CS M117

Computer Networks: Physical Layer

Project Report

MyPoll

Spring 2016

Shijun Huang	904302829
Zhengxu Xia	104250792
Zhanxi Gao	504282654
Jiahui Li	004356402
Yu Wang	704409906

I. Abstract

1.1 Motivation

We all have the experience struggling to make a decision, especially with a large group of people. So we thought it would be great if there is a handy app in our phones on which we can start a vote right away and get instant response from other users, of whom the names remain unknown. With this motivation, our group endeavored to develop this voting app for our project--MyPoll.

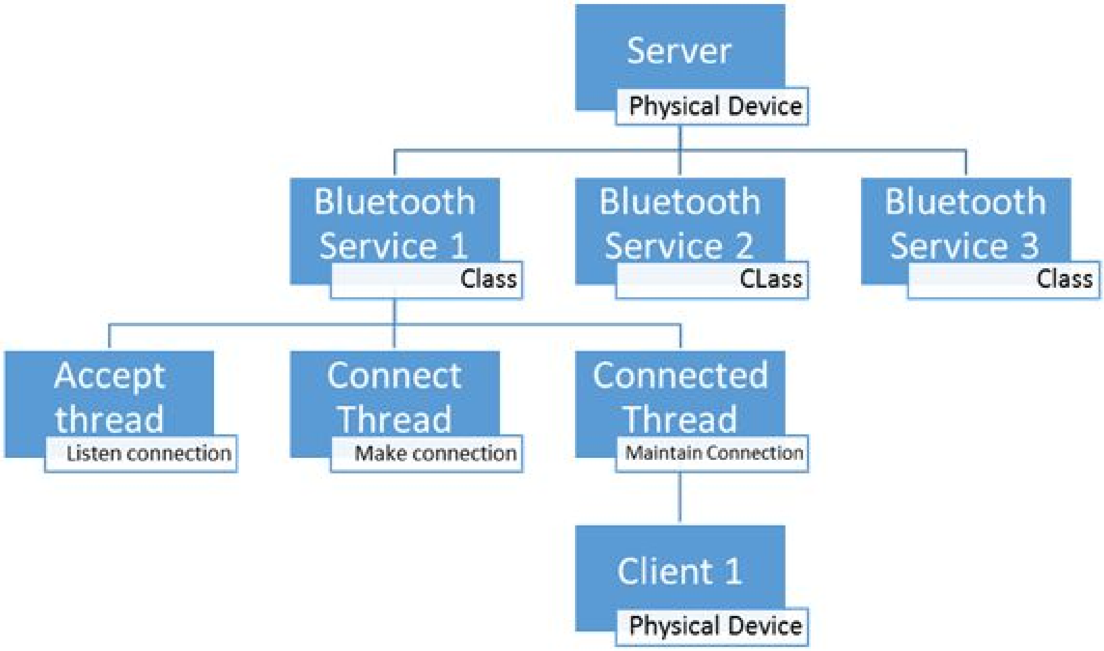
1.2 Features

Main Features	**Description**
Wireless	With smartphones handy on a daily routine, a face-to-face group of users can start a vote at anytime and anywhere they travel. Also, the poll can be completed without requirement of WIFI or Data, since MyPoll uses Bluetooth as the method of communication.
Interaction	Mypoll allows interaction among multiple users. The current version is able to support up to seven users voting at one time.

Instantaneity	One user types in and broadcasts the question and others can receive and vote for it right away. The result is displayed on the screen immediately. Thus, MyPoll is a time-saving application.
Anonymousness	On some occasions, one may not want others to know what he votes for. MyPoll can be used for any situations when a secret ballot is needed.

II. Architecture Details

The multiple device bluetooth connection network is achieved by socket programming with Android Bluetooth Module. The detailed structure is illustrated in the figure below. Each Bluetooth service is able to hold one connection, and communication is achieved through this connection as shown below.



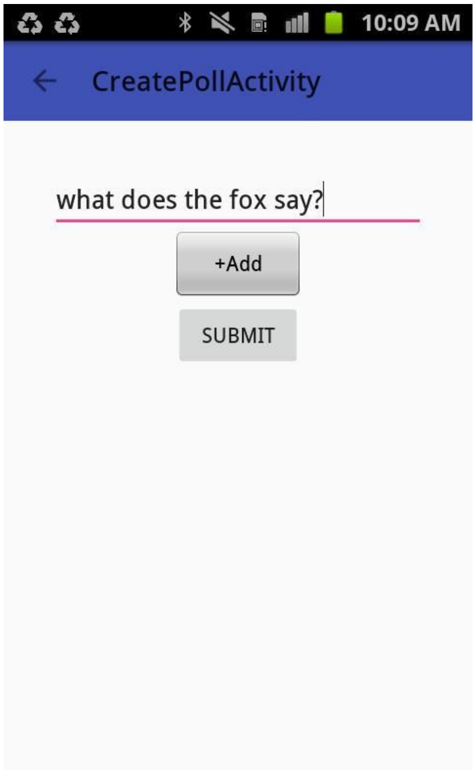
III. Results and Demo

As stated in Section 1.1, our motivation is to create an Android application that allows us to vote and get results instantly. We have successfully reached our goals in MyPoll.

A user can create a survey by editing a question and adding multiple options. This user acts as a server and is able to broadcast the survey via Bluetooth. Others users, acting as clients, can enable their Bluetooth module to search the server device to make a connection. The server device contains up to seven unique UUIDs (each for a different client), it thus supports multiple Bluetooth connections.

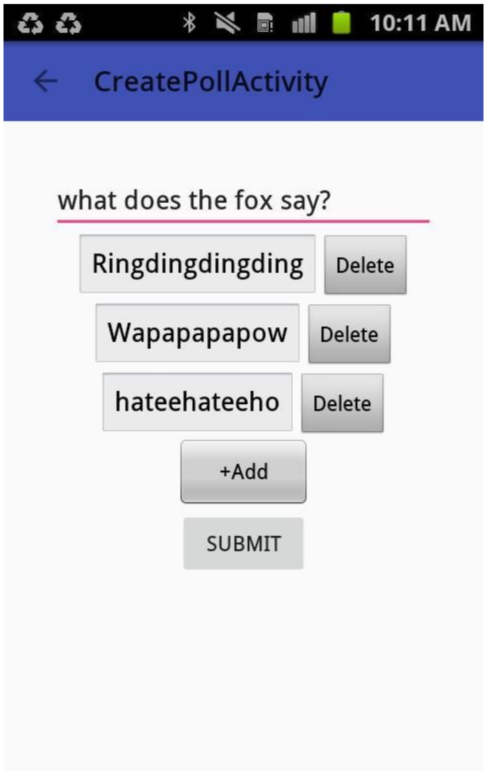
Each client device will receive the survey and make a choice so that it can send back the result to the server. After the server device collects all the results, it can make a final summary and send the result back to each client.

The following figures show the detailed interface of MyPoll.



what does the fox say?

Figure 3.1 Main Interface of MyPoll. This
is the interface shown when users open
the a Figure 3.2 Interface of creating a poll. The
users user can type in the question.
respo
created and broadcasted.



what does the fox say?

Ringdingdingding

Wapapapapow

hateehateeho

Figure 3.3: Interface of creating a poll. The user can add as many as options he wants, and can delete them if he does not want them anymore. After completing the question and options, the user can hit the submit button to submit the question.

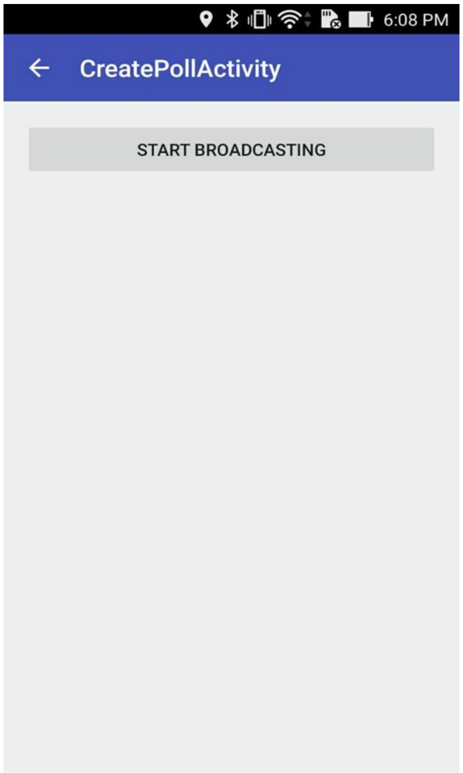
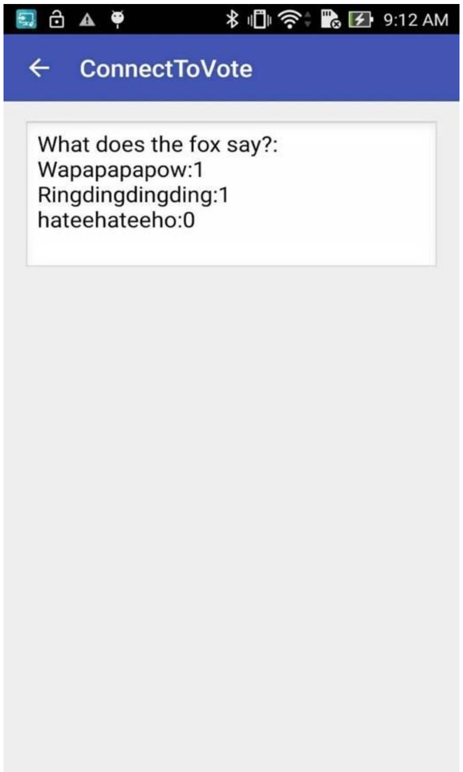


Figure 3.4: interface of creating a poll. After hitting the submit button, the user may start broadcasting his question.



```
What does the fox say?:
Wapapapapow:1
Ringdingdingding:1
hateehateeho:0
```

Figure 3.5: Poll result. This picture is captured by a voter, who received the result after the initiator compute and public the result.

IV. Discussion

During the development, we encountered several difficulties which included not only the learning of a new programming language but also the obstacles in designing. The very first challenge was that we needed to read Java tutorial to learn a new language as well as Android Studio and Android SDK. It was a pretty time-consuming process for most of us due to our limited programming experience as non CS major students.

The second trouble we were struggling with was to design a layer which we called Bluetooth service layer. It was used to support multiple device client-server connections via Bluetooth. Initially, we assumed one Bluetooth service layer can hold multiple connections by using only one accept thread, one connect thread, and several connected threads. However, connection testing always failed for the second client device. After a very long time struggling with this layer design, we finally came up with an idea that we could hold multiple Bluetooth service class objects so that each can support a single server-client connection with its own accept thread, connect thread, and connected thread. The third trouble of this project had to be debugging the code. As always, debugging is a painful process.

V. Conclusion

As we successfully reached our goal and implemented the basic functionality of poll in our app, this is just a beginning, and there are various improvements we can make in the future. First, now our app has a very plain user interface. We may go on to design the user interface to be more versatile and user-friendly. For example, we may sort the results by the number of votes, so that the user can easily see which options are more popular. In addition, we believe that it will be useful if users can share images and other files in the poll, as sometimes it may be hard to fully describe the options in just a line of text.

Moreover, currently, one poll is restricted to seven users because now our app only supports one master device for each poll, as Figure 5.1 has shown. We plan to overcome this limitation by trying to exploit the slave/master mode and have multiple piconets during the communication and poll like the model shown in Figure 5.2.

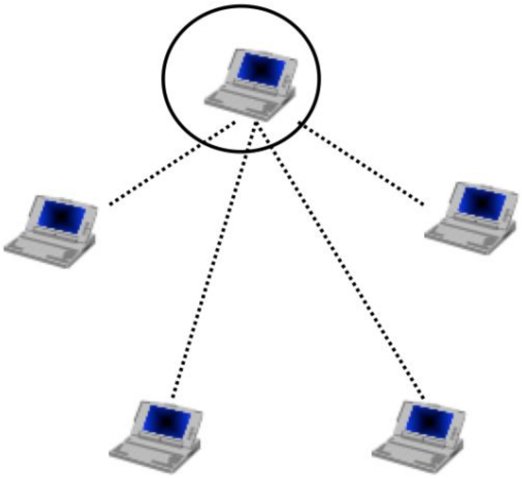


Figure 5.1: A group of users in one piconet with only one master device.

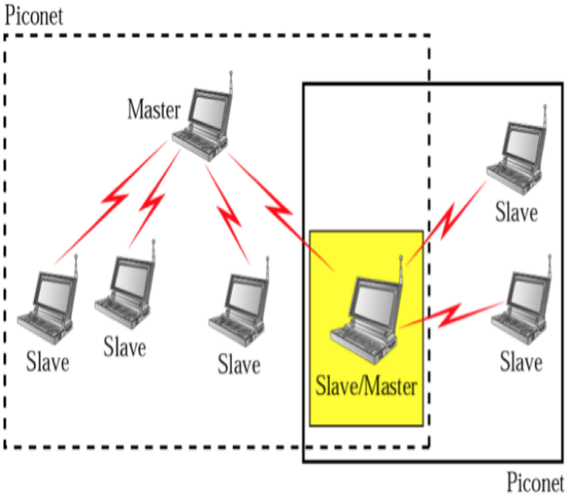


Figure 5.2: A group of users connected in scatternet composed of several piconets, with one of the slaves in the first piconet to be the master of another piconet.