NTNU – Trondheim
Norwegian University of
Science and Technology

Audio Interaction in the Auditorium

Arild Almås Berg

Master of Science in Computer Science
Submission date: Januar 2015
Supervisor: Trond Aalberg, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Preface

This Master Thesis is written at the Department of Computer and Information Science, NTNU, during the fall of 2014. The thesis is a part of the 5^{th} year of the master programme in Computer Science.

The title of the thesis is “Audio Interaction in the Auditorium”.

I would like to thank my supervisor Associate Professor Trond Aalberg at NTNU for guidance and advices throughout the semester.

Trondheim, 27 January 2015

Arild Almås Berg

Abstract

During lectures with many students and conferences with large audiences there is often a wish to establish a two-way communication between the speaker and the audience. Discussions, questions and audience participation normally enhance the value of a lecture or presentation. One barrier for a good two-way communication is that large auditoriums make it difficult for the whole audience to hear a random audience member speak. Today, some conferences support handout microphones or those who wish to speak may queue up in front of a stationary microphone, but this is not usually found at lectures in large auditoriums.

In this thesis, a proposal for a design and a prototype system that simplify the interaction between the lecturer and the audience is described. Today, almost everyone have access to a smart phone, tablet or other handheld device. The design presents a system where these handheld devices act as both microphones and clients to a server which is connected to any auditorium's speakers. The system is a type of interaction system, which focuses on the use of voice from clients to server.

A lecturer can use his or her personal computer, connect it to a speaker and run the server side of the system. Every student may use their own handheld device, connect it to the running server, and speak into the device as microphone for the auditorium. With this system, audience members are able to speak with a normal voice whether they are seated in the front or in the back of the auditorium. As the voice is transmitted through speakers, everyone are able to hear what is said without any need of repetition.

Today, almost all handheld devices support both WiFi and Bluetooth. The thesis gives an overview of the two technologies, and gives a reasoning for WiFi as the chosen the prototype solution. Other covered topics are range and connections. Testing of the prototype reveals a few challenges that should be further looked into. A complete system may improve the overall collaboration inside the auditorium.

Sammendrag

Under forelseninger med mange studenter tilstede og ved konferanser med mange deltakere er det ofte et ønske om om etablere en toveis kommunikasjon mellom foreleser og tilhørere. Diskusjoner, spørsmål og deltakelse fra tilhørerne øker normalt verdien av en forelesning eller en presentasjon. Et hinder for en god toveiskommunikasjo er at store auditorier gjør det vanskelig for alle de tilstedeværende å høre når en tilfeldig person snakker. Ved noen konferanser tilbys idag en håndholdt mikrofon som sendes rundt eller de som ønsker ordet må gå fram til en stasjonær mikrofon. Dette er som regel ikke tilgjengelig ved en forelesning i et større auditorium.

I denne oppgaven er det beskrevet et forslag til design for et prototype system som vil forenkle samspill mellom foreleseren og tilhørerne. Nesten alle har i dag tilgang til en smart-telefon, nettbrett eller lignende håndholdte enheter. Designet viser et system der disse håndholdte enhetene opererer både som mikrofoner og klienter som kan være knyttet til en hvilken som helst høyttaler i et auditorium. Systemet er en form for samhandlingssystem som fokuserer på bruk av stemmen fra klient til tjener.

En foreleser kan benytte sin egen datamaskin, knytte seg til en som snakker og styre tjenersiden av systemet. Hver student kan bruke sin egen håndholdte enhet, koble den opp til den aktuelle tjeneren og snakke i enheten som en mikrofon i auditoriet. Med dette systemet vil tilhørerne ha mulighet til å snakke med normal stemme enten de sitter foran eller bak i auditoriet. Siden stemmen blir overført gjennom høyttalere, kan alle høre hva som blir sagt uten behov for gjentakelser.

Nesten alle håndholdte enheter kan idag støtte både WiFi og Bluetooth. Oppgaven gir en oversikt over de to teknologiske løsningene og gir en begrunnelse for at WiFi velges som løsning for prototypen. Øvrige temaer som belyses er rekkevidde og tilkoblinger. Testing av prototypen avslører noen utfordringer som bør ses nærmere på. Et fullverdig system kan forbedre samhandlingen i auditoriet.

Table of Contents

List of Figures

1. Introduction

In today's lectures and conferences, almost everyone has a handheld device available, for example a smart phone or a tablet. The technology inside these devices can be utilized in more ways than their intended usage area. This thesis will examine the possibilities to develop a system which utilizes the microphones that exist in the different devices. The goal of this system is to create a communication channel where participants are able to speak into their own devices as a microphone and to transmit the sound to a set of suitable speakers in lecture and conference halls or auditoriums.

Another goal of this system is to promote student participation during lectures. A system which allows students to signal and speak through their handheld device instead of raising their hands and talk very loud, might remove some of the mental obstacles of talking in big crowds. Further the system and idea can contribute as a feature on a collaboration system where handheld devices are the main collaboration tools.

1.1 Objective

The overall objective of this master thesis is to explore the use of handheld devices in the communication between student and lecturer in the auditorium to support and facilitate discussion, question asking and student participation in general.

The project should include a state of the art study on relevant collaboration systems and technology, design and development of a prototype solution and evaluation.

1.2 Scope

A prototype solution that allows students to speak in to their own mobile phones for the entire class to hear via the speakers in the auditorium should be developed. Students who wish to speak must sign up on a list, via their phones, for the lecturer to open the communication line between one phone and the speakers. The reason for developing such a system is to make it easier for students and lecturers to hear and comprehend what is being said and asked inside the auditorium.

The prototype should consist of a server application, that lists all participants and gives permission to the one that is allowed to speak, and a client application which can be used by handheld devices. The server application should be connected to the speakers in the auditorium, and the client application should allow users, or clients, to log on to the server, sign up for questions and speak through the handheld device. Last, the audio of the prototype solution should be tested as a proof of concept.

2. Relevant Collaboration Systems and Technology

Many different systems that support interaction between lecturer and students exists today. Some systems are complete Learning Management Systems (LMS), while others can be recognized by the way students and lecturers interact with each other. Interaction and collaboration can happen via browsers, tablets, mobile phones, computers or other special devices. Most of the mobile phones in use today are smart phones. These, and tablets, have an increasing presence in auditoriums, in use by students as well as lecturers. Both smart phones and tablets have support for browsers, and as such it is of little importance which type of device is used during browser interaction. A unifying term for these devices are handheld devices or small, computing devices.

The basic definition of a Learning Management System "is a software application that automates the administration, tracking, and reporting of training events" (Ellis, 2009, p.1). With LMSs educators and students can communicate through chats and discussion forums, and there are possibilities to create and share assignments, profiles or quizzes.

<u>Sakai</u>

Sakai (Sakai 2014) is an open source LMS that support these features. With this system, educators and students can share their knowledge through discussion boards, chats and live participation in lectures, either online, face to face or in another blended environment. Sakai offers an intuitive, modern interface, resulting in a minimal learning curve for all participants.

<u>Kahoot</u>

Kahoot (Kahoot 2014) is a browser based system which can be used by any device with access to a web browser. With Kahoot, lecturers, students or other suitable persons are able to create quizzes, facilitate discussions or making surveys in real-time, also known as Kahoots. The system is set up through three steps, Create, Launch and Play. The educator creates his or her own Kahoot on a suitable device, for example a quiz based on the current lecture. Then, the Kahoot is launched onto a big screen, for the whole class to see. Everyone in the auditorium, or room that is used, can join the quiz through a browser on their own devices, and finally, when everyone is ready, the educator can start the Kahoot. Questions are answered in real-time from a user-friendly interface.

Bluetooth Interaction

In a paper by Bär, Häussge and Rössling (Bär, Häussge & Rößling (2007)) an interaction support system has been designed and presented. Although the system is old in the history of computers, the design and ideas are still applicable today. The paper presents a system that consists of an educator client and student clients, which communicates through web services. The educator client is run on a computer, and the student clients are run on PDAs or mobile phones. A Bluetooth extension was also tested, as this would be free of charge for students who did not want to pay for GSM usage.

Through their system, students and educator can interact via different plug-ins. The paper presents plug-ins that allow students to answer quizzes, evaluate the lecture, submit comments or questions, retrieve textual information or the current slide. The student client design is easy and makes it possible to view all interaction possibilities at once, minimizing the chances of a distracting application. The educator client can separate the different content into pages or tabs, making it possible to view what is considered most interesting and hide distracting elements away from the screen.

DIGVOTE III

Digivote III (Brähler Systems) is a standalone voting system that uses a special device to answer questions, quizzes, votes or other suitable surveys. The system consists of handheld wireless controls, containing a set of keys for transmitting answers, a transponder, and a software that can record the votes from the devices. In a learning context, the educator can run the software on his or her computer, and the students can answer quizzes or evaluations in real-time by using their given controls. This system is comparable to Kahoot in the way that they are both supporting interaction through quizzes or a voting system in real-time, like a game-show. The difference is that Digivote III is a standalone system with special devices, whereas Kahoot makes use of the possibilities that exists in today's personal handheld devices.

The following chapter will present this thesis' design and solution for an interactive support system. With the use of handheld devices explored, the system makes use of today's modern devices, such as smart phones, tablets and computers. Although the new system explores the use of audio transmission instead of textual, the way of communication between the devices is based on some of the ideas from the previously presented systems.

3. Design and development of a prototype solution

In this chapter, the concept, design and development of a prototype of the audio interaction system is presented. The choice of technology for transmitting data is also discussed and reasoned for.

Before discussing the choice of technology and concept, a few requirements to and information about the system are given. The audio interaction system is a communication tool used inside auditoriums during lectures, conferences or other presentations. The system is built upon the idea that any handheld mobile device can be used as a microphone, and that almost all attendees today have at least one such device available.

A system like this can replace existing microphone technologies where microphones are passed round to those who wishes to speak or ask a question. Another non-technological and time consuming solution consists of getting in queues for a stationary microphone. Areason for why an audio interaction system is developed is to improve the quality of discussions in auditoriums. In today's lectures, a student sitting on the first row in an auditorium might ask a question, which the lecturer is able to hear, while those who are seated in the back of the auditorium might not be able to catch. With a microphone and speakers at hand, everyone should be able to hear the question or comment. In another setting, a student at the rearmost seat in the auditorium can speak with a normal voice level into his or her handheld device instead of having to raise his or her voice to get through to the lecturer.

Audio interaction systems in the auditorium might therefore be helpful in both reducing the time spent to wait for an available microphone, and to improve the overall experience with sound, supporting discussion, question asking and participation in general.

From this idea, a few requirements must be set for the system. The most important is that the system should not delay or worsen today's communication inside auditoriums. Therefore, the system should be easy to understand, accessible and easy to use. The system should also be able to transmit all conversations as close to real-time as possible, meaning no delay. The question is; How should the sound be transmitted from a personal microphone to the speakers in the auditorium? This thesis presents a solution where all participants are connected to one server, which is connected to the speakers. The audio interaction system consists of handheld devices, acting as clients, and a computer, acting as the server, which can connect to the speakers. This means that only the server needs to be aware of all connected clients, while the clients do not need to communicate with each other. Such a solution makes it easy to set up the system in any auditorium where speakers are accessible.

3.1 Choice of technology

Today, almost all handheld devices support both WiFi (Diffen LLC) and Bluetooth (Bluetooth[2] 2015). Therefore, this section will give an overview of the two technologies, and give a reasoning for the one that is chosen for the prototype solution. Topics that are covered are range and connections.

3.1.1 Bluetooth

Bluetooth is a wireless radio technology that is built in to countless of products. With Bluetooth, paired devices are allowed to share information such as voice, data or music. It differs from other radio technologies by transmitting radio waves over much shorter distances than for instance radios or TVs. While a TV broadcasting system can reach out many kilometers, depending on the device, Bluetooth may broadcast information within distances up to 100 meters. However, these distances are primarily used in industrial contexts, and the most commonly used class of radio for mobile devices cover a range up to 10 meters (Bluetooth [1-3] 2015).

Two Bluetooth devices can wirelessly communicate with each other via short-range networks called piconets. A piconet is created by two connected devices, and may contain up to eight simultaneously connected devices, meaning that each device can connect to seven other devices in the same piconet. However, a device can belong to multiple piconets at the same time, meaning it can communicate with several devices in different piconets.

3.1.2 WiFi

WiFi is also a wireless radio technology that is found in most handheld devices today, and is normally used to get access to the internet through wireless access points. The typical range for WiFi is around 30 meters indoors with the 802.11b/g standard, but the newer 802.11n standard has even wider ranges. To further increase the range, it is possible to equip access points with strong antennas, or connect antennas together.

Unlike Bluetooth, a connection between two devices is not done by coupling them together, but instead they are both connected to the same WiFi router. The maximum number of connected devices in WiFi depends on the router, which can manage one to tens devices simultaneously.

3.1.3 Chosen technology

Both technologies are applicable for the solution, as devices are able to find and connect with one another. The server side of the audio interaction system is the only one that needs to communicate with, and know, all connected devices, therefore both Bluetooth and WiFi are considered as a wireless technology to be used. However, range is of high importance for this system, and WiFi clearly outperforms Bluetooth in this type of situation. Therefore, WiFi is the chosen technology for this thesis' solution.

3.1.4 UDP and TCP

Connections between devices on a WiFi network is done as a local area network, through sockets, which consists of an IP address and a port. A packet can be sent either through the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). The TCP is a lossless protocol, securing that all packets sent from one device are received in the correct order by the other device. The reliability of the TCP makes it appropriate to establish and maintain connections between communicating units, such as a server and its clients. The UDP is a connectionless protocol, which does not provide any reliability for sent datagram packets to be retrieved at its destination. With UDP, there is no need for an established connection beforehand, and the latency is very low, as no error-checking or guarantee for delivery is given. This makes UDP suitable for real-time systems, where delay is undesirable, such as the audio interaction system.

The system will use both protocols as both have some desirable strengths. The connection between a client and the server is established and maintained through the TCP. Simple requests and messages are also sent through this protocol. However, when a client is given the permission to speak, the sound is sent through the UDP. This choice is made to ensure that the conversation and speech is perceived as real-time. With both UDP and TCP, sockets use ports to ensure that the packets are sent to the right destination. The choice of port number shall be according to the Dynamic and/or Private Ports as defined by Internet Assigned Numbers Authority. These numbers range from 49152 to 65535 (IANA 2015).

3.2 Concept

The prototype design of the audio interaction system is described and presented within this section. Firstly, the architectural concept solution is undergone, revealing how the system should work in the auditorium. Secondly, the server side and the client side of the solution are presented as separate parts in the following sections. Finally, the communication between the two sides of the concept is gone through, describing what happens where, and completing the overview of how a session is run.

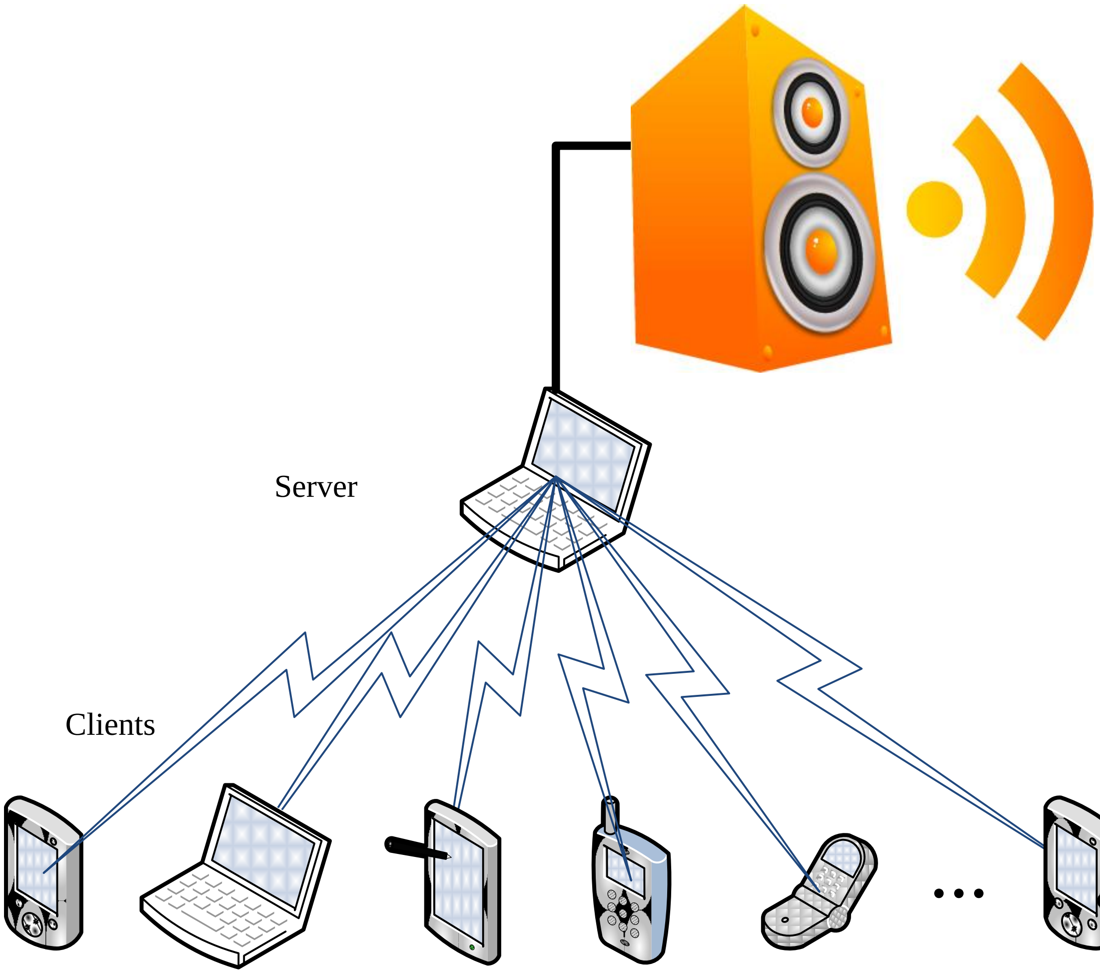

Figure 3-1 Architectural concept of the audio interaction system

Figure 3-1 shows the architectural concept of the audio interaction system of how the clients and the server are connected to each other. The solution consists of one server and multiple clients, each connected to the server via Wi-Fi or internet, and a speaker system. Since the aim of the system is to let clients use the speakers in the auditorium via their own handheld devices, the server must be connected to the appropriate speaker system, allowing any client to make use of the speakers. If a client is allowed to speak by the server, the server reads all datagram packets received from this client, and retransmits the sound to the speakers.

This architecture is quite simple and effective. Any lecturer can use their own personal computer, connect it to a speaker system and run the server side of the system. Every student may use their own handheld device, connect to the running server, and speak into the device as if it is a microphone for the auditorium. This is the idea behind the audio interaction system. Instead of passing round a couple of microphones to the ones who wish to talk, the "microphone" is already at hand within their own mobile phone or other handheld device.

3.2.1 Server side

This section presents and describes a suggested user interface for the server side of the audio interaction system. The server side is designed as a stand-alone desktop application, which can be run from any computer. Considering the statement that the complete system should be easily accessible, for both presenter and listeners, this solution meets this expectation. It is the presenter, or lecturer, that will use this part of the system.

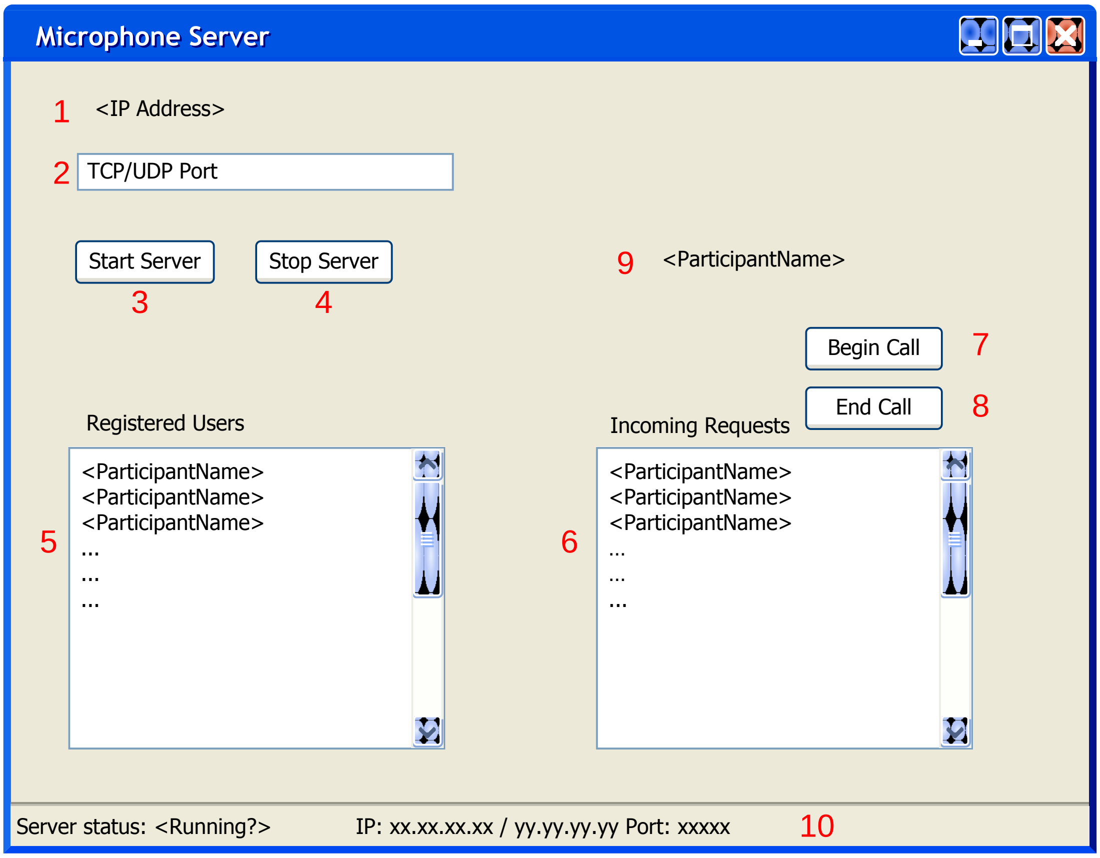

Figure 3-2 Server side desktop template

Figure 3-2 presents a proposal for the user interface of the server side of the system. The interface consists of two information fields, one text field, four buttons, two lists and one status bar. The first information field (1) and the text field (2) represent the connection settings necessary to establish connections between the clients and the server. The IP address in the information field shall be retrieved automatically when launching the application, and displays both global and local IP addresses if available. In the text field, the TCP and UDP port is manually set in the range between 49152 and 65535.

The "Start Server" button (3) is used to start the session, and opens the connection for incoming requests from clients, while the "Stop Server" button (4) shuts down the server session and closes all connections.

The two lists are used to display all connected clients, also known as participants. In the list "Registered Users" (5), the participants are alphabetically listed at all times, not considering whether they are idle, requesting to speak or speaking. The other list, "Incoming Requests" (6), contains a list of all participants who are currently wishing to speak. The list is ordered with a first come, first served queue, meaning that the oldest unattended request is on the top.

Two buttons are cooperating together with the "Incoming Requests" lists, namely the "Begin Call" and "End Call" buttons (7 and 8). To begin listening for datagram packets at the UDP port, and hence allow a client to speak, the client must be selected from the list, and the "Begin Call" button must be clicked. This will result in the client being removed from the list, and its name displayed in the information field (9). Clicking "End Call" will terminate the ongoing connection between the client and the server, and remove the client from the information field. When no one is speaking through the server, nothing is shown in this information field.

At the bottom of the application, a status bar (10) shows the status of the server, whether it is running or not, and the current IP addresses and port.

3.2.2 Client Side

In this section, a suggested design for the client side of the system is described. The client side differs from the server side, in the way that it is designed for mobile or handheld devices. This corresponds with the main reasoning to create this system, as the possibilities with handheld devices are explored. Inside the auditorium, it is the listeners, or students, that will use this part of the system.

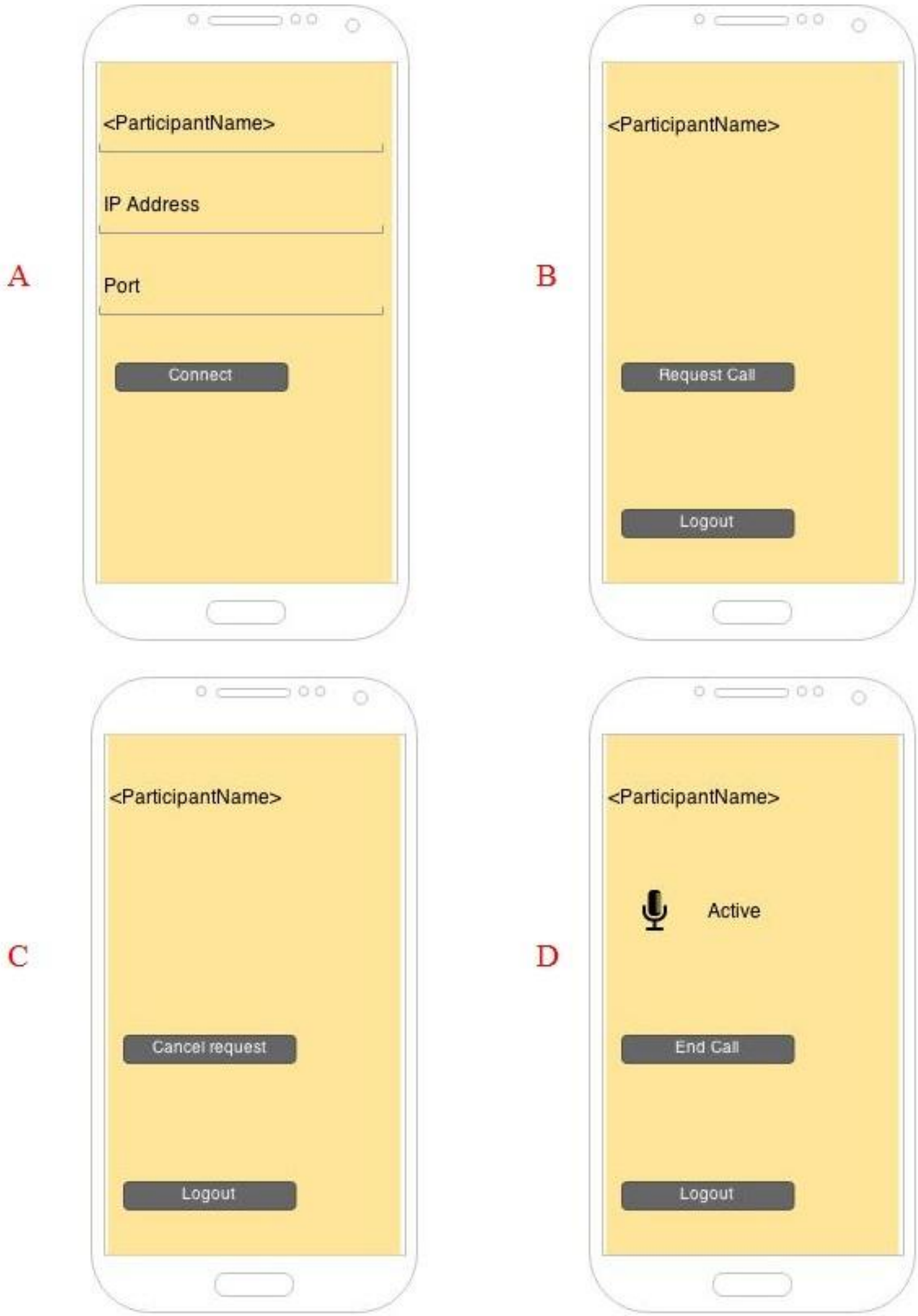

Figure 3-3 Client side mobile application template

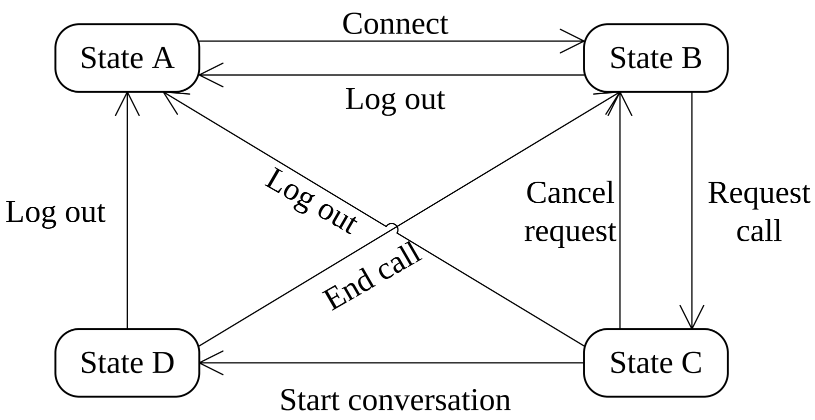

Figure 3-4 Client side state diagram

Figure 3-3 presents a suggestion for the user interface of the client side of the system at a handheld device. Different from the server side, the client side consists of different pages, each representing a different state the client is in. The transitions, with necessary actions, between the states are shown in Figure 3-4.

Upon opening the client side on a handheld device, the user is presented with a login-screen (A). The user must select a user name, preferably his or her own name, and enter the IP address and port number given by the server administrator. Clicking "Connect" will connect the user to the server, and display screen B.

In state B, the user has the possibility to make a request to speak by clicking the "Request Call" button. Clicking this button will send a message to the server and if successful, display screen C. From this state, the user may at any time cancel the previously sent request, taking him or her back to state B.

When a user is given the permission to speak, screen D is displayed. Being in this screen, the client microphone is on, and the client is sending all sound to the server. The user may choose to end the call, hence closing the sound connection, turning off the microphone and being sent back to state B. All this is done by clicking the "End Call" button.

To completely disconnect from the server, the user has the possibility to log out from any of the logged in states. The difference between logging out from an active state (D) or a waiting state (B or D) at the client side is within the handling of turning off the handheld device's microphone. All other responsibilities, concerning the connection and request handling, lies at the server side. By logging out, the user is sent to state A.

3.2.3 Connection between Server Side and Client Side

This section describes how the connection between a client and the server is working at the user level. The different functions and calls are gone through at a high-level design overview to explain the basics of the audio interaction system's functionality. The different possibilities are connecting a client to a server, requesting or cancelling a request to speak from the client side, beginning a call from the server side, logging out from the server or shutting down the server. Figure 3-5,Figure 3-6,Figure 3-7 and Figure 3-8 show simple sequence diagrams to better explain the communication between a client and the server. For continuity reasons, the states presented in section 3.2.2 are still applicable in this section.

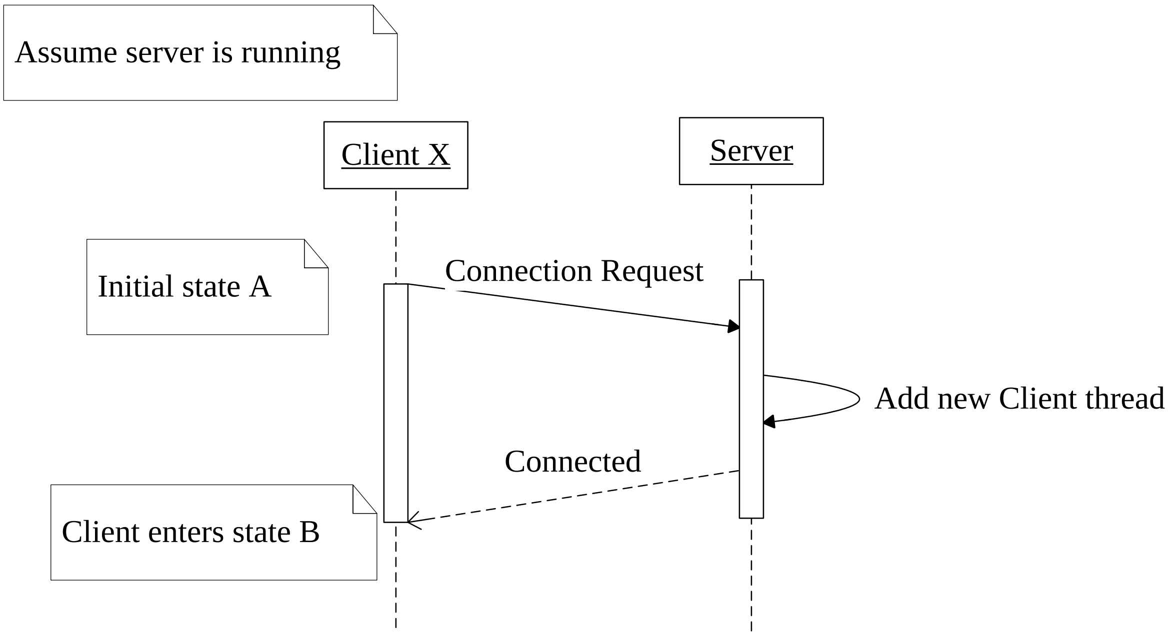

Figure 3-5 New client connecting to server

Figure 3-5 shows what happens when a client tries to connect to the server, which is assumed to be up and running. When the server gets a client request to log in to the session, the server creates a new client ID and a new thread for this client. This is done, so the server is able to maintain control over all connected clients. The connecting client is then added to the list "Registered Users" with all other clients. A response is sent back to client, who enters state B and is presented with the belonging page.

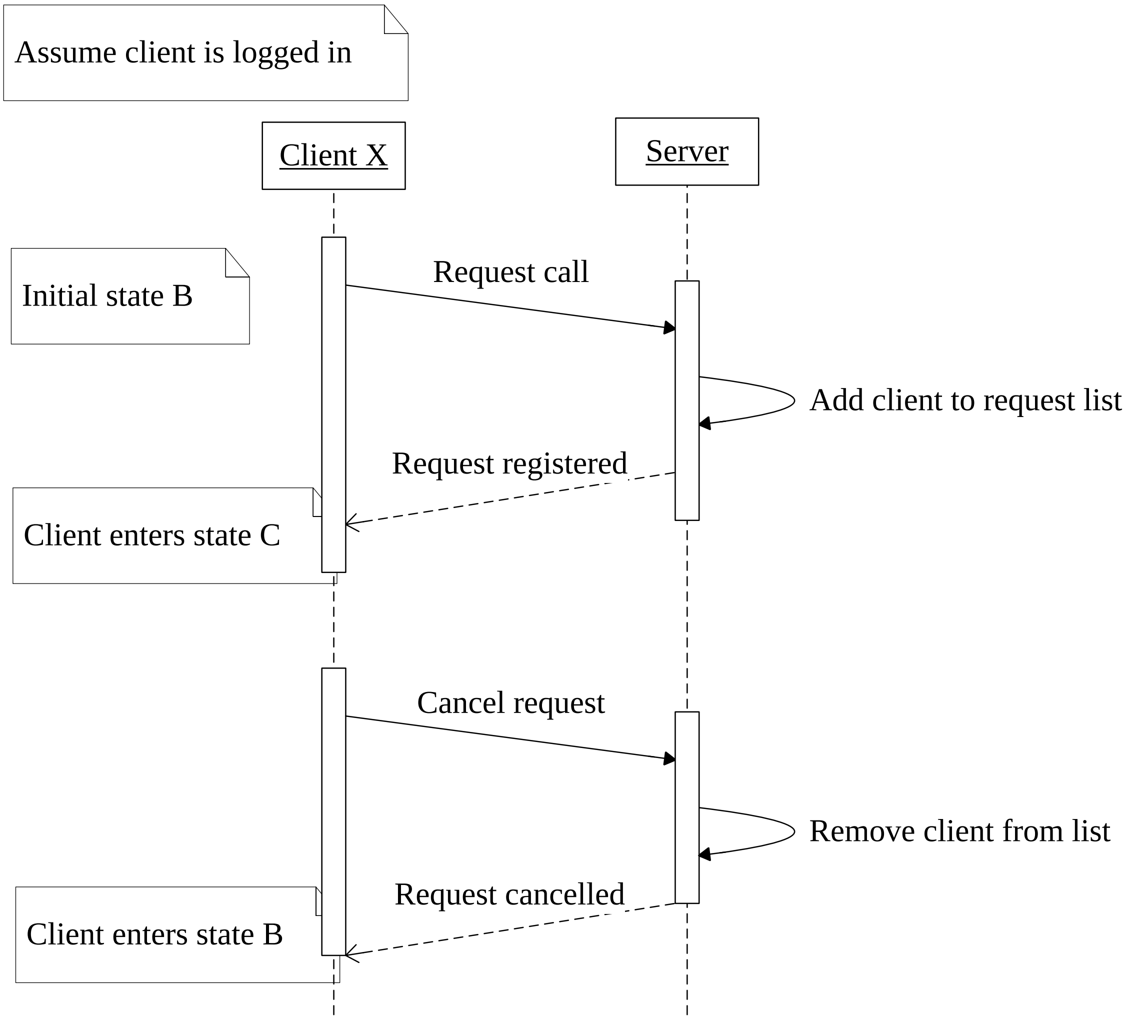

Figure 3-6 Client makes and cancels request to make a call

After a client is connected to the server, he or she might want to ask a question, discuss a topic or say something in general through the audio interaction system. To do this, the client must make a request to the server, which then adds the requesting client to the list "Incoming Requests", as shown in Figure 3-6. After the request is registered, the client is notified and enters state C. In this state, the client has the option to wait for his or her turn to speak or to cancel the request. Cancelling the request is done similarly as making a request, meaning when the server receives the cancellation, the server removes the client from the "Incoming Requests" list and notifies the client of the change. The client has now returned back to its previous state, state B.

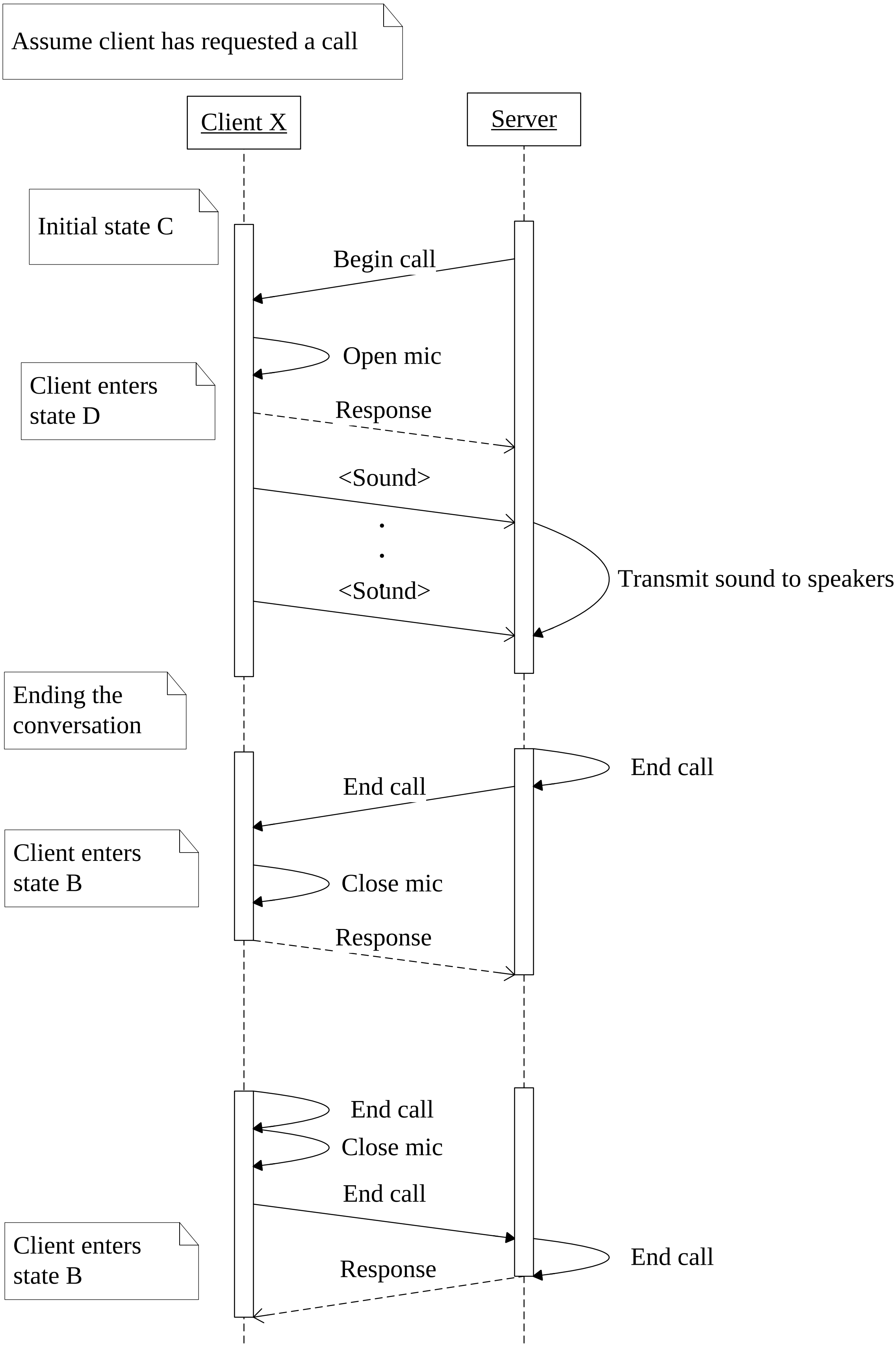

Figure 3-7 A call is initiated by the server, and can be ended from either server or client

The main functionality and operation with the system is to make speakers available for any client connected to the server. The procedure for allowing a client to use the speakers are shown in the upper part of Figure 3-7. It is assumed that the client who is about to communicate through the system has requested a call. The server administrator selects the client and clicks "Begin Call" as mentioned in section 3.2.1. After the call has been initiated, the server moves the client from the "Incoming Requests" list to the "Active caller" information field, starts to listen for incoming datagram packets and tells the client to open its microphone as the client enters state D. From this state all recorded sound in the device is transmitted live to the server, which further transmits the sound to its connected speakers. The transmission is kept alive until either the server or client ends the conversation.

There are two ways to end a conversation between the server and the active client. The first is from the server side, shown in the middle of Figure 3-7, while the second one is from the client side, shown at the bottom of the figure. When the server side ends the call, the server stops to listen for incoming datagram packets and sends a signal to the active client to end the call. The client then closes its microphone and returns to state B. If the client is the one to end the call, the procedure is almost identical, but the necessary methods to end a call at either side of the conversation are now invoked by the client. The reasoning for letting either side close the conversation is to let both speaker and listener decide if they want to end speaking.

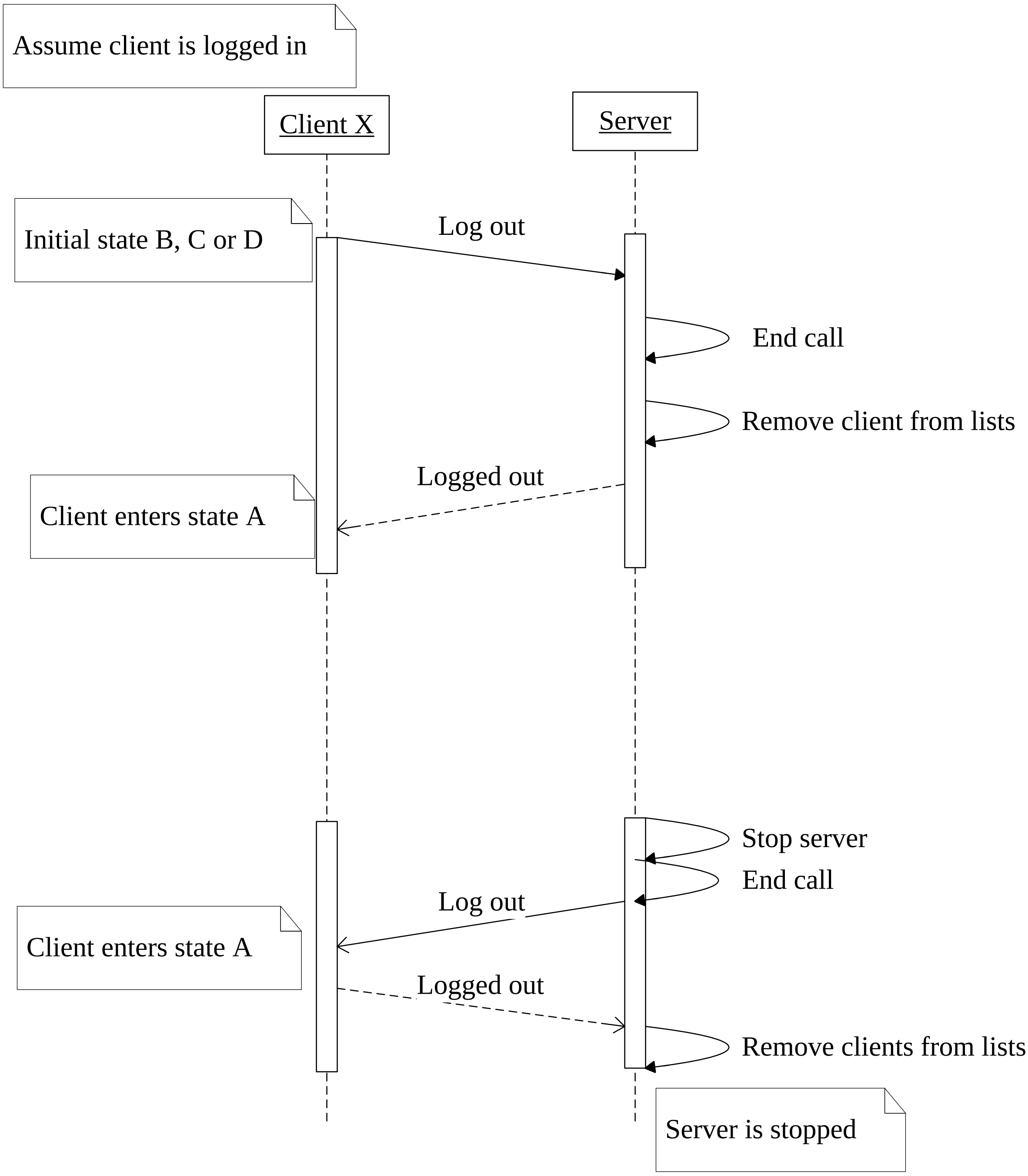

Figure 3-8 Closing a connection from a client or ending the session at server side

The last part of the connection handling between the server and clients is about ending a session from the server side or logging out from the client side. Figure 3-8 shows both of these possibilities. The first possibility is for a client to log out from the server. It is assumed that the client is in either state B, C or D. The client sends a message to the server, informing that it wants to log out. When the server receives this message, it checks whether the client is in a call, and if it is, it runs the procedure for ending that call. Later, the server removes the client from all lists and returns a closing message to the client which is then logged out. The client is now back to stage A, ready to log in for a possible new session.

The second procedure shown in Figure 3-8 is for the ending of a session, when the server shuts down. When the presentation or class is over, or for any reason the server administrator has to shut down the server, the following procedure is conducted. The server checks whether there is any client currently performing a call, and if it is, the server runs the end call procedure with this client. Then, a message is sent to all connected clients, logging them out and returning them to state A. As the responses from the clients are returned, they are removed from all lists and the server is stopped.

3.3 Basic communication system

Based on the proposed design, the following basic communication system has been developed. The communication system consists of a server running on a computer and a client running as an Android application on any Android device. The system is a stripped down version of the concept, only communicating through the UDP.

```
DatagramSocket serverSocket = new DatagramSocket(port);

byte[] receiveData = new byte[4096];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

ByteArrayInputStream baiss = new ByteArrayInputStream(receivePacket.getData());

while (true) {
    serverSocket.receive(receivePacket);
    ais = new AudioInputStream(baiss, format, receivePacket.getLength());
    toSpeaker(receivePacket.getData());
}
```

Figure 3-9 The basics of the server side

Figure 3-9 shows the most necessary functionality of the server side to receive data from clients through UDP. A datagram socket is created with a preset port, the same port which is set in the Microphone Server in Figure 3-2, which the server will receive packets through. The byte array is set to determine the size of the buffer for the incoming packets. The size of this array has effect on both quality and delay of the received sound. A higher byte size will increase the latency from when packets are received until they are played through the speakers. However, higher byte sizes improves the quality of the sound, resulting in a trade-off between quality and delay. The formula for delay is given as:

$$Delay = \frac{byte\ size}{sample\ rate} * 2$$

The sampling rate is set to 48000 Hz as recommended by the Audio Engineering Society (AES) (AES5 (2014)). A byte size of 4096 will result in a speech delay of 170 ms, not considering the transmission time from client to server.

```
DatagramSocket socket = new DatagramSocket();
byte[] buffer = new byte[minBufSize];
DatagramPacket packet;
final InetAddress destination = InetAddress.getByName("192.168.0.101");
recorder = new AudioRecord(MediaRecorder.AudioSource.MIC,sampleRate,
        channelConfig,audioFormat,minBufSize*10);
recorder.startRecording();

while(status == true) {
    recorder.read(buffer, 0, buffer.length);
    packet = new DatagramPacket (buffer,buffer.length,destination,port);
    socket.send(packet);
}
```

Figure 3-10 The basics of the client side

Figure 3-10 shows the most necessary functionality of the client side of the system to send datagram packets to a server via UDP. As with the server, a socket and a packet must be instantiated. An IP address and port must be set as destination of sent packets. These are equal to those set on the client side in Figure 3-3. The minimum buffer size is set according to the sample rate, channel configuration and audio format, and determines the size of the buffer where the audio data is written during recording. This size guarantees a successful creation of the AudioRecord object. The client application must record the microphone on the running device, set a sample rate similar to the sample rate of the server side, an audio channel configuration, audio format encoding and a buffer size for where the audio is written. To send the recording live to the server, the packets are sent as soon as the buffer length is reached.

4. Evaluation

A prototype system has been designed, and a basic communication system has been developed to test the sound transmission between client and server. In this case, the test was performed between an Android application and a server running on a computer connected to speakers. This is similar to the purpose of the system, where handheld devices are connected to a server which is connected to speakers. It is important to notice that the system should not decrease the quality of the lecture as it is today, but rather improve the overall experience. The system should therefore not consume unnecessary time or interrupt the natural flow of a lecture and conversation.

The testing revealed two challenges with the system that should be further looked in to. The first challenge is the trade-off between quality of sound and delay. With the sampling rate and buffer size set according to section 3.3, the calculated delay was 0.17 seconds, and the sound quality was perceived as slightly broken. This means that the sound could be better and clearer, and the delay is considered acceptable, as only those close to the microphone are able to hear the delay. Those sitting far from the microphone was not able to hear the delay, as they did not hear the speaker talking in to the microphone.

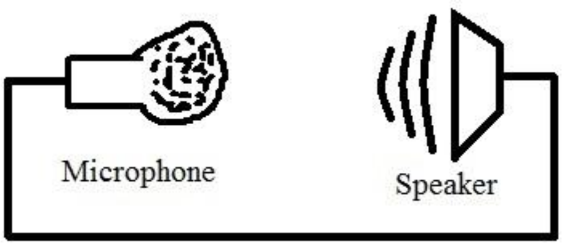

Figure 4-1 Audio feedback loop

The other challenge found is about audio feedback. Audio feedback is a ringing or screeching sound that sometimes are present in systems with microphones and speakers. It is caused when noise is looped between a microphone and a speaker. The microphone transmits sound to the speaker, and then re-records the sound emitted through said speaker, creating a continuous loop, as shown in Figure 4-1.

Comparing the designed prototype system with the other interaction systems presented in Chapter 2, shows that there are differences in both types of data that are transmitted and how the system works between server and clients. This thesis' system works by setting up a server which receives data which is directly retransmitted to speakers. The difference between this and the Bluetooth Interaction system (Bär et. al (2007)) is that Bär et. al's system has a server which both students and educators are connected to, with two different types of clients. The Digivote III system (Brähler Systems) is similar to this thesis' concept solution, in the way that there are several handheld devices acting as clients in a stand-alone system, who all communicate with a server running on a computer. However, Digivote III communicates via a radio frequency and not through WiFi.

Both the designed and developed system are prototypes to prove that the idea works. When further developing a full system there are some alternative solutions or features that should be considered. During a lecture, there are different situations in which a student wishes to speak. The student might have a single question in the middle of the lecture, he or she may have a wish to discuss the lecture as a whole at the end of the lecture, or there may be another reason for why a student wants to speak. To support the various types of participation, the system could provide different lists referring to one type of participation each, making it easier for a lecturer to chose which student should get the permission to speak. In the prototype design, only one list is provided for every type.

To further expand the system, ideas and solutions from the other presented systems in chapter 0 could be included. The solution presented by Bär et al. with a comment or question section, could lower the bar for students to ask a question anonymously, and for the lecturer to answer those when convenient. This can save time, and make the lecture run smoothly as there are no pauses from the lecturers side. Polls or quizzes such as Kahoot can also be featured in an expanded system to increase the alternatives an interaction system between lecturer and students supports.

A fully developed system should consider the availability for both students and lecturer. The developed prototype supports only handheld devices which use the Android operating system. Not all handheld devices support Android, there are for example also iOS and Windows phones on the market, and the system should be available for these as well. Different solutions to this problem could be to develop similar applications for all systems, or create a browser based client side which can be accessed on the local network or internet.

5. Concluding remarks

In the end, a prototype of an audio interaction system has been designed and developed, and it is audio that is the focus of this system. Therefore, neither chat, discussion forum or support for quizzes have been considered as part of this thesis. For future remarks, if a system is to be further expanded and developed, it is important to remember that audio is still a main feature of the system.

The focus of further development should be to make the system available for all common devices and platforms, and investigate and improve the overall sound quality. This will result in a system that is dependable and easy to use and learn inside the auditorium. Later, the system should also be modifiable to easily add or remove features that are suitable for special types of interaction in an auditorium.

This system can improve lectures and other presentations where audience participation is encouraged, as two-way communication is made easier from the audience perspective. The system also distributes the voice of a single person to the entire auditorium without anyone needed to repeat what was said if a microphone was not available. This eases the challenge of being an active participant in a discussion or lecture, both as speaker and listener.

6. References

1. AES5. (2014). *AES recommended practice for professional digital audio - Preferred sampling frequencies for applications employing pulse-code modulation (revision of AES52003)*. Audio Engineering Society, inc. Accessed 26.01.2015.
2. Bär, H., Häussge, G., & Rößling, G. (2007). *An integrated system for interaction support in lectures*. Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education, *37*(3), 281-285. doi: 10.1145/1268784.1268865
3. Bluetooth[1]. (2015). *Bluetooth Basics*. Accessed 19.01.2015, http://www.bluetooth.com/Pages/Basics.aspx
4. Bluetooth[2]. (2015). *Mobile Telephony Market.* Accessed 20.01.2015, http://www.bluetooth.com/Pages/Mobile-Telephony-Market.aspx
5. Bluetooth[3]. (2015). *Fast Facts*. Accessed 20.01.2015, http://www.bluetooth.com/Pages/Fast-Facts.aspx
6. Bluetooth[4].(2015). *Bluetooth Technology and WiFi*. Accessed 21.01.2015, http://www.bluetooth.com/Pages/Wi-Fi.aspx
7. Brähler Systems. *Digivote III*. Accessed 22.01.2015, http://www.braehler.com/DIGIVOTE_III2.html
8. Diffen LLC. *Bluetooth vs. WiFi*. Accessed 21.01.2015, http://www.diffen.com/difference/Bluetooth_vs_Wifi
9. Ellis, R. K. (2009). *A Field Guide to Learning Management Systems*. Learning Circuits - American Society for Training & Development (ASTD).
10. EngineersGarage. *Difference between Bluetooth and WiFi*. Accessed 21.01.2015, http://www.engineersgarage.com/contribution/difference-between-bluetooth-and-wifi
11. Griffin, R. (2015). *Should my wireless audio system use Bluetooth or WiFi?*. Accessed 21.01.2015, http://wrensound.com/should-my-wireless-audio-system-use-bluetooth-or-wifi/
12. IANA. (2015). *Service Names and Transport Protocol Port Number Registry.* Accessed 17.01.2015, http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
13. Kahoot. (2014). *Kahoot!*. Accessed 22.01.2015, https://getkahoot.com/
14. Sakai. (2014). *Sakai*. Accessed 22.01.2015, https://www.sakaiproject.org/

7. Literature

Bär, H., Rößling, G., Tews, E., & Lecher, E. (2006). Bluetooth Interaction Support in Lectures, *Proceedings of Mobile Learning*, 360-364.

Brehm, J., Brancovici, G., Müller-Schloer, C., Smaoui, T. & Voigt, S. (2003). Experimental Tools for a Multimedia-Supported Interactive Lecture, *DeLFI*, 85-94.

Chen, L-J., Kapoor, R., Jung, S., Sanadidi, M. Y. & Gerla, M. (2009). An adaptive ARQ timeout approach for audio streaming over Bluetooth, *International Journal of Wireless and Mobile Computing*, *3*(4), 255-264. doi: 10.1504/IJWMC.2009.029345

Crozier, W. R. (2004). Shyness and Students' Perceptions of Seminars, *Psychology Learning & Teaching*, *4*(1), 27-34. doi: 10.2304/plat.2004.4.1.27

Foster, L. N., Krohn, K. R., McCleary, D. F., Aspiranti, K. B., Nalls, M. L., Quillivan, C. C., . . . Williams, R. L. (2009). Increasing Low-Responding Students' Participation in Class Discussion, *Journal of Behavioral Education*, *18*(2), 173-188. doi: 10.1007/s10864-009-9083-8

Liang, Y. D. (2006). *Introduction to Java Programming - Comprehensive Version* (6th Edition). Prentice Hall.

Mulryan-Kyne, C. (2010). Teaching large classes at college and university level: challenges and opportunities, *Teaching in Higher Education*, *15*(2), 175-185. doi: 10.1080/13562511003620001

Rocca, K. A. (2010). Student Participation in the College Classroom: An Extended Multidisciplinary Literature Review, *Communication Education*, *59*(2), 185-213. doi: 10.1080/03634520903505936

Schon, D., Klinger, M., Kopf, S. & Effelsberg, W. (2012). Mobilequiz - A Lecture Survey Tool Using Smartphones and QR Tags, *International Journal of Digital Information and Wireless Communication*, *2*(3), 231-244.

Shneiderman, B. & Plaisant, C. (2009). *Designing the User Interface, Strategies for Effective Human-Computer Interaction* (5th Edition). Pearson Education.

Waite, W. M., Jackson, M. H. & Diwan, A. (2003). The conversational classroom, *SIGCSE '03 Proceedings of the 34th SIGCSE technical symposium on Computer science education*, *35*(1), 127-131. doi: 10.1145/792548.611950

Wang, Y., Huang, W. & Korhonen, J. (2004). A framework for robust and scalable audio streaming, *Proceedings of the 12th annual ACM international conference on Multimedia*, 144-151. doi: 10.1145/1027527.1027554

Xie, W., Lu, C., Zhang, Z., Lin, Y. & Shen, R. (2011). PPClass-M - A live online lecture system optimized for mobile access. *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*(1-5). Location: Nuremberg

Zhang, Q., Wang, G., Xiong, Z., Zhou, J. & Zhu, W. (2004). Error Robust Scalable Audio Streaming Over Wireless IP Networks, *IEEE Transactions on Multimedia*, *6*(6), 897-909. doi: 10.1109/TMM.2004.837249