VOICE
CONFERENCEING APPLICATION

CO324 NETWORK AND WEB APPLICATION DESIGN

Group No 7

E/15/123: Wishma Herath
E/15/280: Pubudu Prematilaka
E/15/316: Suneth Samarasinghe

Deaprtment of Computer Engineering
Unviersity of Peradeniya

2019

Contents

List of Figures

1 Introduction

1.1 VOIP

Peer-to-peer (P2P) network is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session. Unlike the client/server model, in which the client makes a service request and the server fulfills the request, the P2P network model allows each node to function as both a client and a server. One of the main features of a P2P system is that each node contributes resources such as bandwidth, storage space, and CPU power. As a result, the system gains more capacity as more nodes become involved. This is opposite to a client-server architecture, where the addition of clients always degrades the overall performance.

1.2 Features

One of the most popular applications in p2p is VOICE communication. Skype and Whatsapp are some of the most common applications in daytoday life. In our applications there are two modes of operations,

1. Peer to peer voice communication between two parties.
2. Peer to peer voice communication among multiple parties.

2 Design

3 Quick Guide

To Run the Unicast Application, The client who initiate the call,

```
javac Peer.java
java Peer peer1
```

The client who joins the call,

```
javac Peer.java
java Peer peer2 [ip address of the initiated client]
```

To run the Multicast Application, Who ever can join to the conference using a multicast ip address

```
javac Multicast.java
java Multicast [Multicast ip address]
```

4 Implementation

4.1 Underlying Protocol

The application is built based on UDP protocol. Unlike TCP, UDP is connectionless, which means that data packets can be sent without warning, preparation, or negotiation. UDP also lacks any kind of error control. Not only can packets be delivered in the incorrect order, but they can also get completely left out. UDP is meant for applications where you are more concerned with keeping the stream of information going than making sure you receive every single packet.

For a real-time voice conferencing application, the speed/efficiency of the communication is mandatory. In a VOIP application if a packet is received out of order or few packet missings the user probably won't even notice. One of the most important factor is UDP supports broadcasting and we could able to use multicast feature on that.

4.2 Unicast Implementation

There are 3 threads used by the program for

- Sound capturing serialization sending.
- Receiving packets and deserialization.
- Play buffer to enable duplex communication.

The first iteration follows up mainly 5 steps to do the task of unicast communication

1. Initialize communication parameters by giving ip address as an argument.
2. Audio sufficient is recorded to fill the buffer.
3. Serial number is added to packets in the buffer and send packets.
4. New thread is started and packets are received in receiving party side and deserialize them.
5. New thread is started and received deserialized packets are played in buffer.

4.3 Multicast Implementation

There are 3 threads are used by the program as Unicast implementation for

- Sound capturing serialization sending.
- Receiving packets and deserialization.
- Play buffer to enable duplex communication.

The first iteration follows up mainly 5 steps to do the task of multicast communication

1. Initialize communication parameters by giving ip address as an argument.
2. Audio sufficient is recorded to fill the buffer.
3. Serial number is added to packets in the buffer and send packets.
4. New thread is started and packets are received in receiving party side and deserialize them.
5. New thread is started and received deserialized packets are played in buffer.

4.4 Packet Format

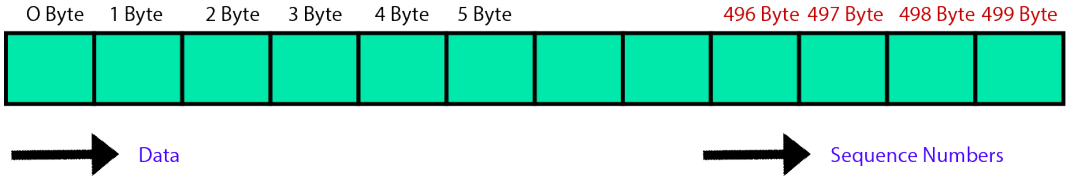


Figure 1: Packet Format

The UDP packet of the voice conferencing application consists of 500bytes by default such that 496 bytes allotted to audio signal and closing 4 bytes are allotted for sequence number. This begins from one and go up to 231. Which is more than sufficient for a single call, that is this application makes use of audio quality such that for 1 second it needs only 128 packets of data so with231 user is able to call for a long period.

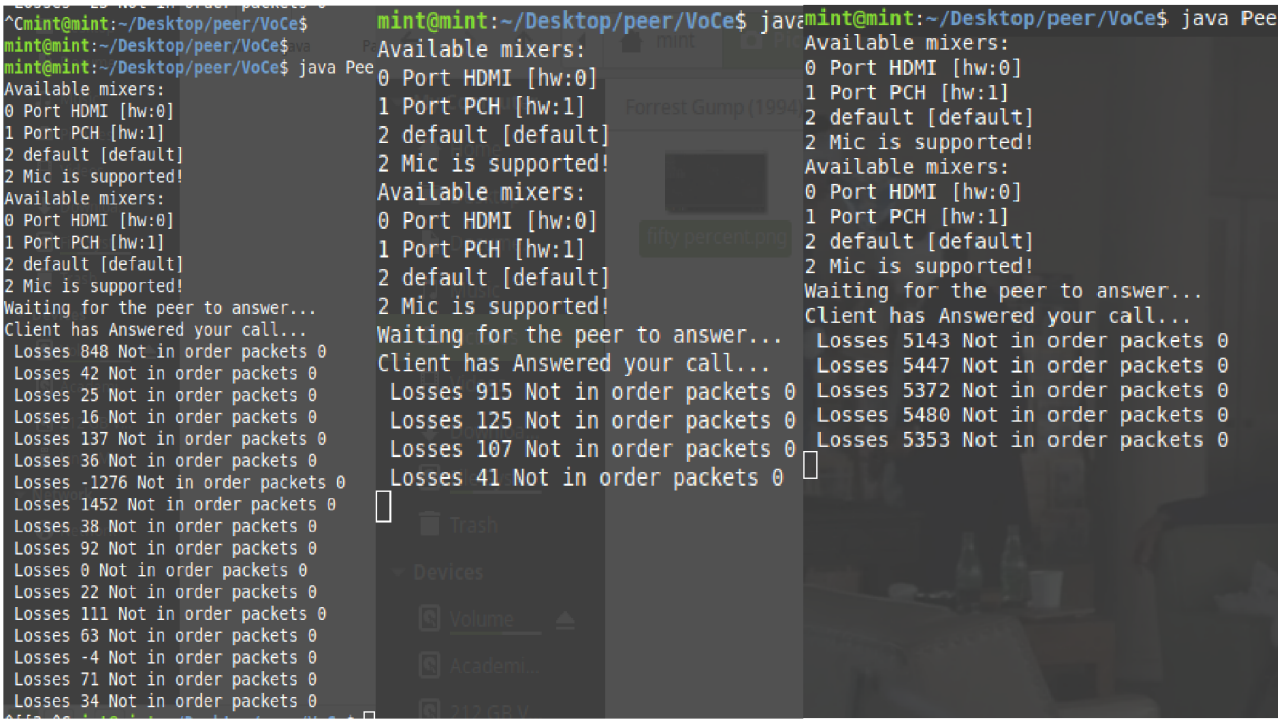
4.5 Packet lost handling and Reordering

In voice conferencing applications, packets loss can be happened. Therefore In case of packet loss, application isn't request to resend that packet in the buffer. (This implementation has a buffer of 1024 packets). In order to solve the small reordering issues automatically, our applications waits for some time to fill the buffer for weak connections. In other words the application won't play the packets as it receives unless the connection is perfect. So initially, our application stores the receiving packets in the buffer and thereafter, plays back from the buffer. There is a common variable which indicates what packet is being played currently when the voice conferencing is going on. The packet will be neglected if the receiving packets serial number is lower than the current playing because of that packet is having unallowable reordering. These severe reordered packets are discarded automatically by the application.

5 Testing & Performance

5.1 Unicast P2P with Packet loss

The application is tested under packet losses of 0%, 5%, 50%

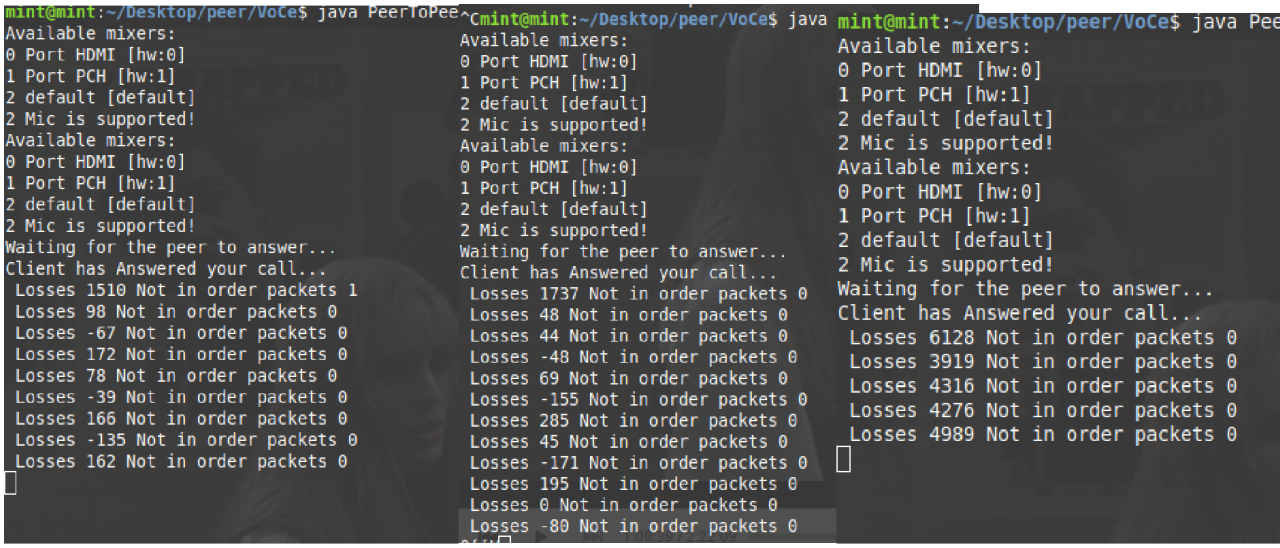


Normal Condition | **5% Packet Lost** | **50% Packet Lost**

Figure 2: Packet loss

5.2 Unicast P2P with Delay

The application is tested on delays 200ms, 700ms, 1500ms

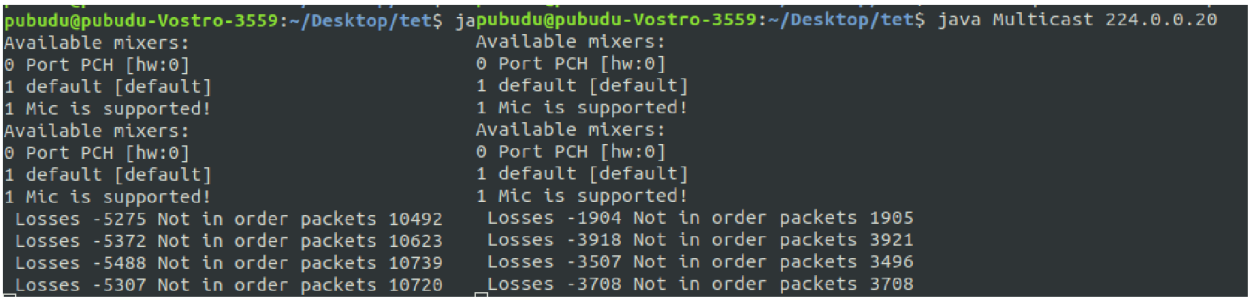


Delay of 200ms **Delay of 700ms** **Delay of 1500ms**

Figure 3: Delay

5.3 Multicast P2P with Packet loss

The application is tested under packet losses of 10%, 50%

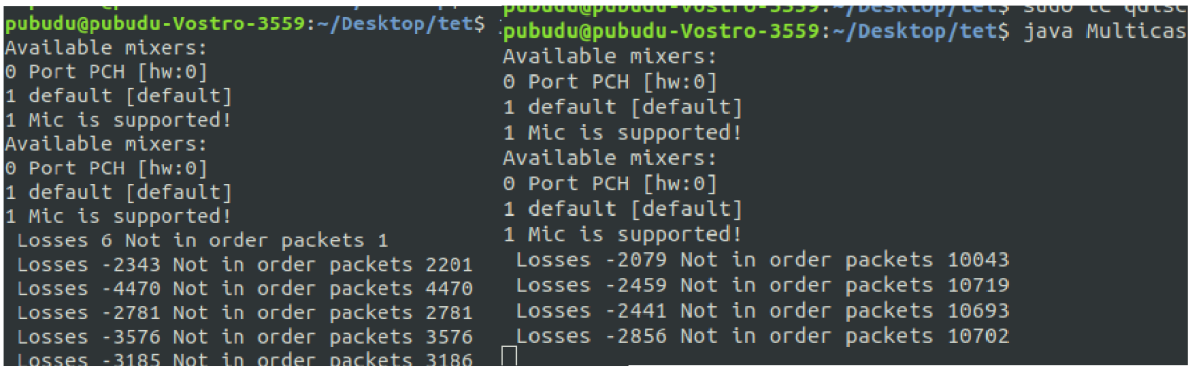


Packet Lost1 10% **Packet Lost1 50%**

Figure 4: Packet Lost

5.4 Multicast P2P with Delay

The application is tested on delays 200ms, 1500ms



Delay of 200ms **Delay of 1500ms**

Figure 5: Delay

6 Reference

1. https://docs.oracle.com/javase/tutorial/sound/sampled-overview.html
2. https://docs.oracle.com/javase/tutorial/networking/datagrams/broadcasting.html
3. https://netbeez.net/blog/how-to-use-the-linux-traffic-control/

END