|  |
| --- |
|  |
| Fuzzy hipster |
| COMP 4985 Assignment 3 |
|  |
| **Aaron Lee (A00621294)**  **Behnam Bastami (A00197752)** |
| **24/01/2013** |

Table of Contents

[Client 3](#_Toc351244169)

[State Transition Diagrams 3](#_Toc351244170)

[Overview 3](#_Toc351244171)

[File Download/Upload 4](#_Toc351244172)

[Streaming 5](#_Toc351244173)

[2-way Microphone Chat 6](#_Toc351244174)

[Multicasting 7](#_Toc351244175)

[UI 8](#_Toc351244176)

[Server 9](#_Toc351244177)

[State Transition Diagrams 9](#_Toc351244178)

[Overview 9](#_Toc351244179)

[Upload Request 10](#_Toc351244180)

[Download Request 11](#_Toc351244181)

[Stream Request 12](#_Toc351244182)

[2-way Microphone Chat 13](#_Toc351244183)

[UI 14](#_Toc351244184)

# 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

#### Create control channel

* Create a TCP socket
* Connect the socket to the servers TCP socket

#### 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 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
* Create a new thread and wait for the file packets to arrive
* While receiving file packets, save to a file
* When EOF received, return to listen for command state

### Upload Pseudocode

* 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 servers approval
* If approved by server, create a new thread
* 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 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
* 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

