# MUSIC COLLAB: An IoT and ML Based Solution for Remote Music Collaboration (Student Abstract)

**Nishtha Nayar[1], Divya Lohani[2]**

Shiv Nadar University, Greater Noida, India
[1]nn469@snu.edu.in, [2]divya.lohani@snu.edu.in

## Abstract

Communication using mediums like video and audio is essential for a lot of professions. In this paper, interaction with real-time audio transmission is looked upon using the tools in the domains of IoT and machine learning. Two transport layer protocols - TCP and UDP are examined for audio transmission quality. Further, different RNN models are examined for their efficiency in predicting music and being used as a substitute in case of loss of packets during transmission.

## Introduction

As the entertainment industry is growing at a rapid rate, the need for collaboration is growing along with it. It is majorly seen in the music industry where musicians are seeking out to create music across geographical boundaries. This brings a lot of financial and logistical challenges and hinders the creativity of the artists. Through this research project, we would focus on solving the issue at hand using the advances in IoT which are already being used for smart cities and the likes, along with using ML. This use case can further be applied to areas like real-time video game streaming and similar multimedia applications.

## Motivation and Objectives

The main motive behind this project was to take steps towards bridging the gap between distant musicians who want to pursue their art together. There have been applications like Jacktrip (Cáceres and Chafe 2010) which aim to enable audio collaboration. Researchers at CCRMA have also tried to analyze and test audio transmission over physical distances (Chafe 2003). Further, researchers at Stanford have studied various other ways of music composition using Naive Bayes, neural network and encoder-decoder models (Kang, Kim and Ringdahl 2018). The main objective is divided in two parts:

- Coding and analyzing various server-client scenarios for the transport layer for the IoT application.
- Training an efficient model to predict and substitute for any missed-out or miscommunicated notes to prevent the real-time music from stopping.

## Experimental Design

### Networking Using TCP and UDP

Transmission Control Protocol (TCP) is a transport protocol which enables two hosts to establish a connection to send data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent, whereas User Datagram Protocol (UDP) uses connectional communication and transfers the data using datagrams and is suitable for purposes where error checking and correction are not necessary. For the transport layer, various scenarios were coded and tested using WireShark over the LAN network at the Shiv Nadar University lab. The cases are given below:

- Server-client communication where the client chooses an MP3 song stored locally in the server to be played. This is done using TCP and UDP.
- Server-client communication where client sends an MP3 song to the server to be played on. This is done using TCP and UDP.
- Server-multi client communication where one client chooses and MP3 song stored locally in the server and client machines. The song is then played on all clients using TCP.

### Predicting Music Using RNNs

The machine learning aspect of the project involved using the Google AI environment Magenta over MIDI files. The MIDI dataset used for the training was on Reddit[i] and 11,000 files of it containing classical piano, guitar, violin

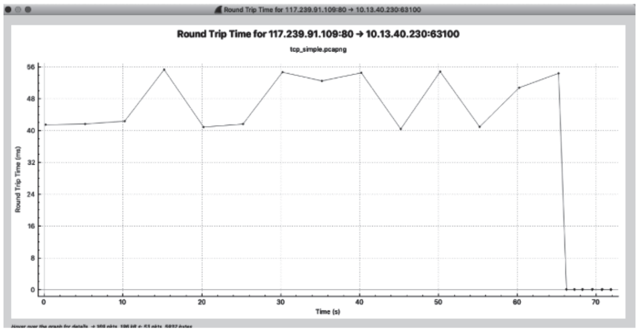

Figure 1: RTT for TCP (local)

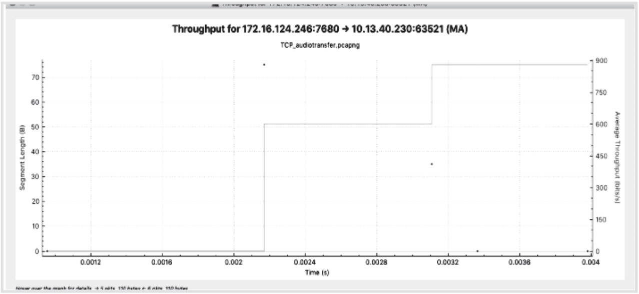

Figure 2: Throughput for TCP (local)

were used. There were three models trained using Magenta's Melody RNN: Basic RNN, Lookback RNN and Attention RNN. Training of the models went as follows:

- MIDI files are all converted into NoteSequences which are protocol buffers and are faster and efficient.
- A TFRecord file of NoteSequences was generated. These NoteSequences are converted into SequenceExamples which contain a sequence of inputs and a sequence of labels that represent each melody.
- The models are then trained one by one using the sequence example files. The number of training steps used at 10,000 for Basic and Lookback RNN and 8000 for Attention RNN.

The parameters of the models were evaluated using Tensor Board. These models are also used to generate 10 MIDI files using a primer MIDI of the song 'River Flows in You' by Yiruma. The model generates 3 bars on the 5 bar song. The results were not evaluated by a validation set, rather they were validated by real musicians.

## Results and Discussion

It is found that TCP is a better protocol since UDP protocol did not provide enough data regarding RTT, latency, packet loss and throughput and the graphs for the latter gave no evident quantitative data, whereas TCP gave consistent results regarding the same. Various graphs like Figures 1 and 2 depict the various parameters obtained for TCP.

| | Min. Loss | Max. Accuracy | Min. Perplexity | Log Likelihood Values |
|---|---|---|---|---|
| Basic | 0.77 | 0.7315 | 2.5 | -260.5 to -171.3 |
| Lookback | 0.91 | 0.7321 | 2.55 | -272.4 to -186.1 |
| Attention | 0.99 | 0.7131 | 2.71 | -280.4 to -232.9 |

Table 1: Parameters Obtained for RNNs

It is also found that machine learning and RNN provides decent generated melodies which can be used to substitute for any lost or corrupted music during communication and hence help in an uninterrupted music collaboration of musicians at different geographical locations. While Basic RNN gave very repetitive and a few unmusical results with gaps, Attention and Lookback RNN gave significantly more pleasant and more musical melodies.

From Table 1, numerically, the training accuracy for Basic RNN is maximum and its loss is minimum. However, there is a very minuscule difference between these metrics of the three models, hence according to experimental data and evaluation of the generated music by real musicians, Attention and Lookback RNN are better than Basic RNN. A further user study can be conducted to confirm the use of this predicted music in real jamming sessions.

## References

Cáceres, J., and Chafe, C. 2010. Jacktrip: Under The Hood of An Engine for Network Audio. *Journal of New Music Research*, 39:3, 183-187. doi.org/10.1080/09298215.2010.481361.

Chafe, C. 2003. Distributed Internet Reverberation for Audio Collaboration. In P*roceedings of Twenty Fourth Conference on Advances in Engineering Software*. Banff.

Kang, D; Kim, J.; and Ringdahl, S. 2018. Project Milestone: Generating Music with Machine Learning. Technical Report, Department of Computer Science, Stanford University, CA.

[i] www.reddit.com/r/datasets/comments/3akhxy/the_largest_midi_collection_on_the_internet/