C. HOCINE
D. MIKOU
S. MADRUGA
Q. ZHAO
Telecom Bretagne
March 2, 2014

Group 18: c2w protocol specification proposal

Abstract

This protocol supports the application Chat While Watching (c2w) and allows the clients to communicate both in public and in private. Server forwards the chat messages among the clients.

Table of Contents

1. Introduction

The goal of this protocol is to support the "c2w" application which allows the clients to chat while watching a movie. The protocol is defined to work on the application layer of the ISO/OSI model and is able to use either UDP or TCP connection.

This specification defines all the communication requirements between the client and the server. Starting with the message’s format, it will then specify all the possible cases for the application. As illusatration, some scenarios will be presented with all communication details. For the sake of simplicity, all the messages will have the same format.

The main function of the "c2w" application is to make it possible for users to chat while they are watching a movie. After the right User name, the Port Number and the IPAddress or the name of the server, the client logs in the application. A new windowis hence opened with the "main room". It contains a list of all the available movies, a list of all users in the system with their status, a chat area and a text input box. When the client decides to join a movie, a new window is opened: the movie room. It shows the movie video, the list of the users watching the same movie and a text−input area for public chat messages. Either in the main room or in the movie room, the client can have a public or a private chat communication. Finally, the protocol ensures a deconnection function.

All these specifications will be seen further with more explications and more details.

2. Packet format

All message have the same format, shown in figure Figure 1.

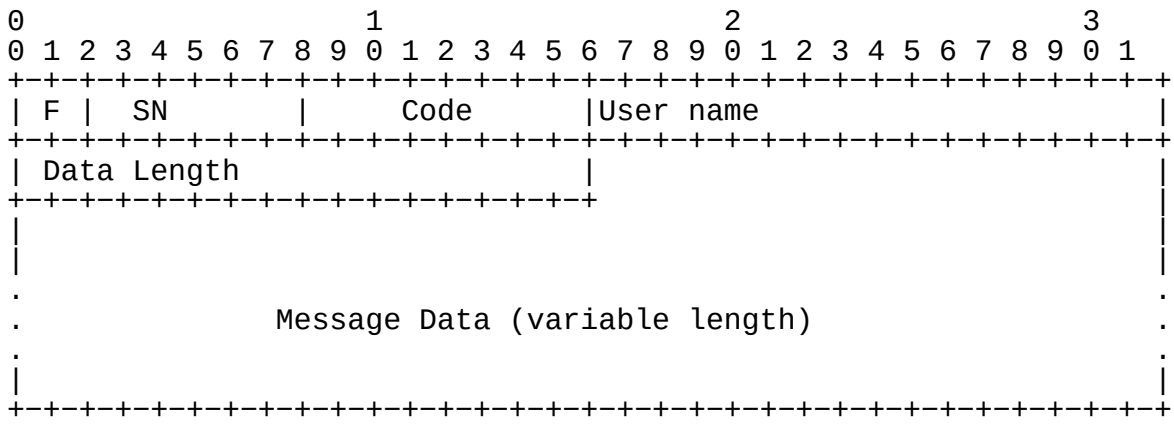


Figure 1

Figure 1

2.1. Flag field (2 bit)

This field is used to identify the source of the message. If the sender is a client, we set this field to 11, if the sender is a server, we set this field to 00. Because the first bit can be easily modified, IF this field is composed by two different numbers 01 or 10, the reception MUST send a ’false’ ACK.

2.2. SequenceNumber (6 bits)

This field specifies the sequence number of the message that the sender sends. The recipient MUST receive the received messages in order. If the received message number is smaller than the waiting message number, the recipient SHOULD drop the message.

2.3. Code (8 bits)

This field is divided into Client part and Server part. The choose denpends on the field Flag.

The table lists from client to server is shown in the figure Figure 2.

Code	Message Object	Message content
00000001	Connection Request	IP address,Port Number User Name
00000010	Main Room Request	
00000100	PriChat Request	User Name,Chat Message
00001000	MultiChat Request	Chat Message
00010000	End PriChat Request	
00100000	Deconnection Request	
01000000	Message ACK	
10000000	WatchMovie Request	Movie name

Figure 2

Connection Request: the client wants to log in the application, and the client MUST send the name (or IP address), port number of the server as well as his/her user−name.

Main room Request: after the commande Connection Request, the client SHOULD send the Main Room Request to server to apply for the main room window.

PriChat Request: if a client wanted to communicate with another client in the same room, he would send the destination user name and the chat message.

MultiChat Request: the client wants to talk to all the members in the same room, he/she sends the chat message to the server.

End Prichat Request: when the client wants to leave the private chat, he/she sends the request to server.

Deconnection Request: the client sends the request to leave the application.

Message ACK: to tell the server that the client has received the sequence. Message ACK could be ’false’ and ’successful’.

The table lists from server to client is shown in the figure Figure 3.

Code	Message Object	Message content
00000001	Message ACK	
00000010	Main room Response	Movie List, User List User State List
00000100	Movie Room Response	A video, User List
00001000	PriChar Forwarding	User Name,Chat Message
00010000	MultiChat Forwarding	User Name,Chat Message
00100000	Update Mainroom List	User List,User State List
01000000	Update Movieroom List	User List,User State List

Figure 3

Message ACK: to tell the client that the server has received the sequence. Message ACK could be 'false' and 'successful'.

Main room Response: the server sends the main room window to the client, including the list of available movies, the list of all the users and the list of their status.

Movie Room Response: the server answers the client the corresponding video, the list of users in the same room.

PriChar Forwarding: after the server receives the request of a private chat, if the client is in available, the server forwards the chat message and its source sender to the destination client.

MultiChat Forwarding: after the server receives the request of a public chat, the server SHOULD forward the chat message and the source sender to all the available users with the same satuts as the sender.

Update Mainroom List: when a client adds or leaves the main room, the server MUST send the new data: the new User List and the new User Statut List to all the user in the main room.

Update Movieroom List: when a client adds or leaves the movie room, the server MUST send the new data: the new User List in watching the same movie.

2.4. User name(16 bit)

This field specifies the source or destination User name of the client, the situation depends on the Flag field. If the Flag equals 00, this field contains the User name as the destination, if the Flag equals 11, this field contains the User name as the source.

2.5. Data Length (16 bit)

This field specifies the total length in bytes of the Message Data field. The sender MUST ensure that this field contains the correct value.

2.6. Message Data (variable length)

This field contains message encoded in ASCII. This field COULD contain no data.

3. Reliability

When using UDP, a client and the server MAY send a request if they wait at least 2 seconds with no answers. It is not the case for TCP as it is reliable.

4. Server configuration

Each server MUST maintain a local data base mapping clients names to their IPadresses and to their User name. In his/her first connection, the client MAY give his/her IP adress or his/her User name.

The server MUST update the user list in the main room and in each movie room when a client joins in or leaves.

5. Example scenario

5.1. Scenario 1: Connection to the application

A client wants to log in the application. After the client sends the correct informations to the server, the server replies the main room window to the client, including the list of all users in the system with their statuts, all the available movies.

The client Bob wants to connect to the application.

MESSAGE 1: Bob sends the Connection Request to the server, if the server does not reply in 2 seconds, Bob resends the same message.

Flag: 11

Code: 00000001

Message Data: Server Name: local host CRLF Prot Number: 8888 CRLF User Name: Bob CRLF

MESSAGE 2: the server sends the Message ACK to the client

Flag: 00

Code: 00000001

Message Data: False

MESSAGE 3: Bob sends the Connection Request to the server, if the server does not reply in 2 seconds, Bob resends the same message.

Flag: 11

Code: 00000001

Message Data: Server Name: local host CRLF Prot Number: 8888 CRLF User Name: Bob CRLF

MESSAGE 4: the server sends the Message ACK to the client

Flag: 00

Code: 00000001

Message Data: Successful

MESSAGE 5: after the client connect successfully to the server, the client sends the request to get to the main room.

Flag: 11

Code: 00000010

Message Data:

MESSAGE 6: the server receives the client’s request and sends the main room contents to the client.

Flag: 00

Code: 00000010

Message Data: Movie List CRLF User List CRLF User Statut List CRLF

MESSAGE 7: the client receives the main room window and he/she sends

a acknowledgement to the server to remind the server to update the User List data.

Flag: 11

Code: 01000000

Message Data: Successful

MESSAGE 8: the new client with a statue 'A' in the User List, the server forwards the new User List and new User Statut List to all the users in the main room.

Flag: 00

Code: 00100000

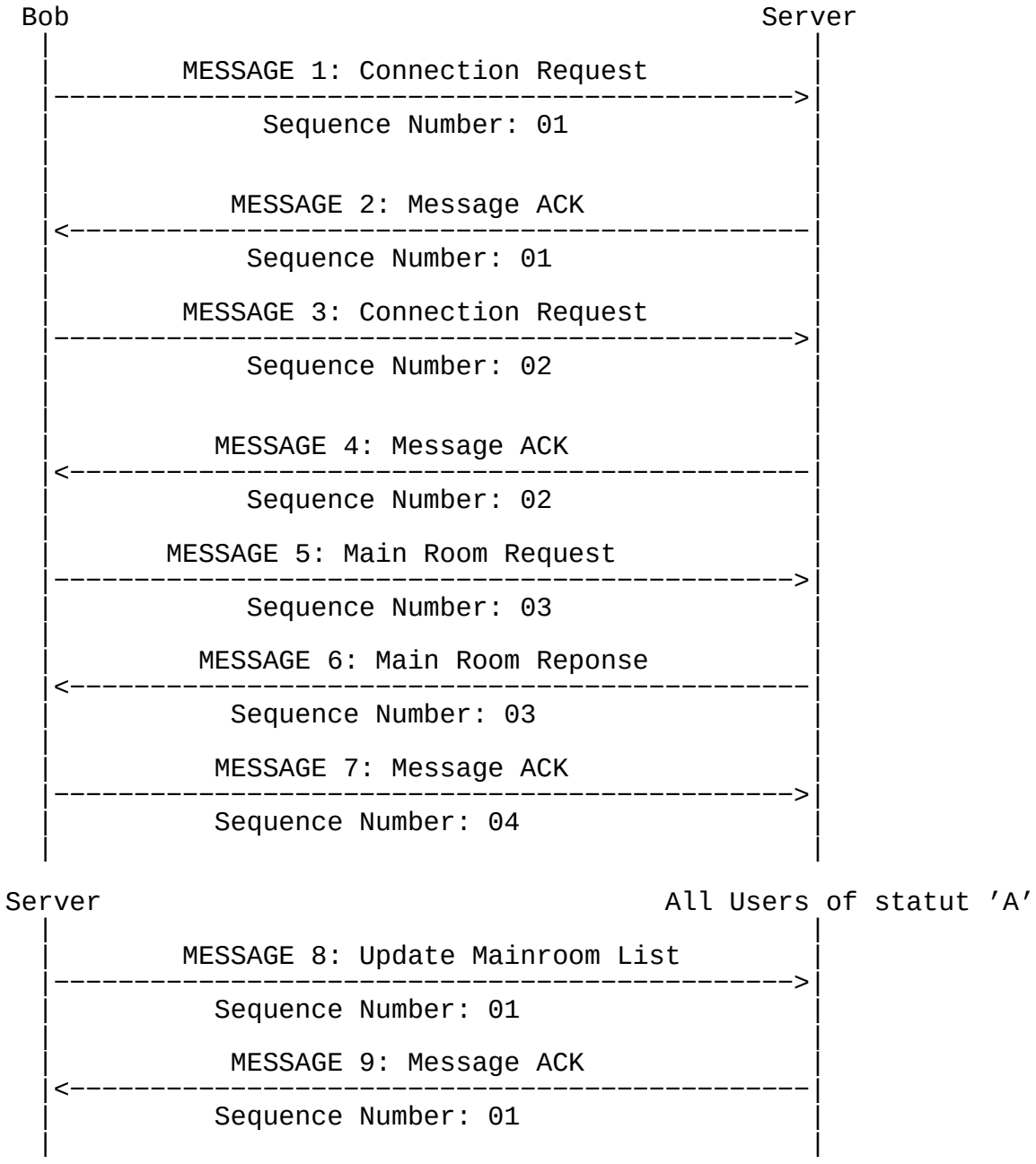
Message Data: User List CRLF User Statut List CRLF

MESSAGE 9: the users in the main room receives the new user date and acknomledge the update.

Flag: 11

Code: 01000000

Message Data: Successful


Bob
Server
MESSAGE 1: Connection Request
Sequence Number: 01
MESSAGE 2: Message ACK
Sequence Number: 01
MESSAGE 3: Connection Request
Sequence Number: 02
MESSAGE 4: Message ACK
Sequence Number: 02
MESSAGE 5: Main Room Request
Sequence Number: 03
MESSAGE 6: Main Room Reponse
Sequence Number: 03
MESSAGE 7: Message ACK
Sequence Number: 04
Server
All Users of statut 'A'
MESSAGE 8: Update Mainroom List
Sequence Number: 01
MESSAGE 9: Message ACK
Sequence Number: 01

5.2. Scenario 2: Access to a movie room

A client wants to access to a movie room. He/She chooses a film and sends a request. Once it gets the message, the application MUST open the movie room window containing the: the list of users in that specific room, the corresponding video, a chat area and a text−input box.

The client Bob wants join the movie "movie1".

MESSAGE 1: Bob sends the WatchMovie Request to server. This request MUST contain the movie, which is one of the movies in the movie list of the main room. If the client doesn’t receive an ACK from the server after 2 seconds, it will resend the request.

Flag: 11

Code: 10000000

Message Data: Movie Name CRLF

MESSAGE 2: The server sends to the client: the video flow of the choosen film and the list of the users who are in the same movie room.

Flag: 00

Code: 00000100

Message Data: Movie CRLF List of the users CRLF

MESSAGE 3: the client sends the Message ACK to the server saying that it received the data.

Flag: 11

Code: 01000000

Message Data: Successful

MESSAGE 4: the server forwards the new User List to the users in the movie room

Flag: 00

Code: 00100000

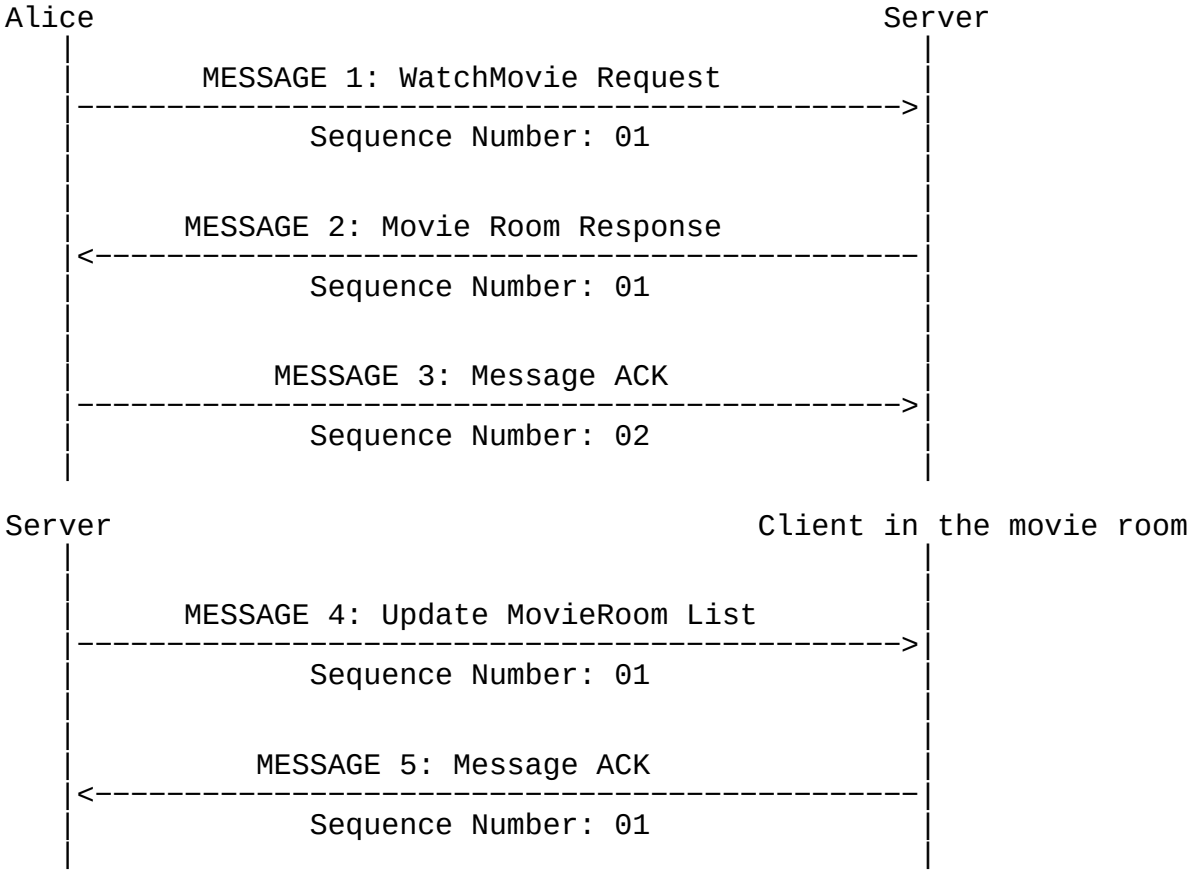
Message Data: User List CRLF

MESSAGE 5: the client in the movie room updates the user data and sends the acknowledgement to the server.

Flag: 11

Code: 01000000

Message Data: Successful



5.3. Scenario 3: send a public message

A client in the movie room or in the main room wants to send a public message to the other clients in the same room.

The client Bob wants to communicate with the users in his room.

MESSAGE 1: Bob sends a MultiChat Request to the server with his chat message.

Flag: 11

Code: 00001000

Message Data: Chat Message: Do you like the film? CRLF

MESSAGE 2: the server sends the Message ACK to the client

Flag: 00

Code: 00000001

Message Data: Successful

MESSAGE 3: the server accepts the message of Bob and forwards the message to all the clients in the same room with Bob, the the room could be the specific movie room or the main room.

Flag: 00

Code: 00010000

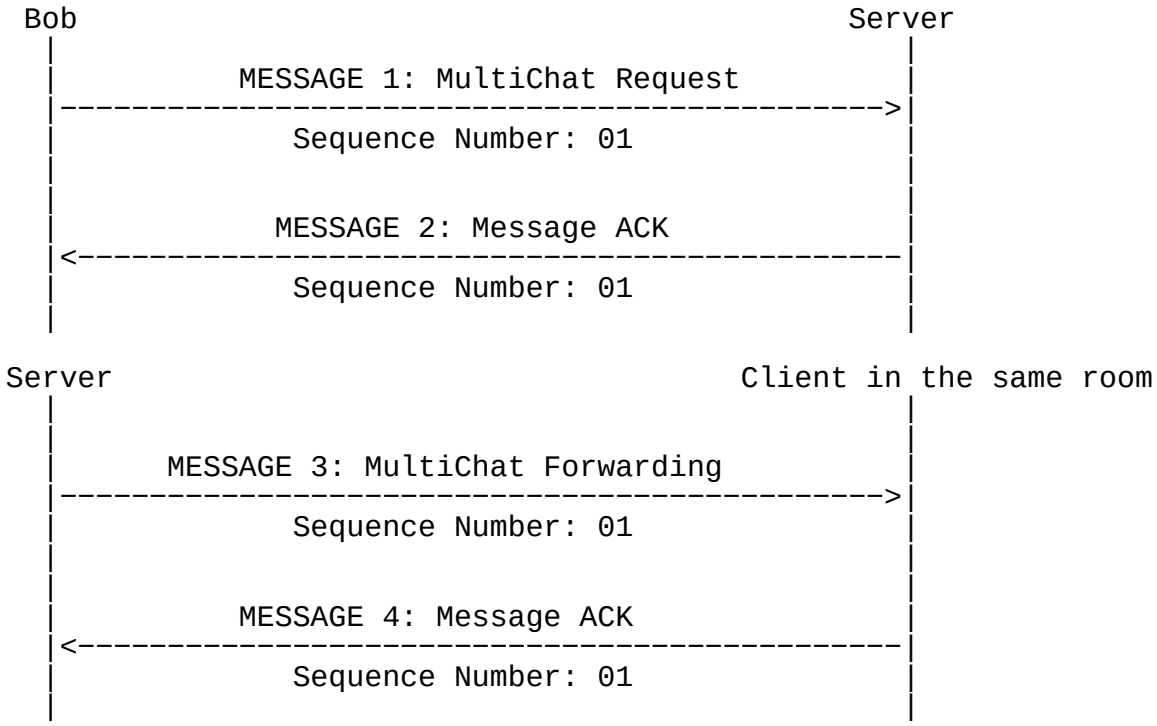
Message Data: User Name: Bob CRLF Chat Message: Do you like this film? CRLF

MESSAGE 4: the clients in the same room send the ackonmledge

Flag: 11

Code: 01000000

Message Data: Successful



5.4. Scenario 4: Open a private chat room

A client wants to talk to one of the connected user privately. The application MUST open a private room window.

The client Bob wants to talk to the client Alice.

MESSAGE 1: Bob sends the PriChat Request to the server. This request MUST contain the destination’s User name and the chat message.

Flag: 11

Code: 00000100

Message Data: destination User name CRLF Chat message CRLF

MESSAGE 2: the server sends the Message ACK to the client to say that it received the client request and that it is sending the chat message to the destination.

Flag: 00

Code: 00000001

Message Data: successful

MESSAGE 3: The server forwards the chat message to Alice.

Flag: 00

Code: 00001000

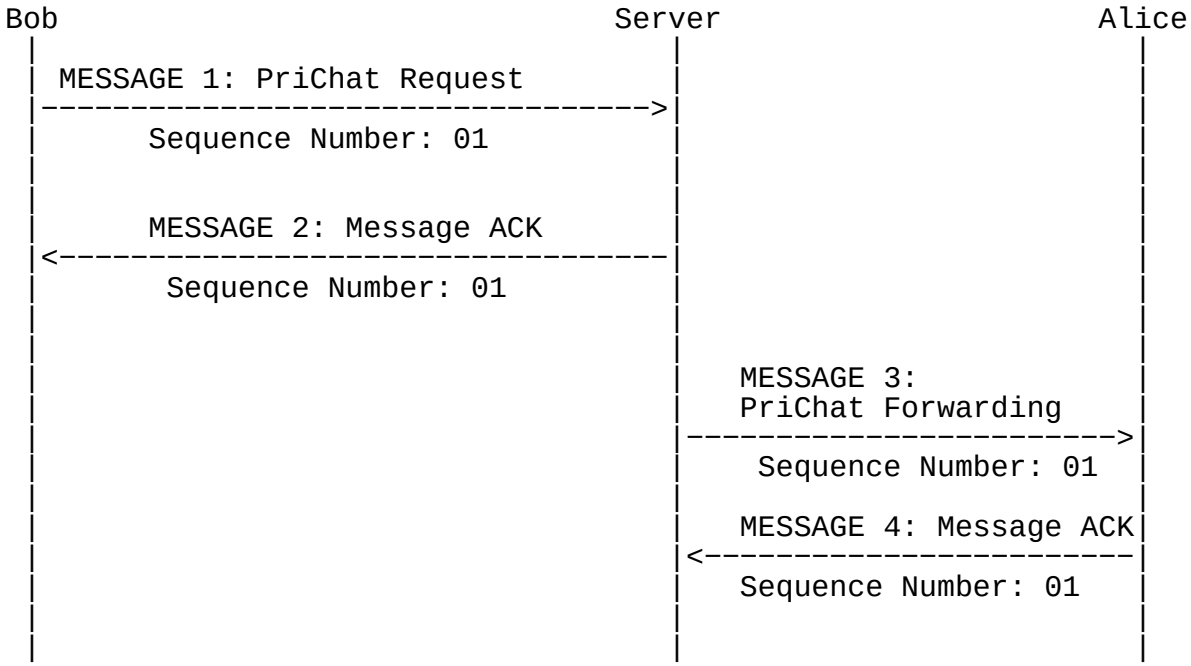
Message Data: User name: Alice CRLF Chat message: "Hello" CRLF.

MESSAGE 4: Alice sends a positive ACK to the server. This ACK means that Alice received the data sent by the server.

Flag: 11

Code: 01000000

Message Data: Successful



5.5. scenarion 5: leave the movie room

A client wants to return to the main room and the server MUST send the new User List to the clients both in the movie room and the main room.

The client Bob wants to quit the movie room and return to the main room.

MESSAGE 1: Bob sends the Main Room Request to return to the main room.

Flag: 11

Code: 00000010

Message Data:

MESSAGE 2: the server receives the client’s request and sends the main room contents to the client.

Flag: 00

Code: 00000010

Message Data: Movie List CRLF User List CRLF User Statut List CRLF

MESSAGE 3: the client receives the main room window and he/she sends a acknowledgement to the server to remind the server to update the User List data.

Flag: 11

Code: 01000000

Message Data: Successful

MESSAGE 4: the server forwards the update User List to the users in the movie room

Flag: 00

Code: 00100000

Message Data: User List CRLF

MESSAGE 5: the client in the movie room updates the user data and sends the acknowledgement to the server.

Flag: 11

Code: 01000000

Message Data: Successful

MESSAGE 6: the server forwards the update User List to the users in the main room

Flag: 00

Code: 00100000

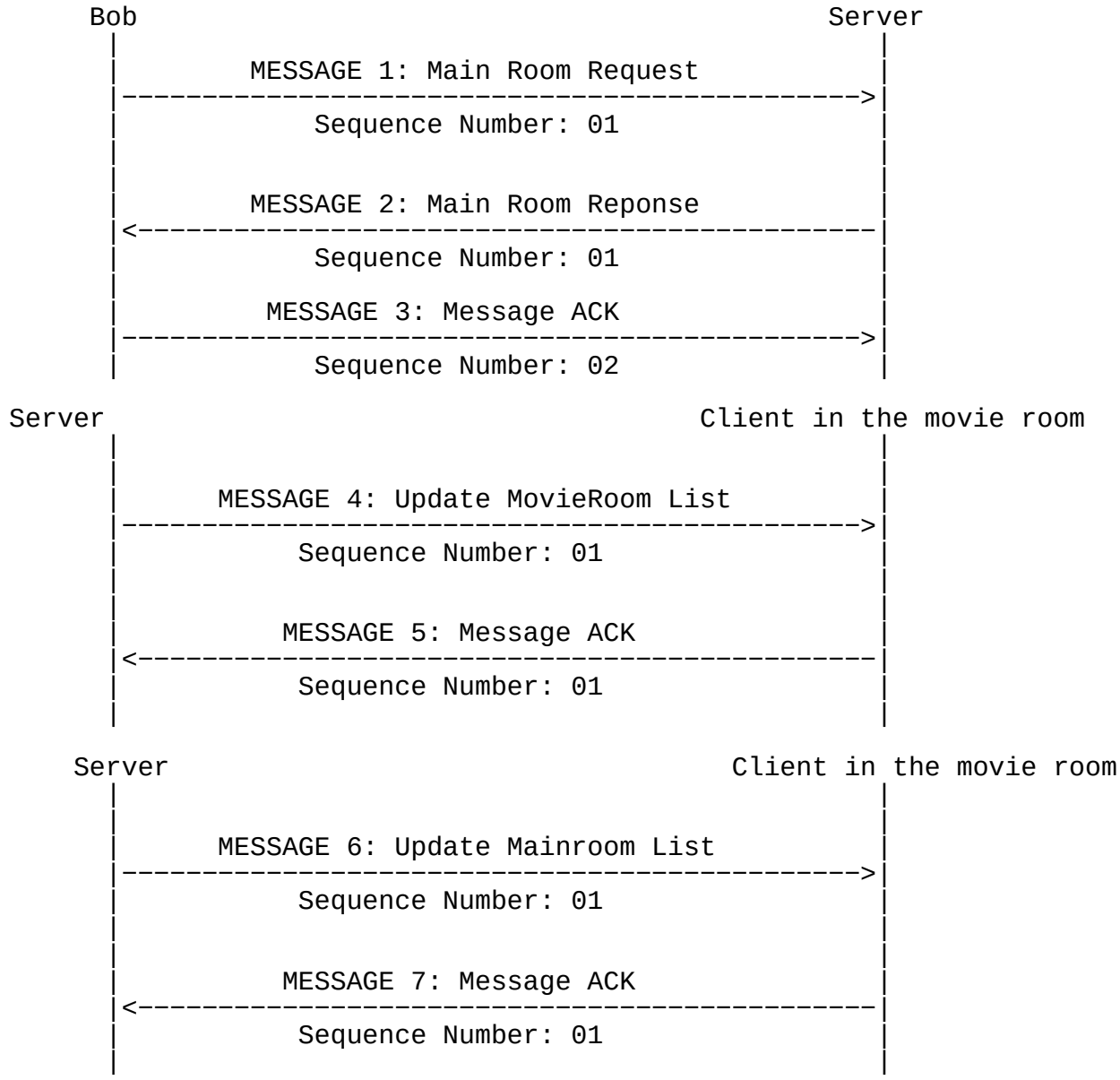
Message Data: User List CRLF

MESSAGE 7: the client in the main room updates the user data and sends the acknowledgement to the server.

Flag: 11

Code: 01000000

Message Data: Successful


Bob
Server
MESSAGE 1: Main Room Request
Sequence Number: 01
MESSAGE 2: Main Room Reponse
Sequence Number: 01
MESSAGE 3: Message ACK
Sequence Number: 02
Server
Client in the movie room
MESSAGE 4: Update MovieRoom List
Sequence Number: 01
MESSAGE 5: Message ACK
Sequence Number: 01
Server
Client in the movie room
MESSAGE 6: Update Mainroom List
Sequence Number: 01
MESSAGE 7: Message ACK
Sequence Number: 01

5.6. scenario 6: quite the main room

A client wants to log out the application. After the client sends the correct informations to the server and he/she leaves the application.

The client Bob wants to quit to the application.

MESSAGE 1: Bob sends the Deconnection Request to the server

Flag: 11

Code: 00100000

Message Data:

MESSAGE 2: the server sends the Message ACK to the client

Flag: 00

Code: 00000001

Message Data: Successful

MESSAGE 3: the client Bob leaves the main room window and the server sends the new User List data to the clients in the main room.

Flag: 00

Code: 00100000

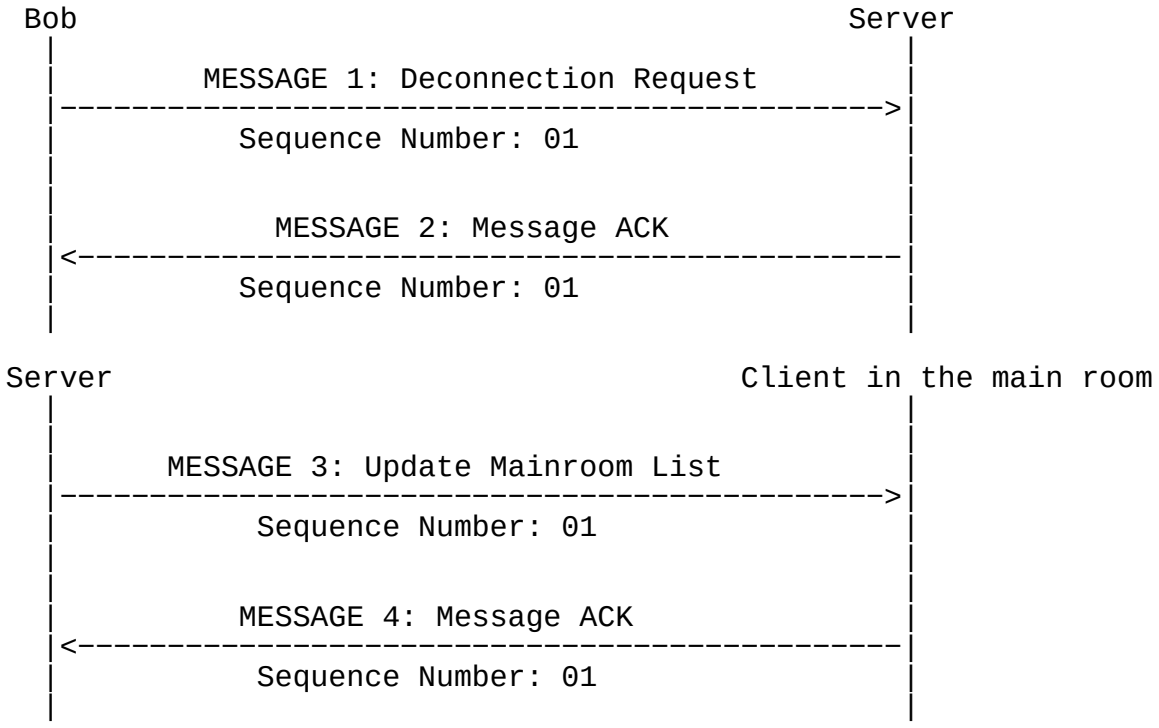
Message Data: User List CRLF User Statut List CRLF

MESSAGE 4: the client in the main room receives the new client list and sends an acknomledgement to server.

Flag: 11

Code: 01000000

Message Data:



6. Conclusion

This document specifies all the communication requirement for a a reliable exchange of messages between the client and the server. The protocol can hence use either a TCP or a UDP communication. For the sake of simplicity, as said before, we have decided to attribute the same format for all the messages. The difference will be in the values of the header fields.

7. References

7.1. Normative References

[1] authSurName, authInitials., "Minimal Reference", 2006.

7.2. Informative References

[2] Mad Dominators, Inc., "Ultimate Plan for Taking Over the World", 1984, <http://www.example.com/dominator.html>.

[3] Davies, E., "Internet Draft Sample", 2006, <http://tools.ietf.org/tools/templates/draft-davies-template-bare.txt>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Celia HOCINE
Telecom Bretagne
Brest,
France

Email: celia.hocine@telecom-bretagne.eu

Dounia MIKOU
Telecom Bretagne
Brest,
France

Email: dounia.mikou@telecom-bretagne.eu

Sarah PONTES MADRUGA
Telecom Bretagne
Brest,
France

Email: sarah.pontesmadruga@telecom-bretagne.eu

Quancheng ZHAO
Telecom Bretagne
Brest,
France

Email: quancheng.zhao@telecom-bretagne.eu