

CSC/ECE 575 INTRODUCTION TO WIRELESS NETWORKING

FINAL PROJECT REPORT: BLUETOOTH WALKIE-TALKIE (A4)

Spring 2017

Name 1: OLGA NAM

Name 2: MANOJ SHARAN GUNASEGARAN

Name 3: PETER CHEN

Name 4: SIDDHARTHA KOLLIPARA

Table of Contents

Abstract	3
Introduction	3
Project Description	3
Deliverables	3
Related Work	3
Application Design and Implementation	4
GUI	4
Flow Diagram	5
Bluetooth/Audio	6
Experimental Results	7
Milestones and Work Breakdown	8
Practical Problems	10
Future Improvements	10
References	11

1. Abstract

A Walkie-talkie is a two way radio that allows communication between two or more parties. There are several accessories and devices that can become Walkie talkie. This project focuses on how to make two mobile android devices function as walkie-talkies over a bluetooth connection. There are two assumptions that are made for the environment: the devices are already aware of each other (paired prior to app launch) and they are within communication range of each other. Communication will take place between the devices similar to a push-to-talk application. This means that when one device is sending audio data, the other device is receiving (not transmitting) and vice-versa. Our Walkie-Talkie app accomplishes every functionality outlined above.

2. Introduction

2.1. Project Description

The goal of this project is to make two mobile android devices function as walkie talkies. There are two assumptions that are made for the environment: the devices are already aware of each other (paired in Bluetooth) and they are within communication range. Communication will take place between the devices as push-to-talk. This means that when one device is sending audio data the other device is receiving and vice-versa.

For this project we are creating an Android application that will allow two devices to communicate over Bluetooth connection when network signal is weak or not available. We assume that the devices are paired prior to launching the app.

2.2. Deliverables

Project Reports
Demonstration
Well commented code (Github)

2.3. Related Work

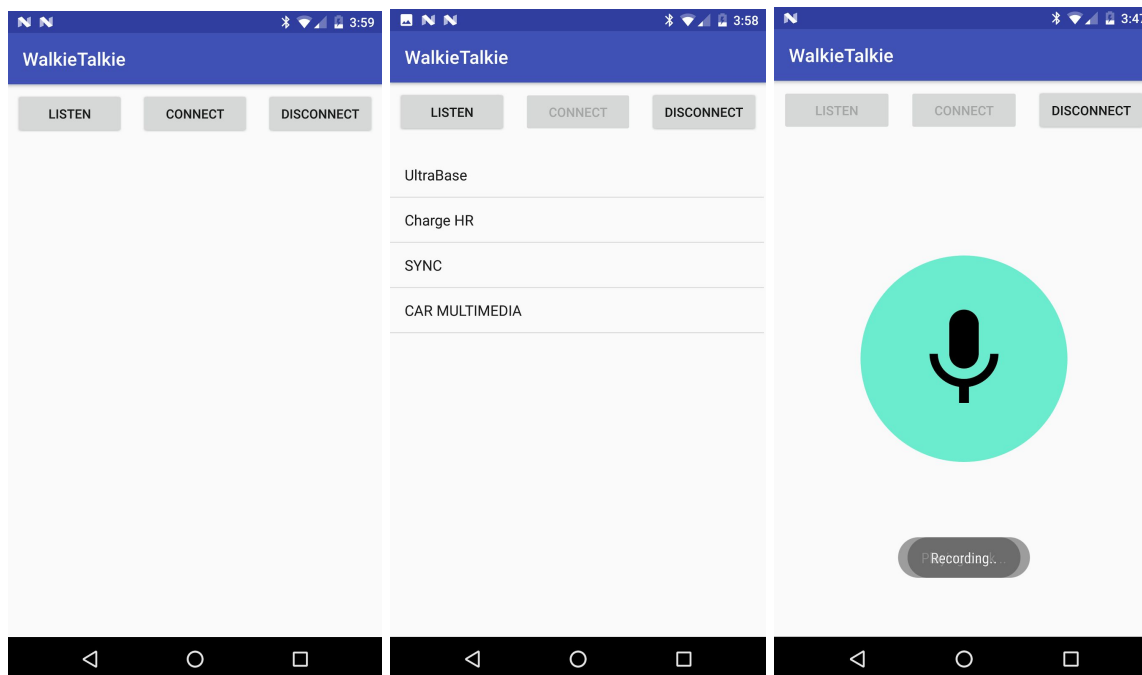
This project is a continuation of the effort by another team in Fall 2011 to implement Android based Walkie Talkie. In their solution this team connected the phones over WiFi. In our project we are connecting the phones over Bluetooth which eliminates the need for network connection all together and makes this application more practical for areas with weak signal or no signal.

3. Application Design and Implementation

3.1. GUI

We built a simple GUI. On app launch, at first we see a screen with three buttons: Listen, Connect and Disconnect. When Connect is pressed a list of all the paired devices is shown. Here the user picks who to connect to. On the other side the second user would press the button Listen to accept the connection.

After they successfully connect they navigate to a screen with Talk button in the middle and buttons Listen, Connect and Disconnect will be displayed on the top. Listen and Connect buttons are greyed out on this screen. The list of the paired devices is hidden at this point. Pushing the green Talk button will call a method to record and send audio and releasing it will stop recording and transmission.



3.2. Flow Diagram

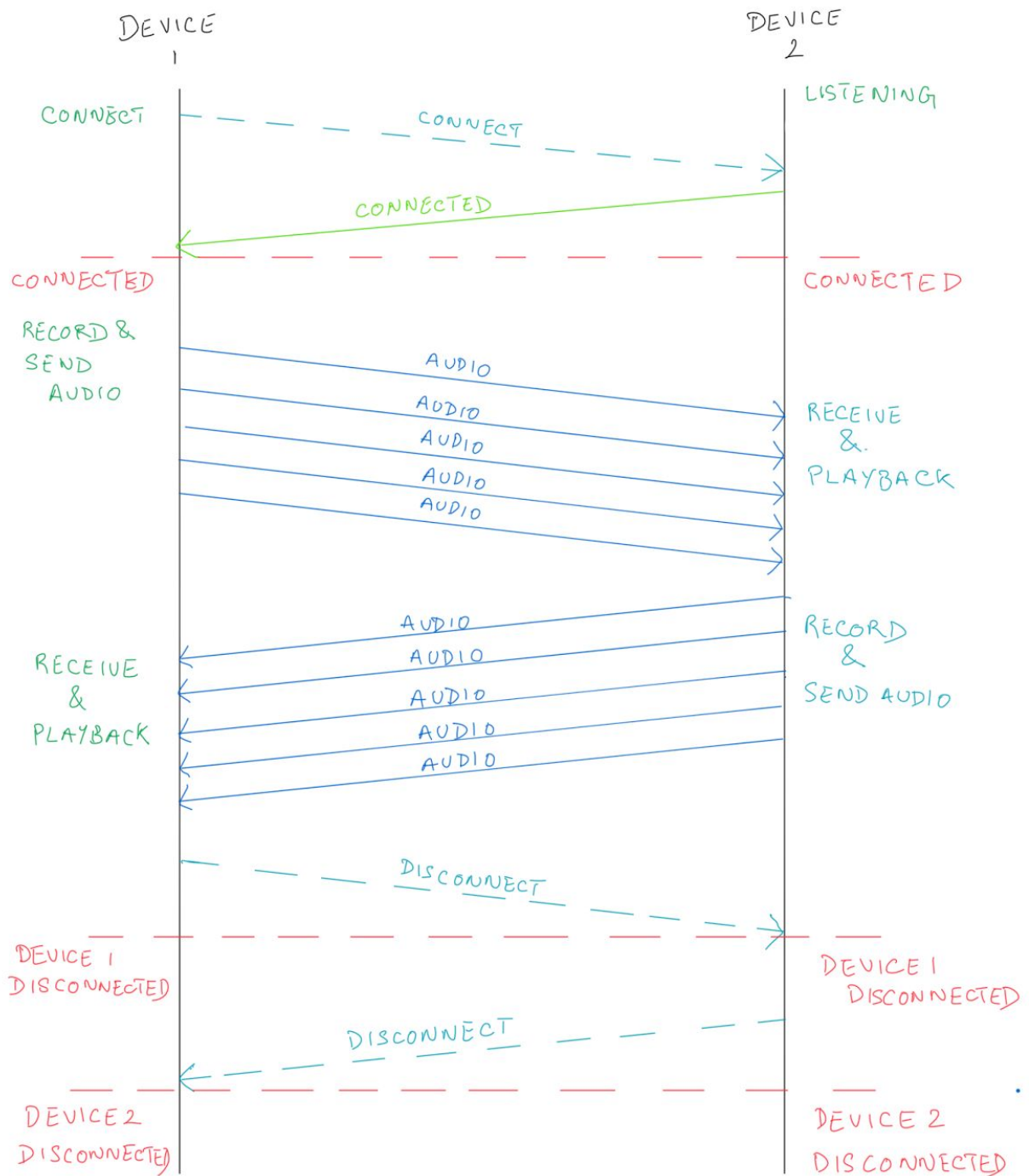


Figure 1. Application Flow

It is assumed that the devices are already paired prior to opening up the app. Then, a user (device 2, in this case) clicks on the 'LISTEN' button while the other user (on device 1) clicks on the 'CONNECT' button. Device 1 then lists every paired device that it has in memory. The user will then click on the device name that he wants to connect (device 2, in this case). This will establish a connection between both devices and set up a socket. A microphone button is then displayed on both devices. If the user on device 1 (or device 2) pushes this button, audio is recorded and transmitted over to the other device, where it is automatically played back. To disconnect, the user simply taps the 'DISCONNECT' button at anytime and that device is automatically disconnected.

3.3. Bluetooth/Audio

The implementation of Bluetooth was completed in several steps. We started by creating a class to hold the device names and MAC addresses of paired devices. Methods for connecting and listening for sockets were implemented in a separate class and returned a true value on success. The listen and connect methods are invoked when their respective buttons are clicked in the GUI. The listview is populated with the information contained in the device objects and displayed in the ListView object. The sockets are closed by running a method in the connect/listen classes and are called by clicking the disconnect button. The manifest required the inclusion of Bluetooth permissions to allow use of Bluetooth.

The audio methods are implemented in a class called *MainConversation*. The two main methods in this class are *startRecording* (sending data) and *startPlaying* (receiving data). When the Bluetooth connection is successful between two devices, the first action called is *startPlaying*. This is because the default action of a device should be to listen when not actively sending data. The audio microphone button appears when the connection is successful and will allow data transfer when held down. At this time *startRecording* is invoked and will record audio on the active user's device. That data is transmitted as byte data through a buffer to the other user (listener). When the button is released, the recorder device will be halted and the listening method will become active again. The manifest requires the inclusion of microphone permission to allow the use of audio features.

4. Experimental Results

Sr. no	Testcase	Expected output	Actual output
1	Test bluetooth connection	Make sure two devices which were previously paired - are able to connect	Both devices connected successfully
2	After connection is established test if audio packets are being sent and received	The receiver is able to hear the sender's message	The receiver is able to hear the sender's message
3	Make sure that when connection is lost the application exits gracefully	When the two devices are moved out of range during the conversation they both receive the warning and the application moves to the first screen of the application	Bluetooth connection disconnected and application moves to first screen.
4	Connecting to the device that is not in range	The application fails to connect and handles the exception properly	The application fails to connect and handles the exception properly
5	Test the disconnect feature of the application	When one of the participants presses disconnect, the connection is terminated and the app redirects to the first screen of the application.	The connection is terminated and the app redirects to the first screen of the application.

5. Milestones and Work Breakdown

Figure 2 outline the list of tasks and their corresponding assignment as well as duration and the relative risk of every task.

	Task Name	Duration	Start	Finish	Predec	Resource Names	Risk
1	• Walkie Talkie		Thu 2/23/17				1- low
2	• Design Brainstorming	55.13 days	Thu 2/23/17	Mon 3/13/17			
3	Setting up env	9 days	Thu 2/23/17	Mon 3/6/17		Manoj_Olga_Peter_Siddhartha	1
4	GUI	5 days	Tue 3/7/17	Mon 3/13/17		Olga	1
5	Bluetooth	5 days	Tue 3/7/17	Mon 3/13/17		Siddhartha__Peter	1
6	Audio	5 days	Tue 3/7/17	Mon 3/13/17		Manoj	1
7	• Product Development	109.13 days	Tue 3/14/17	Wed 4/19/17			
8	• GUI	28.13 days	Tue 3/14/17	Thu 3/23/17			
9	Mock up of the app	3 days	Tue 3/14/17	Thu 3/16/17	4	Olga_Manoj	3
10	Finilize the flow of the app	2 days	Fri 3/17/17	Mon 3/20/17	9	Olga_Manoj	3
11	User selection	3 days	Tue 3/21/17	Thu 3/23/17	10	Olga_Manoj	3
12	Communication window (talk/listen/diconnect)	3 days	Tue 3/21/17	Thu 3/23/17	10	Olga,Manoj	3
13	• Bluetooth/Audio	109.13 days	Tue 3/14/17	Wed 4/19/17			
14	Echo audio (single device)	9 days	Tue 3/14/17	Fri 3/24/17	5	Peter	5
15	Establish a connection	10 days	Tue 3/21/17	Sun 4/2/17	6	Siddhartha	5
16	Send text between devices	6 days	Mon 4/3/17	Sun 4/9/17	15	Siddhartha__Peter	4
17	One way communication	3 days	Mon 4/3/17	Wed 4/5/17	15	Siddhartha_Peter	4
18	Two way communication	10 days	Mon 4/3/17	Thu 4/13/17	15	Siddhartha_Peter	4
19	Two way communication with mic blocking	4 days	Fri 4/14/17	Wed 4/19/17	18	Olga_Manoj	4
20	• Product Integration	70.13 days	Mon 3/27/17	Wed 4/19/17			
21	Connect GUI and bluetooth/audio	20 days	Mon 3/27/17	Wed 4/19/17	14	Manoj_Olga_Peter_Siddhartha	5
22	• Testing	91.13 days	Tue 3/14/17	Thu 4/13/17			
23	Testing GUI	11 days	Tue 3/14/17	Tue 3/28/17	4	Manoj_Olga	3
24	Testing bluetooth/audio	25 days	Tue 3/14/17	Thu 4/13/17	5	SiddharthaPeter	3
25	• Optimization	5 days	Thu 4/13/17	Sat 4/15/17			
26	GUI enhancement	5 days	Thu 4/13/17	Sat 4/15/17	24		1
27	Multiple user enhancement	5 days	Thu 4/13/17	Sat 4/15/17	24		1
28	Communication speed	5 days	Thu 4/13/17	Sat 4/15/17	24		1

Figure 2. List of Tasks and Assignments

GUI

- Mock-up of the app (Task 9):
Design a mock-up of the GUI to reflect actual design flow and navigation.
- Flow (Task 10):
Handle navigation between the UI screens depending on user actions or network errors. For example, pressing the disconnect button will cease communication and returns to the opening screen. If the connection is dropped when in the communication window, the app will redirect to the opening screen.
- User selection (Task 11):
A UI screen that lists paired devices and allows selection of one user at a time.
- Communication Window (Task 12):
A single UI screen with a PTT button and a button to disconnect. Listen and Connect buttons are greyed out on this screen.

Bluetooth/Audio

- Echo audio (Task 14):

This task involves writing code that records audio as an input and then outputs audio on the same device.

- Establish a connection (Task 15):

Pairing of the devices in Bluetooth is one of the assumptions in the project. This is a requirement before a connection can be established. The code for this part will involve assigning one of the devices as a discoverable host and the other acts as a client, searching for a device.

- Send text between devices (Task 16):

Once the connection is established, a test will be required to verify if the connection is valid. This will be done by using simple text read/write between the devices.

- One-way audio communication (Task 17):

With a connection established, have audio data pushed from the host to client. The client device will output the audio data received from the host.

- Two-way audio communication (Task 18):

With a connection established, have audio data sent/received between the host and client.

- Two-way audio communication with microphone blocking (Task 19):

In addition to having the two-way communication functioning, this step will involve disabling microphone input when receiving audio data. This can be summarized as forcing one device to only receive when the other device is sending.

Product Integration

- Connect GUI and Bluetooth/Audio

Android applications use a model like the traditional MVC (Model, View, Controller). The GUI phase primarily focuses on what is the equivalent of a 'View' in MVC – called User Interface, while the Bluetooth/Audio phase focuses on the equivalent of a 'Model' in MVC – called Resources. The 'Product Integration' phase handles connecting the UI to the Resources through connectors. This happens concurrently with the Bluetooth/Audio tasks (Tasks 14 to 19). As each task gets completed, the GUI integration begins for that task.

Some of the sub-tasks for this phase include:

- Handling method calls for connection establishment/termination and transferring data with the GUI
- Handling errors such as dropped connections, dropped packets, etc.
- Establishing Action Handlers and Navigation Controllers to maintain flow

6. Practical Problems

During the course of this project there were a few problems that we ran into. In our original design, we had planned on using a chat system where users could join channels and talk to other users in that channel freely. This ended up become very complex and so we changed the scope of the design to be a direct device-to-device communication between two users. Another issue that came up was that our UI was becoming very cluttered as we kept adding functionality. This was solved by manipulating the visibility of controls based on the current task being performed. Towards the end of the project we had a stability issue where task threads were getting blocked indefinitely on read calls. We were able to resolve the issue by checking for available data as a socket instream method before issuing the read data command. We had originally wanted to disable the talk button on the receiving device when audio data was being sent, but that proved difficult when working with one socket for communication. Currently if both talk buttons are pushed simultaneously, the received audio will play when the talk button is released (delayed but stable).

Based on our experience with this project, there are a few suggestions for future development that can improve functionality. Using more than one communication socket between each pair of devices can greatly help with flow control and device interactions. Use threads to run blocking methods (read/write) so that your UI does not freeze during execution. Use fragments to encapsulate UI functions. This allows for more screen real-estate and the ability to add similar functionality to future projects (modular code).

7. Future Improvements

In this project we have implemented a simple Bluetooth Android Walkie Talkie. Future improvements include:

- The GUI was kept simple, so there is potential for better graphical design
- This application could be extended for multiple users. At the moment we only have pairing between two users. It is possible to extend this to more users and have one user broadcast when others listen.
- List only paired devices that are in range when Connect button is pressed. At the moment all of the paired devices are shown.

8. References

1. "Android AudioRecord example"
<http://stackoverflow.com/questions/8499042/android-audiorecord-example>
2. "Android Developers AudioRecord"
<https://developer.android.com/reference/android/media/AudioRecord.html>
3. "Android Developers AudioTrack"
<https://developer.android.com/reference/android/media/AudioTrack.html>
4. "Android Developers Bluetooth"
<https://developer.android.com/guide/topics/connectivity/bluetooth.html#ManagingAConnection>
5. "Create a Bluetooth Scanner With Android's Bluetooth API"
<https://code.tutsplus.com/tutorials/create-a-bluetooth-scanner-with-androids-bluetooth-api--cms-24084>
6. "Live Audio Recording and Playing in Android and Thread & callback handling"
<http://stackoverflow.com/questions/9413998/live-audio-recording-and-playing-in-android-and-thread-callback-handling>