Superluminal Communication Voice Codec

Software Engineering Project 2024, Semester 2

Introduction

It is widely believed that superluminal communication (communication faster than the speed of light) is not physically possible. One of the few remaining physical phenomena which may achieve this only theoretical - a valid solution to the equations of General Relativity known as a spacetime wormhole[1]. Many science fiction writers have postulated that wormholes could be constructed and then used to allow humans to easily travel between distant stars etc. However, if one were ever to be created, it would be at enormous cost and energy and any first applications would likely involve *information* transfer only at very high cost per bit!

As Computer Scientists, let's assume they will come along one day and that one of the first applications we have to develop is instantaneous voice communication to a future human colony on Mars. Present voice communication to Mars is not very conversational because even at closest approach to Earth a radio signal travelling at the speed of light takes over 3 minutes[2] to reach the Red Planet and the same again for a response. Superluminal communication via a wormhole that can transmit bits of information could make this instantaneous and so turn communication to people on Mars into a regularly paced conversation.

Because of the expected extremely high cost per bit, it is imperative that we take the speech and represent it with as few bits of information as possible in the communication channel. This step is known as encoding and, although widely studied with a long history, is again a wide open field of development with the advent of AI and machine learning techniques. At the other end of the channel the bit stream must be decoded back into something intelligible again.

Your project is to write a voice encoder and decoder (combined known as a codec) that achieves maximal compression whilst still maintaining intelligibility. This project will be run simultaneously with many teams in a competition format where each team has the opportunity to find the best tradeoff between compression and intelligibility to win the final prize.

[1] https://www.britannica.com/science/wormhole

[2] https://www.space.com/24701-how-long-does-it-take-to-get-to-mars.html

Functional Requirements Detail

Audio Codec

Your audio codec *name* must either be provided as *name*.py, assumed to be a Python script, or *name*.exe, assumed to be a standalone executable.

It will be run like this during the encoding step:

```
name --encode < input.wav > payload.bin
```

And then like this during the decoding step:

```
name --decode < payload.bin > output.wav
```

In other words, your codec must be a single script/exe that is able to run in two modes depending on the command line flag given and be able to deal with binary files on standard input and output. Your codec may have additional dependencies that can be provided together with the entry point script/exe in a file *name*.zip.

The codec will be run on a virtual machine running Windows on Azure. The .wav file is a single channel LPCM[3] sampled at 44.1kHz with 16 bits per sample consisting of human voice audio, but other than that, you know nothing about its contents.

compression = (1 - size_in_bytes(payload.bin) / size_in_bytes(input.wav)) * 100%

accuracy = a qualitative measure of how intelligible output.wav compared to input.wav between 0-100%

Good luck and have fun!

Both compression and accuracy will be shown on the competition leaderboard.

There is no limit on the number of entries you can make into the competition throughout the semester. The best teams will set up a development pipeline of ideas that exhibit continuous improvement throughout the course, both in competition performance and in software engineering practice.

Your final mark is not determined by your competition results or position on the leaderboard but rather the team processes and software engineering practices your group adopts in getting to your best result.

[3] Linear Pulse Code Modulation aka uncompressed audio, consisting of regular samples at regular amplitude quantisation increments.