

Joshua Labasbas

Music Recommendation Protocol

Internet Engineering Task Force  
INTERNET-DRAFT

Joshua Labasbas  
University of Florida  
April 2023

## Music Recommendation Protocol

### **Abstract**

Music Recommendation Protocol(MRP) is a music genre and band recommendation protocol that is based on the client's mood. MRP is lightweight, simple, and easy to impement in numerous applications while at the same time has plenty of room for future features and scales. This document will review the format of data between the client and server, compilation of an example app based on the command line, and possible features that could be implemented in the future.

## **Contents**

## 1. Introduction

MRP provides a simple way for users to discover music in a lightweight and simple way. The hope is for users to be able to find new music that fits their mood. Hopefully in the future a more complex backend can be created for the app to provide a more efficient way to recommend music. But the basic crux of the protocol, the client side will send a number relating to a certain emotion over to the server, and then the server will send over first the number of lines, or the size, of the recommendation, and will send over the actual text strings.

## 2. Compilation and Execution on Linux Machines

The application can be compiled on Linux machines by using the 'javac' command, followed by the title of the file, either 'musicServer' or 'musicClient.' The compilation was performed on the computer lin114-05 in the CISE computer lab over SSH. Also over SSH on the same machine, the server was run and then a client was connected to the server from a Windows machine on the same network. In order to run the server on the Linux machine, using the java command to execute the musicServer command followed by the port number for the server to run on; over testing, the port number was 3960. In order to run the client application on the Linux machine, the same java command is used followed by the name of the application, musicClient, and then the address of the server followed by the port number on the server to connect to.

## 3. Protocol Description

In depth, the idea behind the protocol is a streamlined way for the client to simply express the way the client feels and then delivering a music recommendation based off of that. Often, a music listener might want to try to find some new music and does not know where to start, the goal of the protocol is to make finding new music as simple as possible for the user, supplying simply the way they feel. The client side application sends over data pertaining to the how the user feels, in this implementation, this data is described as an integer. Then the server sends the size of the recommendation and then sends the actual text string.

## 4. Application Code Structure

The application itself is incredibly simple, meant as a proof of concept to be expanded upon with future efforts. The client side application begins by connecting to the server's welcome socket and then prompting the user for their feeling, printing out the welcome message and the various options for them choose from. After the user's choice is selected, the client side application will write to the socket the integer that describes the user's choice. Then the server will receive this integer, known as mood, then the server will select a recommendation(for the sake of the application, the recommendations are hardcoded). Then the server will send over an integer containing the number of lines that the recommendation takes up then the server will send over the recommendations. Finally, on the client side, the application will read each recommendation string, knowing how many times to read from the recommendation size it received from the server, and then print out the strings to the user.

## 5. Possible Future Features

The protocol and the example application has a lot of room to grow. For example, the applicaiton could have a stronger backend to deliver personalized and dynamic recommendations for the user depending on past ratings of recommendations and personal preferences. As of right now, the recommendations are hardcoded strings on the server side, so a more efficient and dynamic method of recommendation would be a starting point to expand on the example application. In terms of expanding the protocol, future versions can allow for the user to stream the audio file directly instead of getting the recommendation and then finding some way to listen to the music with another source. Considering the protocol making use of TCP, converting the audio file into a byte array and then transmitting the byte array to be played to the user over TCP is within the realm of possibility.