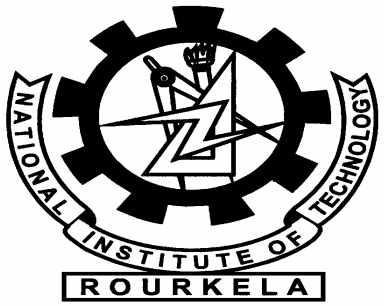
# A Bluetooth Messenger Application

**Arnab Banerjee**
**111CS0118**




**Department of Computer Science**
**National Institute of Technology, Rourkela**
**May 11, 2015**

# A Bluetooth Messenger Application

*Thesis submitted in partial fulfillment*
*of the requirements for the degree of*

**Bachelor of Technology**
*in*
**Computer Science and Engineering**

*by*

**Arnab Banerjee**
**111CS0118**

*Under the supervision of*

**Prof. Suchismita Chinara**


**Department of Computer Science**
**National Institute of Technology, Rourkela**
**May 11, 2015**


Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 11, 2015

# Certificate

This is to certify that the work in the thesis entitled *A Bluetooth Messenger Application* by Arnab Banerjee is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Suchismita Chinara,
Assistant Professor,
Dept of Computer Science and Engineering,
NIT Rourkela

# Acknowledgement

The successful completion of the thesis involves help and contribution from several people.

First and foremost, I would like to express my gratitude towards my project guide, Prof. Suchismita Chinara, whose guidance gave me the proper direction along which I gleefully worked. I also would like to thank her for allowing me to incorporate my ideas while doing the project. I would also like to thank my project review panel for their time and their sincere advices of improvement.

I would also like to thank my fellow researches who also gave their precious time for helping me out whenever required. It would have been very difficult without their support.

I would also like to express love for my parents for always supporting and guiding me through all my endeavours.

# Abstract

One of the most important application of Bluetooth technology is to exchange information between or amongst devices. The information can be a text file, media file or any normal text. This project aims to design Android application for server-client interaction using Bluetooth. If two Blutooth-enabled android phones are in close proximity to each other, this application will help them in exchanging information. This scenario may arise when two or more persons sitting in a room wants to exchange messages without speaking.

The project contains two different applications, namely server and client. Since in Bluetooth programming both have their own distinctive task they need to be designed separately. The responsibility of the server is to wait for the client's request and on the other hand the client has the responsibility of discovering the server and sending a connection request. Once connected, both of them can send or receive messages till the connection is not closed. The entire project was designed using Java programming language.

Prior to attempting to design Android app, a Windows console application was designed for the same and the successful results motivated the designing in Android.

# Table of Contents

# Chapter 1
# Introduction

## 1.1 Bluetooth technology:

Bluetooth is a technology designed by Ericsson in 1994 to wirelessly connect devices. It is used to exchange data among devices within short range, usually 10 metres. Bluetooth operates in the frequency range 2400 MHz to 2483.5 MHz which is the ISM unlicensed band. Bluetooth uses the concept of frequency hopping to avoid collisions. It usually performs 1500 hops per second[3].

Bluetooth exchange data using the concept of packets. The architecture used by Bluetooth is master-slave architecture. One master can communicate with maximum 7 slaves in a piconet structure. In order to connect more than 7 slaves a scatternet needs to be formed. When more than one slave is connected then bluetooth operates in round-robin fashion. Typically, the master switches from one device to another for a fixed period of time[3].

There are several applications of bluetooth out of which few are mentioned below:

a) Connecting two nearby computers for data exchange.

b) Wireless communication with tablets and speakers such as iOS and Android devices.

c) Sending small advertisements from Bluetooth-enabled advertising hoardings to other, discoverable, Bluetooth devices.

d)Transfer of files, contacts and other information.

e) Wireless control of and communication between a mobile phone and a handsfree headset.

Some of the limitations of Bluetooth technology are mentioned below:

a) As compared to Wi-Fi, which transfer data at 54 Mbps, Bluetooth is comparatively slower and its transfer rate is about 3 Mbps.

b) Distance Limitations: For a bluetooth hardware device installed in our computers or mobile devices, the range is upto 30 feet which sometimes be too short.

c) Interference: Since bluetooth operates in the unlicensed frequency spectrum, interference among Bluetooth signals is possible.

### 1.1.1 **Frequency Hopping**

Frequency Hopping is a technique which is used by Bluetooth so as to avoid interference amongst the signals. Bluetooth operates in the open unlicensed spectrum and thus is not protected from interference from other signals like Wi-Fi and other radio signals. The bluetooth frequency spectrum varies from 2400 MHz to 2483.5 MHz. This frequency range has been divided into 79 different channels and the packets being transmitted hops among these channels (changes channels) so that interference is less in case that frequency is being used by some other technology signal. Bluetooth hops about 1500 times per second thus lending efficiency in transmission[5].

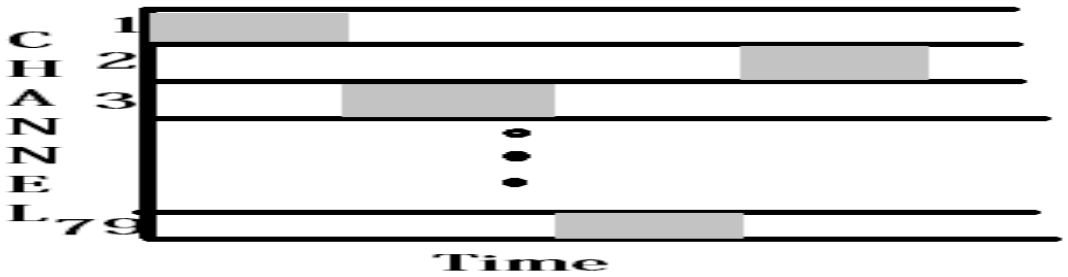



**Fig. 1: Frequency Hopping**

Fig.1 shows how the packet data hops across different frequencies at different intervals of time.

## 1.2 Android Operating System:

Android is a mobile operating system developed by Google. It is based on Linux kernel. Android is primarily designed for mobile devices such as smartphones, tablets, Android televisions, wrist watches. All the above mentioned devices are touchscreen devices. Android uses touch inputs like swiping, tapping, pinching and reverse pinching on objects visible on screen along with a virtual keyboard. It could be also used in game consoles and other electronics[7].

The list of features in Android Operating system are mentioned below:

a) Messaging: SMS and MMS are the means of exchanging messages. Android Google Cloud Messaging(GCM) is also a part of Android Push Messaging service.

b) Web Browser: It is based on caopen-source Blink layout engine coupled with Chromes V8 JavaScript engine. The browser is one of the most effective browsers and scored 100/100 in Acid3 Test on Android 4.0.

c) Screen Capture: Android supports capturing a screenshot by pressing the power and volume-down buttons at the same time.

d) Connectivity: Android supports connectivity technologies including GSM/EDGE, Wi-Fi, Bluetooth, LTE, CDMA, EV-DO, UMTS, NFC, IDEN and WiMAX[7].

## 1.3 Bluetooth Connectivity in Android:

The Android platform supports Bluetooth data transfer. Exchange of data is done using the inbuilt Bluetooth network stack. This stack allows two or more Bluetooth enabled devices to wirelessly exchange messages with each other. Bluetooth APIs are present for the users in order to access the Bluetooth network stack. The APIs help establish one to one connection or one to many connection wirelessly without having to know about the hardware details[6].

Using the Bluetooth API's an android application can do the following tasks:

a) Scan for other Bluetooth devices in the vicinity.

b)Query the local Bluetooth adapter for paired Bluetooth devices.

c)Establish RFCOMM channels with other Bluetooth enabled devices.

d)Connect to other Bluetooth devices through service discovery protocol.

e)Transfer data to and from other devices.

f)Manage multiple connections[6].

**Android apps** can be designed which performs the above mentioned tasks. As mentioned in the step (e) which states that we can transfer data to and from other devices, this fact can be exploited to send questions and receive answers in response. Android apps, designed to chat with fellow devices can also be designed.

# Chapter 2

# Literature Review

## 2.1 General Bluetooth Programming Steps:

### 2.1.1 *Outgoing connection(Client)*

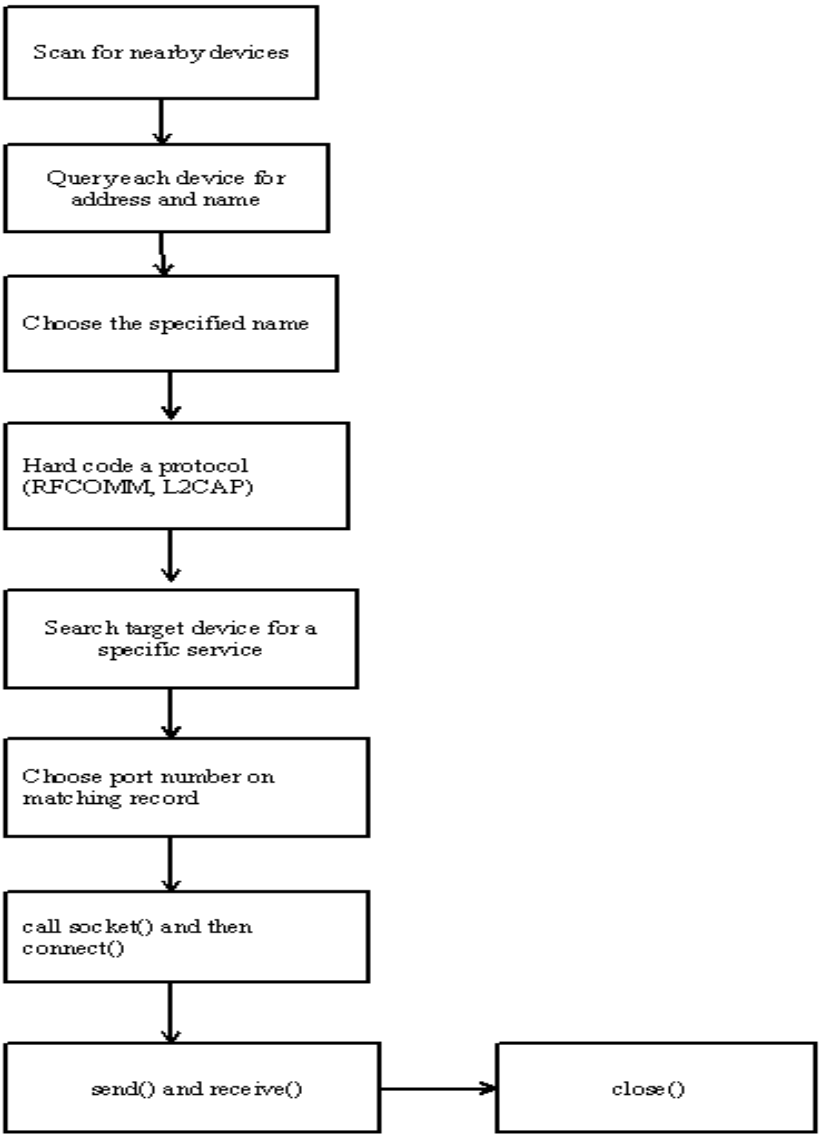



Fig.2:Client Flowchart

Fig.2 shows the steps performed by the client in establishing a connection with the server.The first and foremost step is to search for nearby devices, which is also known device inquiry. Every bluetooth hardware comes with a globally unique 48-bit address referred to as the *Bluetooth address*. When a Bluetooth enabled devices inquires about the nearby devies its actually they enquire about this 48-bit address. For two devices to communicate with each other both of them should know each other's Bluetooth address[1].

The next step is to query each device for its device name. Every bluetooth device can be given a *Bluetooth name* which will work as alias for the Bluetooth address. Whenever a device searches for its nearby devices, it actually asks their nearby devices for the Bluetooth address along with its name[1].

The next step is to select device with whom one wants to connect[1].

The next step is to select a common protocol so as to communicate between them. The most commonly used Bluetooth protocols are RFCOMM and L2CAP. Other than these two there are other protocols such as OBEX, ACL or SCO.

The next step for the device performing an outgoing connection is to select the service from the list of services as present in the SDP records of the device at the other end. The port number in which the server is listening is generally hard-coded or predefined. Bluetooth devices uses Service Discovery Protocol(SDP) in which every Bluetooth device maintains an SDP server listening on a well-known port number. When a server application starts up, it registers a description of itself and a port number with the SDP server on the local device. Then, when a remote client application first connects to the device, it provides a description of the service its searching for to the SDP server, and the SDP server provides a listing of all services that match[1].

The next step is to establish a socket and to connect to a listening socket on the device which is accepting the connection. After the connection has been established messages can be sent or received depending upon our requirements. In the last step whenever exchange of data has ended the client needs to close the socket connection[1].

2.1.2*Incoming connection(Server):*

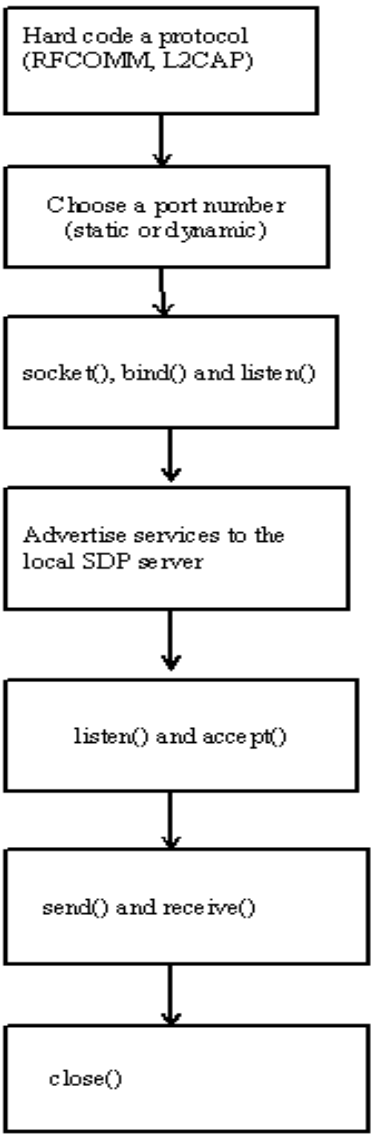



Fig.3: Server Flowchart

Fig.3 shows the steps performed by the server in establishing a connection with the client. The first and foremost step is to hard code the transport protocol which will be used for establishment of the connection. The server also needs to choose a port number on which it will listen or it may be dynamically assigned one. After this it will create a socket, bind the socket to

the port and listen to the port till an incoming connection request arrives. Along with the previous step, the server needs to advertise its services using its SDP server so that the client could recognize what are the types of services that the server could provide. Whenever an incoming connection arrives, the server could either accept or reject the connection. Normally, the server moves forward with the connection and establishes a connection with the client. After connection establishment exchange of data by sending and receive messages takes place and when the data exchange is over simply the socket connection is closed[1].

## 2.2 Bluetooth Programming in Windows Environment:

JSR-2 package is provided in Java which contains API's for Bluetooth programming[1].

### 2.2.1 *Sockets and Connections:*

Sockets are referred to as connections in Java. A RFCOMM socket is represented by StreamConnection interface while a L2CAP socket is represented by L2CAP connection interface, both of which implement the Connection interface. Connected sockets are different to listening sockets. A listening socket is referred to as StreamConnectionNotifier[1].

### 2.2.2 *Service Discovery Protocol:*

In JSR-82 a listening socket is created along with the advertisement of an SDP service. Atleast one Service class ID must be specified while creating a listening socket. The service record can be modified after creation of listening socket but cannot be unadvertised before the closing of socket[1].

### 2.2.3 *Dynamically assigned port numbers:*

In JSR-82, port numbers are always dynamically assigned and it is not possible to use programmer-assigned port numbers. It may come as an inconvenience for those who want to put a quick Bluetooth service[1].

### 2.2.4 *Tighter integration of Bluetooth security:*

Bluetooth security is tightly integrated into JSR-82, and options for Bluetooth authentication and encryption are often built into the various connection methods[1].

## 2.3 Bluetooth programming in Android Environment:

### 2.3.1 *Classes&Interfaces required:*

a)BluetoothAdapter: Represents the local Bluetooth adapter (Bluetooth radio)[6].

b)BluetoothDevice: Represents a remote Bluetooth device. Use this to request a connection with a remote device through a BluetoothSocket or query information about the device such as its name, address, class, and bonding state[6].

c) BluetoothSocket: Represents the interface for a Bluetooth socket (similar to a TCP Socket)[6].

d) BluetoothServerSocket: Represents an open server socket that listens for incoming requests (similar to a TCP ServerSocket)[6].

e) BluetoothClass: The general characteristics and capabilities of a Bluetooth device are described in this class[6].

There are other classes and interfaces as well but is not required for the current work.

### 2.3.2 *Permission required:*

Bluetooth permission *BLUETOOTH* is needed to use the Bluetooth features in your application. This permission is required to perform any Bluetooth communication such as requesting a connection, acceptance of connection and data transfer[6].

*BLUETOOTH_ADMIN* permission is required in order to initiate device discovery or manipulate Bluetooth settings. Most application needs it to discover local Bluetooth devices[6].

# Chapter 3
# Proposed Work

The work has been divided into two phases. First phase involves designing a Windows application and in next phase designing an Android application.

3.1 *Messenger Application in Microsoft Windows Environment:*

Using the javax.bluetooth package a client and a server program was written in Java programming Language. The server and client programs were designed in order to exchange messages,in this case perform chatting, with each other. The server is also capable of interacting with several clients at the same time, using the concept of piconet. The flowchart of both the server and the client are as shown below:

*3.1.1 Server Flowchart:*

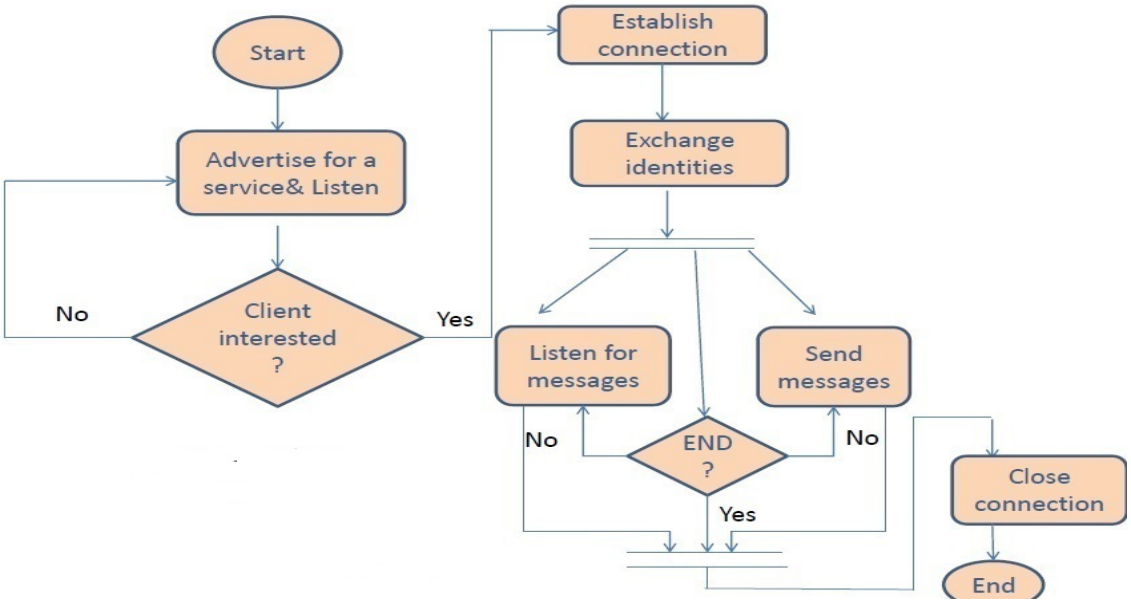



Fig.4: Server Flowchart

Fig.4 displays the flowchart followed by the client in order to establish the connection. The steps performed by the server are already explained in the previous section.

*3.1.2Client Flowchart:*

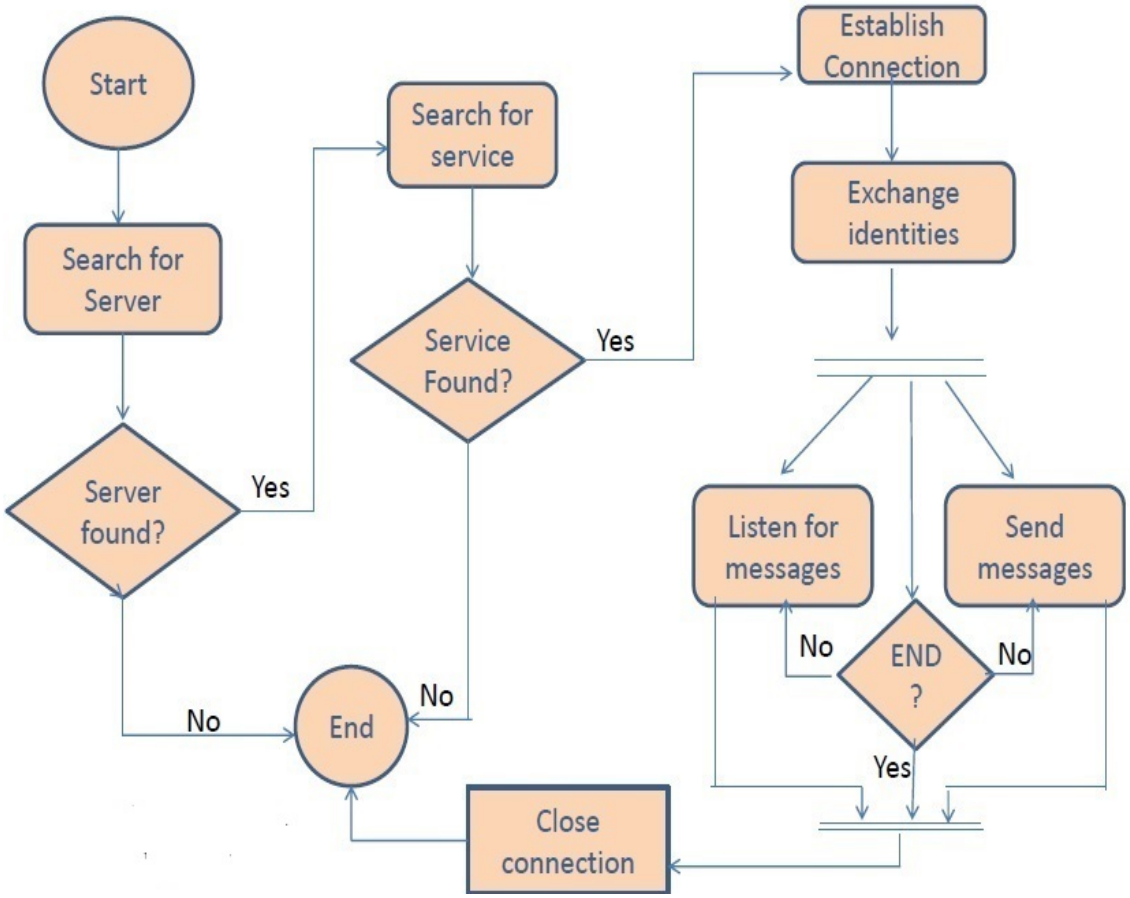



Fig.5: Client Flowchart

Fig.5 displays the flowchart followed by the client in order to establish the connection. The steps performed by the client are already explained in the previous section.

*3.1.3 Working Model of Single Server- Multiple Client Application*

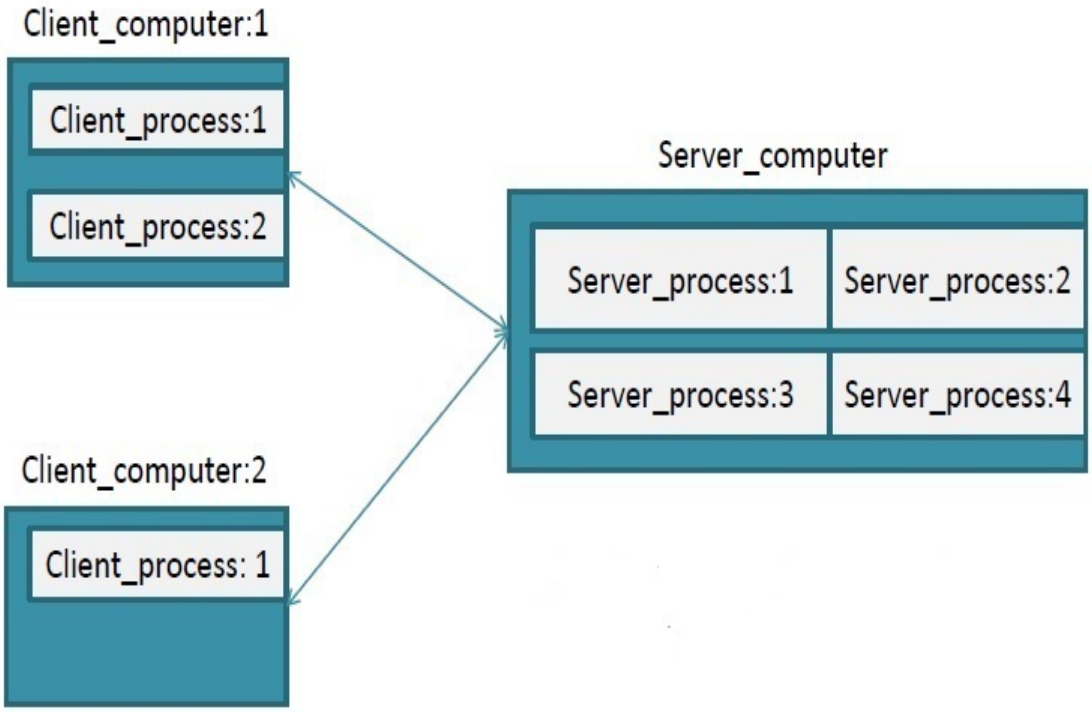



Fig.6: Many to one communication model

The above figure shows the many to one connection model. The figure shows a server computer running four server processes while two different client computers have established connection with server. The first client runs two instances of the client process while the other client computer has only one client process running.

*3.1.4 Output Snapshots*

a) Client 1:

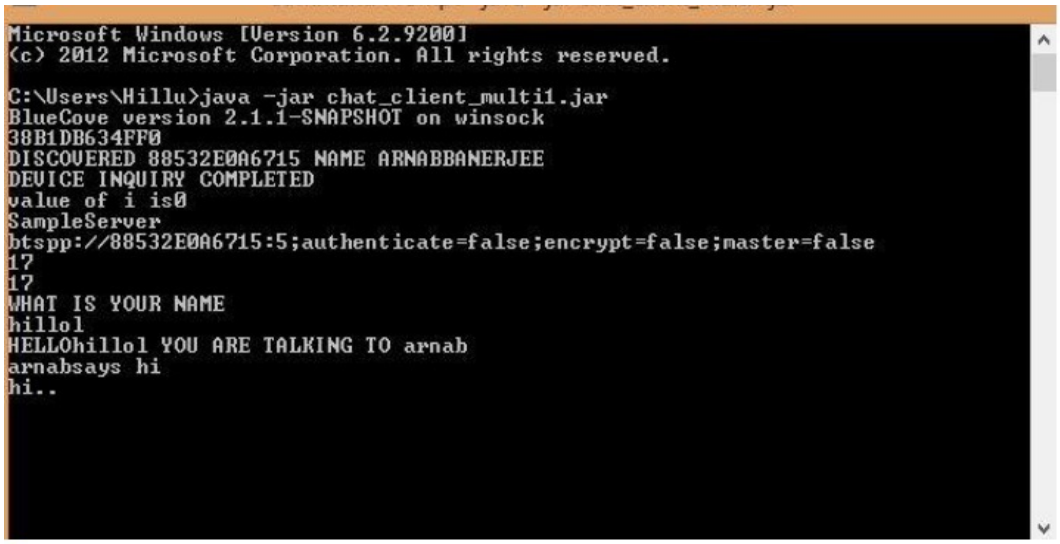



Fig.7: Console output for Client:1

Fig.7 shows the output console of the first client application. The console shown in the figure is used to exchange messages. The output shows the steps performed to run the application, bluetooth discovered devices, service discovery, identity exchange and lastly message exchange.

b) Client 2:

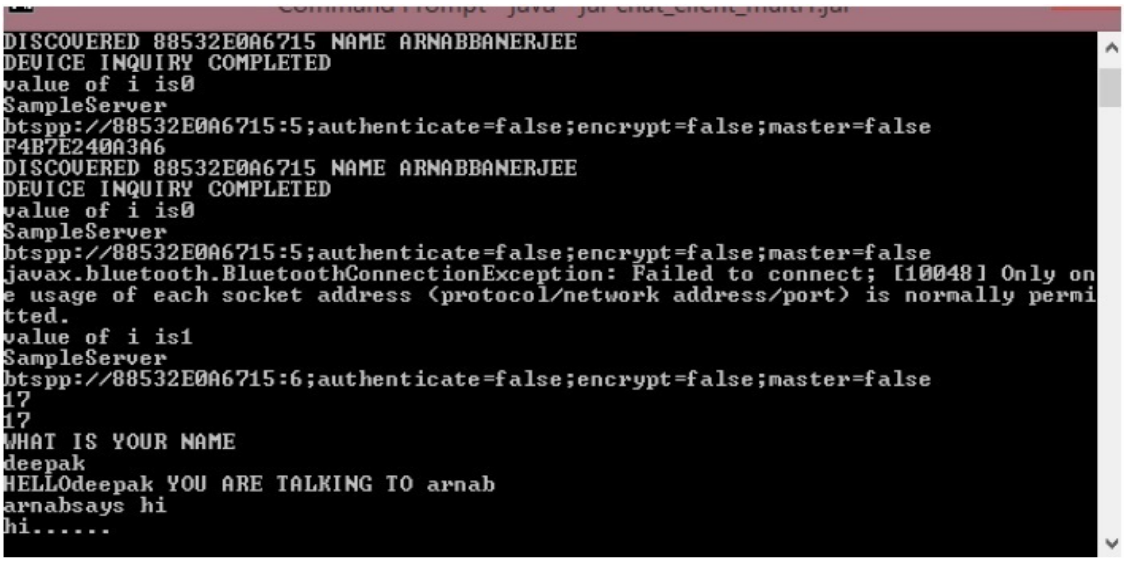



Fig.8: Console output for Client:2

Fig.8 shows the output console of the second client application. The same steps are shown in the console of the second application.

c) Server:

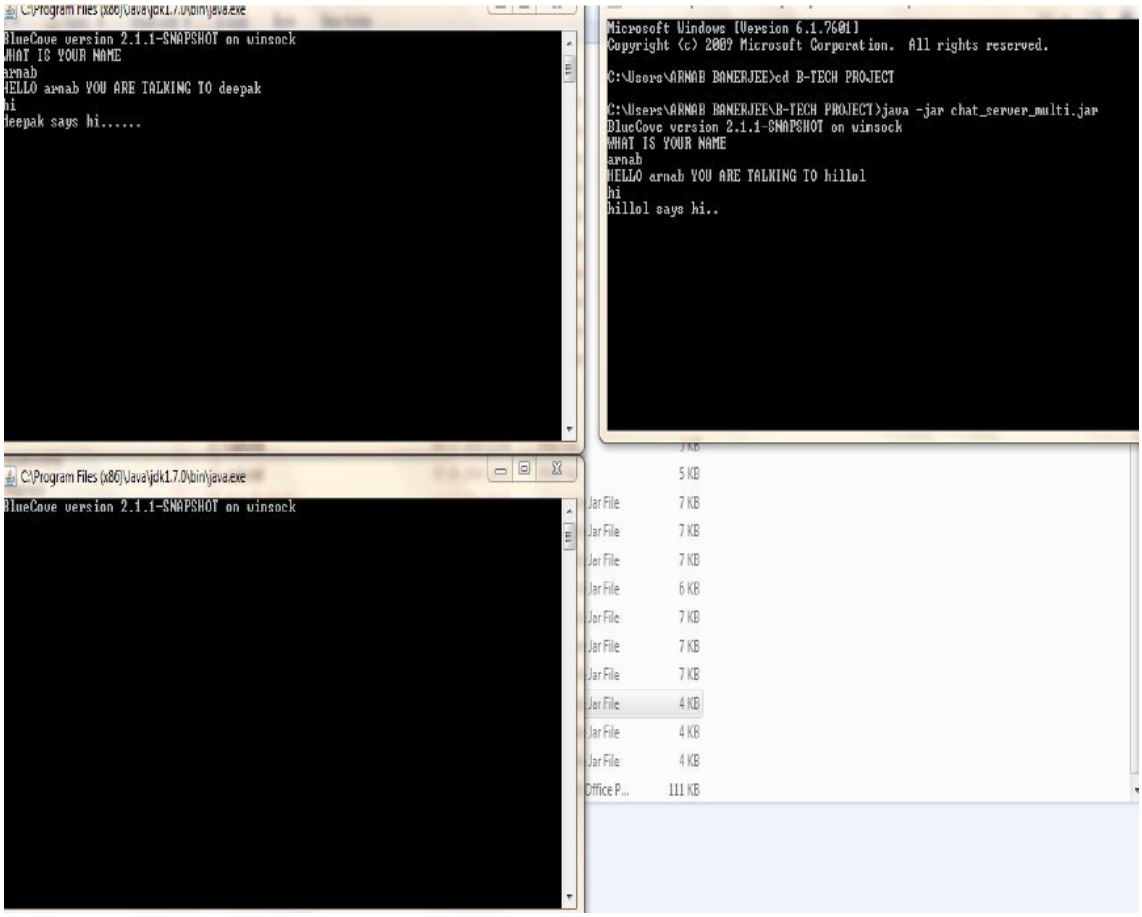



Fig.9: Console output for Server

Fig.9 shows the output console of the server applications running on the server computer. Since there are two client applications running, there are exactly three serverapplications. The first two are communicating with the existing client applications and the third server application is listening for the next client's inititation.

### 3.2 *Messenger Android Application in Android Environment*

Since Android mobile phones are more easily available, the above application can be replicated in Android environment. Using the android.bluetooth package API's a messager app can be built. The steps remains the same but the functions required to perform scanning, establishing connection, sending and receiving are different than the functions in Windows Environment.

**Snapshots:**

a) server:

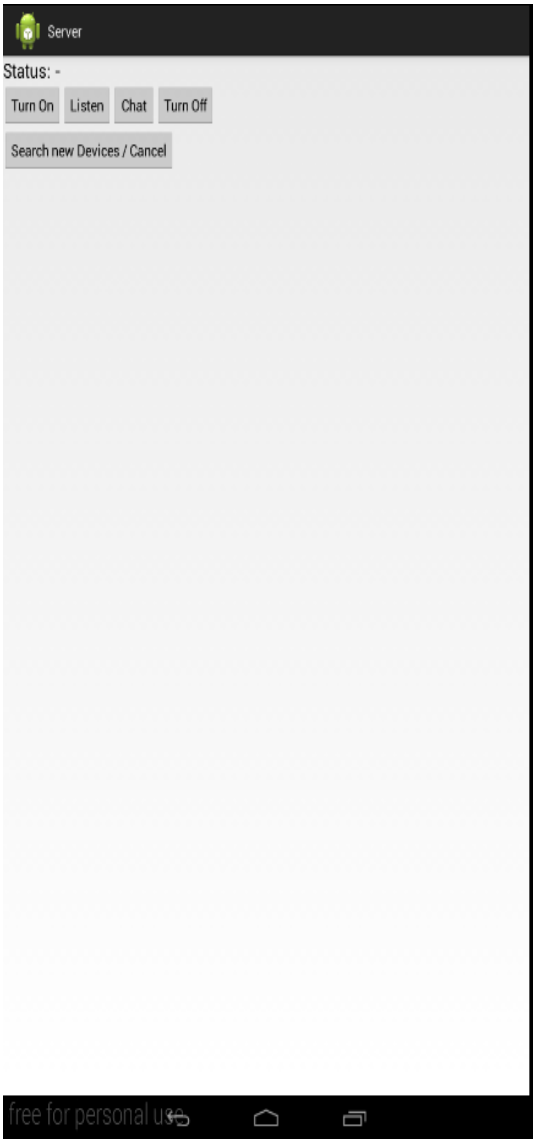



Fig.10: Server application

Fig.10 shows the snapshot of the Server application designed in Android platform. The buttons and their uses are specified below:

a) Turn On: Turn on the bluetooth in mobile.
b) Listen: Server starts listening.

c) Chat: Start exchaning messages.
d) Turn Off: Turn off bluetooth.
e) Search for Devices/Cancel: This button is used to start/stop searching for nearby Bluetooth enabled devices.

b) client :

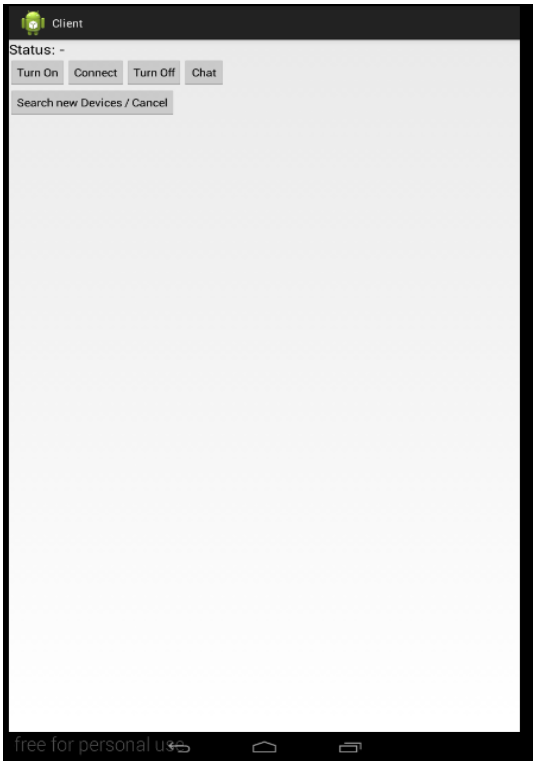



Fig.11: Client Application

Fig.11 shows the snapshot of the Client application designed in Android platform. The buttons and their uses are specified below:

a) Turn On: Turn on the bluetooth in mobile.
b) Connect: Client sends request for connection.
c) Chat: Start exchaning messages.

d) Turn Off: Turn off bluetooth.
e) Search for Devices/Cancel: This button is used to start/stop searching for nearby Bluetooth enabled devices.

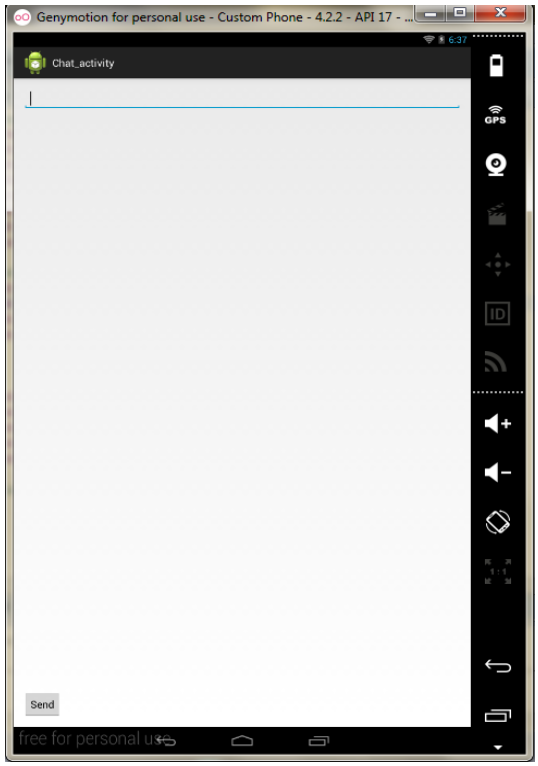



Fig.12: Snapshot of an Android Activity

Fig.12 shows the Android activity through which message can by typed and sent to the other device. The figure shows a textbox in which we can insert text for sending and send button is used to send the message.

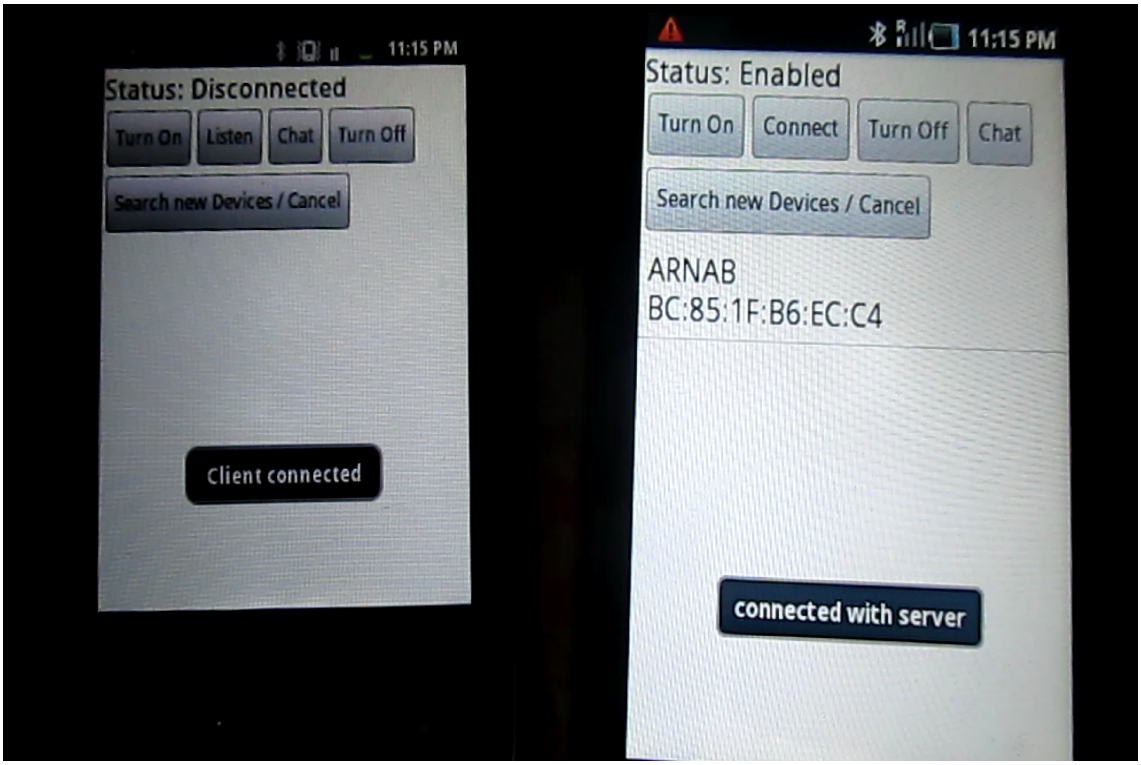



Fig.13: Snapshot of connection establishment

Fig.13 shows a snapshot of the working of the client and the server application. The left side mobile, which is the server informs that the client is connected while on the right side which is the client informs that server is connected. After connection message exchange takes place.

# Chapter 4
# Conclusion&Future Scope

The messenger application using Bluetooth technology was successfully completed in Windows platform and in Android platform.

a) Bluetooth consumes low power as compared to Wi-fi technology and on the other hand Wi-fi has a range of almost ten times to that of Bluetooth. In order to account for the limitations in range, we can incorporate Wi-fi communication in our application. Whenever our client application finds that server is not within range, the client may start scanning for the server using Wi-fi. This will ensure the range limitations does not arise.

b) In a piconet atmost 7 clients can connect with the server. In order to connect with more number of devices a client itself need to act as a server. Then we can form a scattenet(combination of piconets).

# Bibliography

[1] Thompson T.J., Kline P.J. & Balakumar C(2008),*Bluetooth Application Programming with Java APIs*,(4th ed.)

[2]Gavalas, D., & Economou, D. (2011). Development platforms for mobile applications: Status and trends. Software, IEEE, 28(1), 77-86.

[3] Huang A.S. & Rudolph L(2007), *Bluetooth Essentials for Programmers*(1st ed)

[4] Hopkins B. & Antony R.(2003), Bluetooth for Java

[5] Peterson, B. S., Baldwin, R. O., & Kharoufeh, J. P. (2006). Bluetooth inquiry time characterization and selection.textitMobile Computing, IEEE Transactions on, 5(9),1173-1187.

[6] Bluetooth. (n.d.). Retrieved March 8, 2015, from http://developer.android.com/guide/topics/

[7]Alliance, O. H. (2013). Android (operating system). Marketing, 4, 5.