|  |
| --- |
|  |
| Fuzzy hipster |
| COMP 4985 Final Assignment |
|  |
| **Aaron Lee (A00621294)**  **Behnam Bastami (A00197752)** |
| **18/03/2013** |

Table of Contents

[Client 3](#_Toc351285529)

[Overview 3](#_Toc351285530)

[Overall STD 3](#_Toc351285531)

[Overall Pseudocode 4](#_Toc351285532)

[File Download/Upload 5](#_Toc351285533)

[Download/Upload STD 5](#_Toc351285534)

[Download Pseudocode 6](#_Toc351285535)

[Upload Pseudocode 6](#_Toc351285536)

[Streaming 7](#_Toc351285537)

[Streaming STD 7](#_Toc351285538)

[Streaming Pseudocode 7](#_Toc351285539)

[2-way Microphone Chat 8](#_Toc351285540)

[2-way microphone STD 8](#_Toc351285541)

[2-way microphone Pseudocode 8](#_Toc351285542)

[Multicasting 9](#_Toc351285543)

[Multicast STD 9](#_Toc351285544)

[Multicast Pseudocode 9](#_Toc351285545)

[UI 10](#_Toc351285546)

[Server 11](#_Toc351285547)

[State Transition Diagrams 11](#_Toc351285548)

[Overview 11](#_Toc351285549)

[Upload Request 12](#_Toc351285550)

[Download Request 13](#_Toc351285551)

[Stream Request 14](#_Toc351285552)

[2-way Microphone Chat 15](#_Toc351285553)

[UI 16](#_Toc351285554)

# Client

## Overview

### Overall STD



### Overall Pseudocode

#### Get User Settings:

* Allow user to choose between server and client modes
* Allow user to enter server address and port number
* Validate settings; if invalid, display error and prompt user to input valid values

#### Create control channel

* Create a TCP socket
* Connect the socket to the servers TCP socket
* Retrieve song listing from server and populate song list on GUI
* Retrieve connection info (port number) for multicast from server

#### Wait for user command

* This is the listen for command state on client
* Valid user commands are :
  + - * Request to Download a music
      * Request to Upload a music
      * Request to stream a music
      * Request to start 2-way microphone communication
      * Join the server’s multicast channel

## File Download/Upload

### Download/Upload STD



### Download Pseudocode

#### Send DL request

* Send a packet to the server requesting a file download
* The request is in the following form
  + [ header | filename ]
    - Header includes the following
      * Type: REQDL
      * Size: size of the filename string

#### Wait for file

* Create a new thread and wait for the file packets to arrive

#### Save to file

* While receiving file packets, save to a file
* When EOF received, return to listen for command state

### Upload Pseudocode

#### Send UL request

* Send a packet to the server requesting a file upload
* The request is in the following form
  + [ header | filename ]
    - Header includes the following
      * Type: REQUL
      * Size: size of filename

#### Wait for approval

* Create a new thread to wait for server approval
* If approved, server will echo the request packet back to the client

#### Send File

* While not EOF Packetize the file and send over TCP
* When finished, go back to listen for command mode

## Streaming

### Streaming STD



### Streaming Pseudocode

#### Send stream request

* Send a packet to the server requesting a file stream
* The request is in the following form
  + [ header | filename ]
    - Header includes the following
      * Type: REQST
      * Size: size of the filename string

#### Stream song

* Create a new thread and wait for the file packets to arrive
* While receiving file packets, play the music
* If a stream is already in progress when another stream request is made
  + Kill the previous stream
* Send a new stream request

## 2-way Microphone Chat

### 2-way microphone STD



### 2-way microphone Pseudocode

* Stop any other streaming that is in progress
* Create a microphone UDP channel on a new thread
* When the user starts the microphone chat, read data from microphone and send to server over the UDP channel
* When the user stops the microphone chat, perform cleanup and close the socket

## Multicasting

### Multicast STD



### Multicast Pseudocode

* Stop any other streaming that is in progress
* Create a new thread to join the server’s multicast channel
* While receiving packets on the multicast channel, play the music data

### UI



# Server

## **State Transition Diagrams**

### Overview



### Upload Request



### Download Request



### Stream Request



### 2-way Microphone Chat



### UI

