Durarara!

Hrishikesh Karale
hhk9433@rit.edu

1 Overview

The purpose of this document is to provide an update status report on the design and implementation of our CSCI60501.2145 semester project: a voice-over-IP(VOIP) java chat application.

2 Introduction

Voice over IP (VoIP) is a very popular service. The ability of multiple users to have a voice conversation over the Internet is a valuable one. VoIP greatly increased the value of the Internet and gave rise to IP voice switching, a method used by new 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. Global telephone networks may now be set up, unbounded by country borders or political policy.

3 General Specification

The objective behind the voice chat application is to implement a working Voice over Internet Protocol (VoIP) system. The application will be used over the local LAN and should enable users to initiate voice conversations with other users. The application will consist of a server and multiple clients.

3.1 The Server

The role of the server is to coordinates all the activities of the clients in such a way that the clients interact as fast and efficient as possible. The Server should implement a minimal GUI to give feedback on client activities. The server will accept connections from multiple clients, but multi-threading is

not necessary. When a client terminates its connection, it should be done in a clean fashion where the server operations are not disrupted.
With voice calls, the servers only role is to check and see 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. Users should also be able to send text messages and these should be correctly displayed by the receiving clients

3.2 The 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 communications, between clients are direct, without any server interference. The server should merely act as a proxy and look-up service, assisting clients in making the connections. Text communications should also be possible. Voice communications should occur in real-time

4 Project Specifications

1. A Server that co-ordinates all the request to and from clients and adhers to the following criteria:

   - Handel all the various requests from the client.
   - Client will have to get permission from the server to inittiate a voice conversation.
   - Handel connections\disconnections and correctly update the user list.
   - A minimal GUI should be used containing:
     - A main window hat outputs client activity.
     - A list of users currently active.

2. A Client that interfaces with the user and adhers to the following criteria:

   - A Client must be able to disconnect\reconnect without incident.
   - A list of currently connected users must be shown, and updated as needed from the server.

- A Call command to call another user must be implemented.
- Real-Time voice transmissions.
- voice quality should be as high as possible.
- A minimal GUI should be used containing:
  - A main window hat outputs message and descriptions of all the actions performed by it's users.
  - A list of users currently active usres, sorted by host-name.

3. All voice communications shouldoccur using UDP protocol.
4. Sound quality should be atleast 8000 samples per second, 16-bit sample size and 1 channel.

5 Implementation

Below are the goals that have been achieved so far.

1. Client-Server connectivity.
2. Simultaneous trnasfer of text from client to server and vice versa.
3. File transfer
4. Sending and recieving text messages during voice calls.
5. Voice Input from device default microphone and output from default speaker
6. Server and Client GUI.

6 Future Implementations

1. Working on voice quality, distortion, echos and dead zones in audio.
2. Sending and recieving of voice concurrently.
3. Sending pre-recorded voice messages.

7 References

Here are few of the resources that were helpful:

- Code example of how to record sound injava:
  http://www.developer.com/java/other/article.php/1565671/Java-Sound-An-Introduction.htm
- For basic Java tutorial:
  http://java.sun.com/docs/books/tutorial
- To get started with Socket Programming:
  http://java.sun.com/docs/books/tutorial/networking
- Datagram tutorials:
  http://java.sun.com/docs/books/tutorial/ networking/datagrams/index.html
- GUI:
  http://java.sun.com/docs/books/tutorial/ui