Project 4: VoIP Chat Program

Contents

1. Introduction

Voice over IP (VoIP) is a very popular service. The ability of multiple users to have a voice conversation over the Internet has greatly increased the value of the Internet, and given rise to IP voice switching (a method used by telephone companies to establish voice connections between users, using IP packets instead of hardware switching circuits). Using this technology, telephone companies' infrastructure costs and service costs can be reduced.

2. General Information

Please note the following points when implementing your solution:

1. You may code in any programming language, however we recommend Java. The assistants might not know your chosen language and would be unable to help you if you run into trouble.
2. You are to work in allocated groups for this project.
3. This project is to be accompanied by a report
4. The use of virtual networking tools such as Hamachi for Linux or ZeroTier for Mac must be used when communicating between the clients and the server.
5. All project source code and reports will be checked for plagiarism, including being checked against solutions from previous years. If you are found guilty there will be serious consequences, including receiving a mark of zero for the project.

3. Project Specifications

The objective of this assignment is to implement a working Voice over Internet Protocol (VoIP) system. The program should enable users to initiate voice conversations with other users, and consists of a server and of multiple clients.

Server

The role of the server is to coordinate all the activities of the clients in such a way that the clients interact as quickly and efficiently as possible.

With voice calls, the server's only role is to check if the requested user is still active, and then to initiate the session. All further communication between the two users are handled by the client programs. Text messaging between clients should also be possible. The server must adhere to the following criteria:

1. Handle all the various requests from the clients.
2. Clients will have to get permission from the server to initiate a voice conversation.
3. Act as a mediator, proxy and directory for the clients.
4. A stable environment should be created where unusual actions/requests by the clients do not disrupt the server.
5. Multiple clients must be able to connect simultaneously.
6. Handle connections/disconnections and correctly update the user list.
7. A minimal Graphical User Interface (GUI) should be used containing:
    a. A main window that outputs client activity.
    b. A list of users currently online.

Client

The role of the client is to interface with the user, translate requested operations, and interact with the server. It is essential for the client to have a GUI to act as an interface for the user.

Clients may initiate voice calls to other clients by having the user give a 'call' command and a host name to call to. Voice communication between clients are direct, without any server interference. The server should merely act as a proxy and look-up service, assisting clients in making connections. Text communication should also be possible, and voice communication should occur in real-time.

The client must adhere to the following criteria:

1. Clients must be able to disconnect/reconnect without incident.
2. A list of currently connected users must be shown, and updated as needed from the server.
3. Individual usernames are not necessary. The use of IP numbers is sufficient.
4. The ability to text to each other during a call.
5. Conference (group) calls should be implemented.
6. Conference channels (group chats) should be implemented.
7. The ability to send pre-recorded voice notes and listen to them through the GUI (not merely sending a file).
8. A call command or GUI interaction to call another user should be implemented.
9. Calls must use real-time voice transmission.
10. Voice quality should be as high as possible (at least 8000 samples per second, 16-bit sample size and 1 channel).
11. Distortions, echoes and dead zones in the audio should be minimised.
12. All voice communications should occur using the UDP protocol. Voice notes can be implemented using TCP.
13. A GUI is essential for interfacing with the users and should contain:
    a. A main window that outputs messages and descriptions of all actions performed by its users.

b. A list of users currently active, sorted by IP address (or usernames if implemented). More marks can be achieved if the user list is interactive through the GUI.
c. A ‘command’ field, where messages and commands are entered. When an unrecognised command is receive, it should be ignored and an error message should be displayed in the main window. The handling of commands is only necessary if commands are implemented, otherwise GUI interactions are fine.

4. Repos and submission

Each group shall be given a GitLab repo to work in. The contents of the repo must contain a ***README.md*** to explain how to compile and run your program, and a ***report.pdf*** to be read by your markers. For submission, compress your repo into a single tar or zip file and only one person from the group submits it to SUNLearn.

5. Helpful Hints

1. Start early, do not leave this project to the last few days while thinking that you will finish on time.
2. Look at the marking scheme and at what needs to be done to help you estimate your progress.
3. Do not think you are almost done after you have successfully had your client connect to your server.
4. It will make life easier if you run the recording, playback and the main client program as separate threads.
5. Look into Java Mixers before considering implementing your own mixing algorithm.
6. Find someone to test with, because you cannot use the same account to test your program in the labs.
7. If you use tokens to transmit commands, make sure all command characters have been added.
8. The client is much more complex than the server. Do not spend all your time perfecting the server and leave the client until the last minute. More time should be spent on the client than the server.
9. Keep in mind that the sound quality is dependent on the operating system as well as the Java JDK, so make sure to test your code in the lab where the demo will take place.

6. Useful Resources

Here are some resources that may be helpful:

- For basic Java tutorials look at the “Trails Covering the Basics” section on Sun’s Java website.
- To get started with Java socket programming, this tutorial is all you’ll need.

- Java Datagram tutorials
  http://java.sun.com/docs/books/tutorial/networking/datagrams/ index.html

7. Guidelines

Report

The page limit for the report is strictly 8 pages. Fit the following sections into this page limit:

1. Title
   Your document must have a title which reflects that of the project you are implementing. Furthermore, you must indicate the names, student email addresses and roles of all group members.

2. Introduction / Overview
   Explain what the purpose of the project was and give an overview of the main sections in the document in a short paragraph, not a table of contents.

3. Features
   Since you will implement most features, indicate which of the required features you did not implement. Also indicate what additional features you implemented.

4. File Descriptions
   Provide concise descriptions of the source files present in your project. You may ignore files that have minimal functionality, for example a file containing only two member fields.

5. Program Description
   Provide a description of your program flow. We suggest that you tailor this according to the design of your implementation. Since most implementations will be event based / reactive, we recommend that you structure your program flow description as follows. Start with a birds-eye view of what components your implementation comprises of, how these components interact with one another, and what purpose of each is.
   Finally, explain the program flow that takes place for all features as specified in this document. Example of features include client disconnection handling, client interaction, and message routing.

6. Experiments
   The report should have at least three experiments done, where each experiment follows the scientific method which includes a question and/or hypothesis, experimentation, and results/conclusion. Make sure you provide the data that you have gathered, as well as identify the dependent and independent variables for each experiment.

7. Issues Encountered
   Give a brief description of any problems or issues that you encountered.

8. Design
   Mention any interesting design decisions that you made in your implementation.

Source Code Documentation

1. Class level

2. You must use Javadoc format to provide a short summary of the program description from the report, i.e. a birds-eye overview of the main functionality for each class.
3. Method/Function level
   Again, use Javadoc format. Generally, methods/functions should be documented with the following:
   - A single line describing the main utility
   - A paragraph describing the algorithm
   - Inputs, outputs and return value
   - Caveats and limitations
4. In-line level
   Use comments sparingly inside method bodies; for intricate code only. Don't state the obvious. Also, provide comments that explain why you are doing something rather than what you are doing, since what you are doing should be obvious from the method name.

8. Marking Scheme

Client GUI	**8**
Display list of users	1
Update list of users	2
Interactive list of users	2
Listen to voice notes	2
Displayed actions of users	1
Server GUI	**3**
Display list of users	2
Display actions of users	1
Stability	**20**
Concurrently send/receive messages from many users (not group)	2
Concurrently send/receive messages from many users (in group channel)	3
Concurrently send/receive voice	5
Make consecutive calls	2
Receive consecutive calls	2
Call when in call	2
Receive call when in call	2
Connections/disconnections	2
Functionality	**7**
Voice notes	3
Send/receive messages while in call	3
Clients only request calls via server	1
Quality	**8**
Distortion	2
Echos	2
Dead zones in audio	2
Minimal delay (2 seconds max) beween transmission and playback	2
Report	**16**
Include all sections specified in report guidelines	8
Language, grammar and spelling	3
Project description	2
Experimental hypotheses and results	3
Documentation	**8**
Class level	2
Method level	2
In-line level	2
Readme present	2
General	**5**
Presentation quality	2
Hamachi/ZeroTier	3
TOTAL	**75**