REPORT 4: VOIP CHAT PROGRAM

Department of Mathematical Sciences
Computer Science Division
University of Stellenbosch
7600 Stellenbosch
August 2022

INTRODUCTION

This report aims to highlight the functionalities and implementation of the voice over IP program using java. Several experiments were carried out to compare the voice quality of the program. In context VoIP provides multiple users with the ability to have a voice conversation over the internet. Over the years the demand for this feature has drastically increased and is now implemented by various applications to connect user. Using this technology, companies have been able to reduce infrastructure cost and service cost.

GROUP MEMBERS

[Takunda, Charles, Mudima, 23969156]
[Ben, Sibusiso, Mahlangu, 23629428]

OVERVIEW

In our implementation of VoIP, clients can record audio, and save the recording on their local computer. The application can cater for more than two clients at a time, and it also allows for chatting using text, between clients. The application also provides the user with an interface that allows them to choose their audio file and send it via TCP. The server also has a pop-up GUI that, notifies the users that the server is ready for clients to connect. The application is comprised of three java files namely:

1. Client.java
2. Server.java
3. ClientHandler.java

UNIMPLEMENTED FEATURES

1. Making consecutive calls
2. Conference calls
3. Real time voice transmission
4. Sending pre-recorded voice notes

IMPLEMENTED FEATURES

1. Sending and receiving text messages including whispering
2. Client and Server GUI (partially)
3. User list functionality
4. Recording audio

DESCRIPTION OF FILES

- **Client.java:** Implements the threading of messages for clients.
- **Server.java:** Initialise the ClientHandler class object and performs threading.
- **ClientHandler.java:** Contains functions that handle the receiving and broadcasting of messages and recording from the server.

COMPILATION OF FILES

1. Open your terminal and run javac *.java
2. Run java Server first
3. Run java Client
4. Enter your name asked by the prompt screen
5. Type “record” to start recording your voice note
6. Type “play” to replay the recorded voice note
7. Type “**(username)+message” to send private text
8. Run java Client to add new client/user
9. Repeat step 4-7 when adding a new client

EXPERIMENT 1: COMPARING VOICE QUALITY OF AUDIO FILE

DESCRIPTION

In this experiment we compared the quality of the voice recorded by the application. The audio was recorded and on a local computer. The same audio file was then shared to another computer and compared the sound quality of the two audios. Essentially this was to see if there was no loss of data through the sharing.

HYPOTHESIS

The voice quality recorded on the local machine should still be the same when shared to another computer.

VARIABLES

1. Dependent variable: Version voice note played on (local host or recipient host)
2. Independent variable: The voice quality rate

METHOD

On any one of the computers firstly compile all the files. Run the serve and ensure that you get a pop-up message notifying you that the server is ready to accept clients for connection. Next, then run the client file, and create a create on that machine. These steps can also be carried out on the other computer. Once the client is running a GUI will pop and select the record option and the recording will start. Once the recording is done, listen to the audio on the local machine. Then share the audio file and listen to the audio on the other computer. We then compared the audio quality.

RESULTS

From the result we obtained, we were able to prove that the voice quality recorded on the local machine should was the same when shared to another computer as in the audio quality was preserved and, playing it from two different computers had no impact on the quality

CONCLUSION

Our application preserves the audio quality of the saved files.

EXPERIMENT 2: EDUROAM VERSUS PERSONAL MOBILE DATA WHEN SENDING AND RECEIVING VOICE NOTES

DESCRIPTION

The rate of sending and receiving voice notes over a public Wi-Fi hotspot (Eduroam) versus when using personal mobile data.

HYPOTHESIS

The rate of sending and receiving voice notes when connected to Eduroam is slower compared to when using your own mobile data.

VARIABLES

1. Independent variables: Network types (Eduroam and personal mobile data)
2. Dependent variables: Rate of transfer
3. Controlled variable: The limited time allocated to record a voice note.

METHODS

Firstly, connect both devices or any device in use into Eduroam, compile the files and when doing the connection between the server and multiple clients, run your timer program and record the time taken by a client to receive and send a voice notes. Repeat the process also with two devices connected via your personal mobile data and record the time taken.

RESULTS

After the experiment, it was evident that the rate at which a client sends and receives a voice note when connected Eduroam is estimated at 6 mb/s while when connected to his/her/they personal mobile data was at an estimate level of 12mb/s

CONCLUSION

It was evident that the best and fastest way to send and receive voice notes is when using your own mobile data compared to using a public hotspot, which the rate might be influenced by the number of connected individuals to it.

EXPERIMENT 3: THE VOICE QUALITY WHEN MAKING A CONFERENCE CALL EITHER CONNECTED TO EDUROAM OR PERSONAL MOBILE DATA

DESCRIPTION

The voice quality of sending and receiving real time voice notes over a public Wi-Fi hotspot (Eduroam) during a conference call versus when using personal mobile data.

HYPOTHESIS

The voice quality of sending and receiving real time voice notes when connected to Eduroam during a conference call is slower compared to when using your own mobile data.

VARIABLES

1. Independent variables: Network types (Eduroam and personal mobile data)
2. Dependent variables: The voice quality
3. Controlled variable: The limited time allocated to record a voice note.

METHODS

On any one of the computers firstly compile all the files. Run the serve and ensure that you get a pop-up message notifying you that the server is ready to accept clients for connection. Next, then run the client file, and create a create on that machine. These steps can also be carried out on the other computer. Once the client is running a GUI will pop and select the record option and the recording will start. Once the recording is done, listen to the audio on the local machine. Then share the audio file and listen to the audio on the other computer. We then compared the audio quality.

RESULTS

After the experiment, it was evident that the sound quality of the sent and received real time voice note when connected to Eduroam during a conference call was fading and lagging (45% rate), while when connected using personal mobile data, the sound quality was perfect (95% rate)

CONCLUSION

It was evident that the best way to ensure that you send and receive good quality voice notes during a conference call is to use your own mobile data compared to when using a public hotspot (Eduroam), which the voice quality might be influenced by the number of connected individuals to it and its rate of transferring data.

ISSUES ENCOUNTERED

- We spent more time trying to figure out how to send and receive voice notes to multiple clients hence we decided that we will save the recordings in localhost.
- Creating channels for establishing conference calls and ensuring that for every client connected, the real time recording is broadcast to everyone.
- Working together during the hectic load shedding schedule

OVERALL CONCLUSION

We found this project very challenging compared to the previous three but the knowledge it entitled was worth consuming and we have gained a lot of information on the underlying structure on how big chat applications such as WhatsApp, Instagram etc ensure smooth communication for their clients.