

# C0324- PROJECT - 01

VOICE CONFERENCE

GROUP 09

AMARASINGHE W.G.N.M (E/11/019)

BANDARA K.M.P (E/11/040)

# C0324- PROJECT - 01

---

## VOICE CONFERENCE

### ABSTRACT

We have to create java application to design and code a basic peer to peer voice conferencing application similar to Skype in two iterations.

1. In the first iteration, implemented voice communication between two parties
2. In second iteration ,extended first iteration to support multi-party call conferencing

### TABLE CONTENT

- 1). Introduction
- 2). System design and Architecture
- 3). Implementation
- 4). Conclusion and future works

### INTRODUCTION

With the development of the whole world, more and more people need to communicate with each other even if they live in different parts of the world, especially for the businessmen who tried to make profits all over the world. In that situation, it is clear that an audio conference system will be popular because it permits people live in different places to discuss something with each other easily.

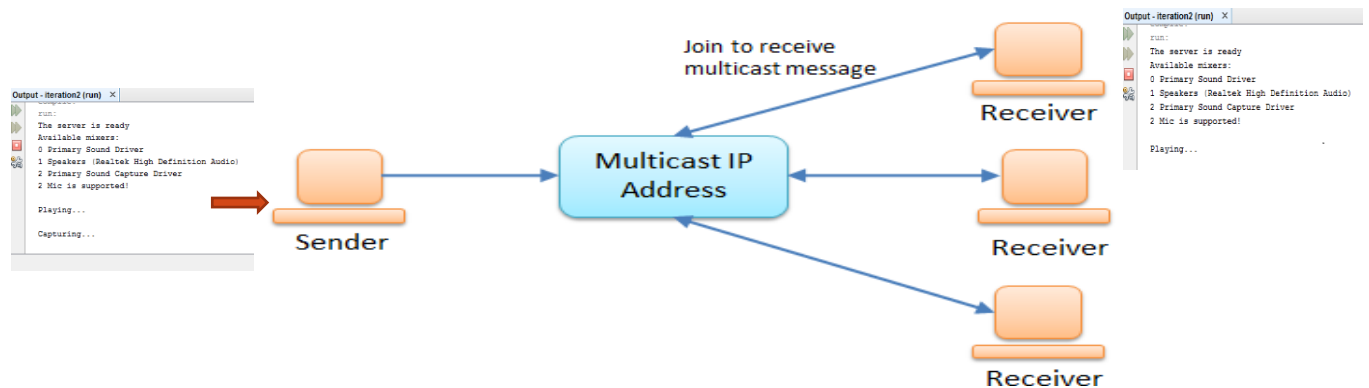
The audio conference system can not only provide service for people to talk about something with each other, it can also automatically recognize the user who is speaking and send his/her name to all the other clients. With this special function,

There are two parts in this audio conference system. The first part is a voice The second part is a client-server program. The client program can send the content a user says to the server and receive the recognition result from the server; the server program is responsible for forwarding the contents from one client to others and calls voice recognition system to recognize the user who is speaking.

We have to make a simple application to transfer voice message to a single person or group who are connected through a Wi-Fi network. This is a real time application. When a server records the message, it will be broadcast to the connected Wi-Fi clients.

## SYSTEM AND ARCHITECTURE

- 1). when server records the message it also store in data packets.
- 2). Then the packets are send to a specified client or group of clients who are connected through a Wi-Fi hotspot on time.
- 3). After the start program ,the person who want to send the audio message to other person ,must press the ENTER button (capturing...),then he/she can transmit the message to others ,at that time others press enter to ENTER button (playing...) ,then thay can hear what he/she is saying.



## IMPLIMENTAION

When we run capture .java. The program will be start.

After record voice and transmit it using a socket.

```
private void captureAndSend() {  
    this.stopCapture = true;  
    try {  
        int readCount;  
        while (true) {
```

```

        if (!this.stopCapture) {

            readCount = getTargetDataLine().read(this.tempBuffer, 0, this.tempBuffer.length);
            //capture sound into tempBuffer

            if (readCount > 0) {

                // Construct the datagram packet

                DatagramPacket packet = new DatagramPacket(this.tempBuffer,
this.tempBuffer.length, this.host, 55001);

                // Send the packet

                this.socket.send(packet);

            }

        }

    } catch (IOException e) {

        e.printStackTrace();

    }

}

```

There has play.java class when it run playback the sounds.

```

public void run() {

    try {

        // Construct the socket

        this.socket = new MulticastSocket(this.port);

        this.socket.joinGroup(this.host);

```

```

        System.out.println("The server is ready");

        // Create a packet
        DatagramPacket packet = new DatagramPacket(new byte[this.packetSize], (this.packetSize));
        this.playAudio();

        while (true) {
            if (!this.stopPlay) {
                try {

                    // Receive a packet (blocking)
                    this.socket.receive(packet);

                    // Print the packet
                    this.getSourceDataLine().write(packet.getData(), 0, this.packetSize); //playing the
audio

                    packet.setLength(this.packetSize);

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

We used two threads because we want to different kind of works should be done parallel.

```

public static void main(String[] args) {

    // Check the whether the arguments are given

```

```
if (args.length != 1) {  
    System.out.println("DatagramClient host ");  
    return;  
}  
  
Capture cap = null;  
Play ply = null;  
  
try {  
  
    cap = new Capture(InetAddress.getByName(args[0]));  
    cap.start();  
  
    ply = new Play(InetAddress.getByName(args[0]));  
    ply.start();  
}
```

## CONCLUSION AND FUTURE WORKS

Now we can send voice messages like mobile system, via Wi-Fi network and receive it by multi-client or group of clients in same Wi-Fi network.

We hope to develop the interface to make this app user friendly and market this by including more promotions.