



LUNDS UNIVERSITET

Lunds Tekniska Högskola

SimpleAudioChat Network Programming (EDA095)

Niclas Lövdahl, (dic13nlo@student.lu.se), C13
Jacob Hegelund, (dic13jhe@student.lu.se), C13
Alexander Ljungqvist, (har12al2@student.lu.se), C13
David Jaenson, (dat11dja@student.lu.se), D12

22 maj 2015

1 Background

The goal with this project was to create a type of phone application with the ability to provide similar functionality as Skype. With guidance from the Java Sound API, the was supposed to be a complete sound and video-transferring program. Thus the project was named SimpleVideoChat. With the help from DatagramSockets[?] and the Sound API[?] methods: Send, receive, write and read everything was provided to create the program the group had in mind.

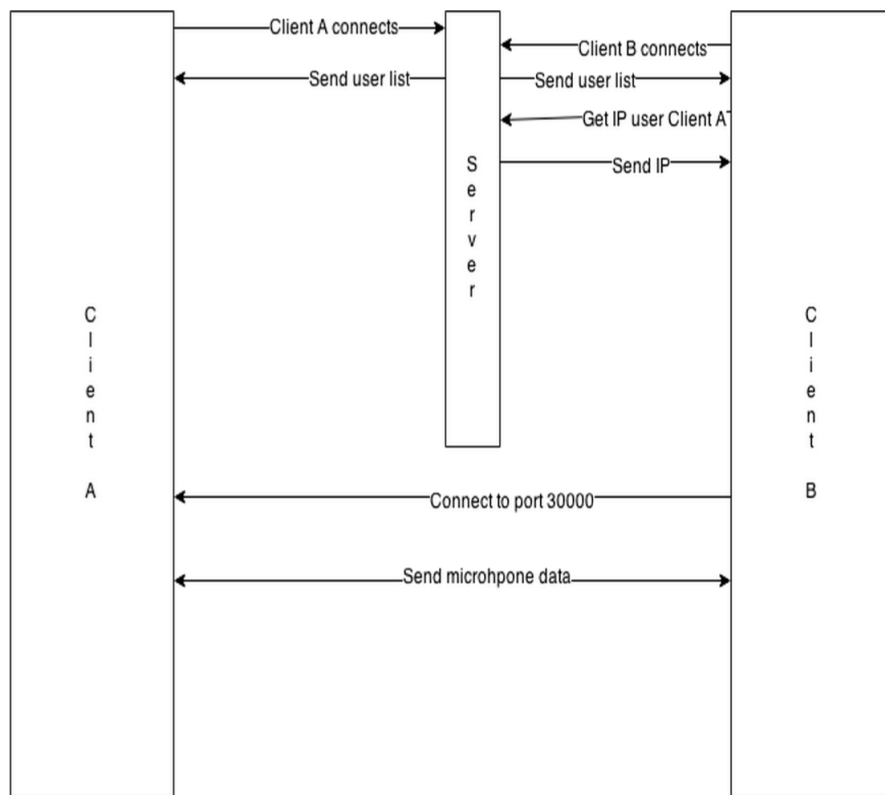
2 Requirements

The main requirement with the system was that it should be a fully functional video chat system. In the perspective of design, the program should be stripped down and minimalized to make it easier to interact with. When a user connects to the program, a list of other online users will be shown. Then a user should be able to choose a another user and call him.

- The system should be able to manage more than one user in call, creating a group conversation.
- The system should provide video transmission.
- The system should be able to provide sound transmission.
- The system should be able to provide with both video and sound transmission simultaneously.
- The system should be able to provide contact information in the form of name.
- The system should be able to provide simplest version of buttons for the user.

3 Model

Client A connects to the stationary server with a chosen username. The server saves the username of the client to the IP-address. Client B then does the same thing with another name. Then, Client B can see that client A is online with the specified username. Client B asks the stationary server for the IP-address for the name of Client A, and then has the ability to make a call to Client A.



MainServer - The main server class is the so called Spider for our stationary server. In this spider, everything about the users and updates are saved online.

ConnectionListener - The ConnectionListener lies in the background waiting for new users to connect. It starts the class called ServerThread which runs in the background waiting for commands from the user.

ServerThread - See above.

Server - The server class is located on the client side and listens to incoming connections from other clients.

Client - The class Client is the class on the client side that opens a Socket and sends it to the threads.

SendMicThread - SendMicThread takes care about the OutPutStream for a client. It encodes the data that is recorded by the client and sends the data after a connection.

GetMicThread - The GetMicThread class handles the InputStream for a client. IT both receives and decodes the data.

4 Manual

Startup: The program asks you for your Username which you will have to type in.

At launch: User receives a list of other connected users and has the possibility to call.

At call: User chooses which other users he wants to call. Then user pushes the call button and the call may start.

- Call: When the call button is pressed, caller calls the chosen whom has been chosen in the list.
- Hang up: When the Hang Up button is pressed the conversation is terminated. If user is not in a conversation, the Hang Up button does nothing.
- Quit: When the Quit button is pressed the application is terminated and the user is logged off". If a user is in the middle of a conversation, it would look like the user hung up before logging out.

5 Evaluation

Our first idea was to create the video chat system. It was a quite including project that we luckily in time understood was impossible to perform within our range of time. Instead, we chose to focus solely on the sound transmission and the name of the project was changed to SimpleAudioChat. This requirement we succeeded in fulfilling. We were not able to fulfill many of our initial requirements. As earlier said, video was out of time range. Group call was as well. Every other requirement was fulfilled.

In order to fix every requirement, the time frame would have needed to be longer. It would only require more implementation and it would not cause much altering in the code as it is right now.

5.1 The project

The group thought the project was a good idea and the fact that they were given pretty much full liberty paved way for more initiatives and more effective work. However, there comes pressure from the exam, which nullifies a lot of the liberty for the project. This can create various problems where the project is only seen as an obstacle before attending the exam.

6 Program

<https://github.com/NiclasLovdahl/SimpleVideoChat/tree/master/SimpleVideoChat/src>