client would like to enter the MovieRoom. Defautly, we set the value of MovieRoom as 04 to identify a chatting request. For responses(i.e., messages sent from the serveur to the client), defautly, we set the value of MovieRoom as 41 to identify an accepted chatting reply and 40 for refusing this chatting reply.

2.5. Scenario 5: List of Clients

```
                        ListClient<CRLF>
                        IPAddress<CRLF>
                        PortNumber<CRLF>
                        Username<CRLF>
                        Userstate<CRLF>
                        RoomNum<CRLF>
```

Userstate

Userstate indique whether a client is online or not. If a client is online, Userstate will be 1, if it is offline, userstate will be 0. The list of client message will be sent to all the clients online when a client logs in or out.

ListClient

Defautly we set the value of ListClient as 05 to identify a message of reserving users’ states.

RoomNum

RoomNum stores the number of movie room where the user is on. If the user is just in the main interface, RoomNum=NUll.

2.6. Scenario 6: List of Movies

After log in, a server will receive the list of the names of movies.

```
ListMovie<CRLF>
IPAddress<CRLF>
PORT<CRLF>
Username<CRLF>
MovieNameList<CRLF>
```

ListMovie

Defautly, we set the value of ListMovie as 61 to identifer it is a message for sending the list of Movies’ name.

MovieNameList

Server MUST reserve the list of movies’ name in its local database.

2.7. Scenario 7: LogOut Message

```
LOGOUT <CRLF>
IP<CRLF>
PORT<CRLF>
USERNAME<CRLF>
```

LOGOUT

For requests(i.e., messages sent by a client), this is the identifer to tell that this message is a message of LogOut. Defautly, we set the value of LOGOUT as 07 to identify a login request. For responses(i;e., messages sent from the serveur to the client), If this login request is accepted, LOGOUT in the message sent from the server will be 71;If it is refused, LogOut will be 70.

The serveur MUST always reply to a request. Clients MUST send at most one request every second.

3. Reliability

When using UDP, a client MAY resend a request for which it has not yet received a response after a certain time. In order not to overload servers, clients SHOULD wait at least 2 seconds before resending a request.

4. Server Configuration and Multiple Servers

Each server MUST maintain a local data base mapping names to IP addresses. If a server does not find a certain name in its local data base, instead of immediately sending a negative response, the server MAY contact one or more other servers, asking if they know the IP address corresponding to the name requested by the client. Upon receiving a response the server should forward it to the corresponding client. Note that in this case it is up to the server to resend the request to another server, if it has not received a response after a certain time.

5. Examples

5.1. Example 1: while a client logs in

Suppose a client would like to log in the ’chatwhilewatching’ application. It should send the following message to the serveur(recall that CR represents the ASCII character "Carriage Return" (13) and LF represents the ASCII character "Line Feed" (10))::

```
01CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
```

If the username is correspondent to the ip address, the server MUST respond with the following message:

```
11CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
```

If, instead, the requested name does not exist, the server MUST respond with the following message:

```
10CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
```

5.2. Example 2: Exchange chatting messages

At 3.35 pm on March 3rd, 2010, a client sends the following message in the dialog box or movieRoom:"Hey, anybody online?"

```
02CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
1267626900: Hey, anybody online?CRLF
```

The server will transfer this message to its list of ip adresses with this format:

```
31CRLF
192.168.1.1;192.168.2.1;192.168.2.3;CRLF
1267626900: Hey, anybody online?CRLF
```

5.3. Example 3: while a client would like to enter a MovieRoom

Suppose a client would like to enter a certain movie room which plays a movie "FINALWORLD" . It should send the following message to the serveur(recall that CR represents the ASCII character "Carriage Return" (13) and LF represents the ASCII character "Line Feed" (10))::

```
04CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
FINALWORLDCRLF
```

If it is accepted, the server MUST respond with the following message:

```
41CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
FINALWORLDCRLF
```

If, instead, the server MUST respond with the following message:

```
40CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
FINALWORLDCRLF
```

5.4. Example 4: While a client logs in and a server reserves the list of client

Once a client has logged in the main interface, the server will receive it and add to the list of users. For exemple, user David has logged in:

```
05CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
1CRLF
NULLCRLF
```

Once a client has logged out this application, the server will refresh ist of users. For exemple, user alerto has loged out this application:

```
05CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
0CRLF
NULLCRLF
```

Once a client has logged in a certain movie room, the server will receive it and add to the list of users in this room. For exemple, user David has entered in the movie room whose number is 32:

```
05CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
1CRLF
32CRLF
```

Once a client has logged out a certain movie room, the server will refresh ist of users in this room. For exemple, user alerto has loged out of the movie room whose number is 32:

```
05CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
0CRLF
32CRLF
```

5.5. Example 5: While a server sends the list of movies

Once a client has logged in, it will receive a list of movies. For exemple, Titanic is in the list:

```
06CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
FINALWORLD;SPIDERMANCRLF
```

5.6. Example 6: While a client wants to log out

We suppose that a client will log out,it will send the log out message to the server, and the server will transfer its state. The message by a client SHOULD like :

```
07CRLF
192.168.1.1CRLF
2333CRLF
albertoCRLF
```

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Author’s Address

Jie SONG, Chaochen MA
IMT Atlantique
Brest, Bretagne 29280
France

Phone: +33 06 50 07 86 34
Email: jie.song@imt-atlantique.net; chaochen.ma@imt-atlantique.net