Networks Lab 02

Hands-On PHY Layer

Design Document

22b0915, 22b0931, 22b1014, 22b1022

Contents

1 Introduction

In this report we outline the design and implementation of the system for transmitting and receiving a message using sound signals. The system consists of a Sender and a Receiver, where the Sender encodes a message into sound and sends this using PyAudio Python library , and the Receiver decodes the sound back into the original message but to emulate the real world scenario of bit errors in transmission we will flip at max 2 bits in the original message bit string. To incorporate this error detection and correction we will be using Cyclic Redundancy Check (CRC) algorithm to ensure that the receiver gets the correct message even after bits error.

2 System Design

The system is designed to handle the transmission of binary messages with a length of up to 20 bits. To ensure reliable transmission, we are going to transmit the bitstring in packets of 5 bits for a duration of 1 second, allowing for clear differentiation between the two frequencies at the receiver's end. Some redundant bits are added to the message before the transmission following CRC algorithm at the sender's end and then verification happens on the receiver's end.

2.1 Bit Encoding

The bit encoding process involves converting the 5 bits packet into a decimal which will correspond to a particular frequency, now there can be 32 possible frequency encoding which gets detected with a little error also and frequency-based encoding ensures that even with potential noise interference, the differences in frequency can be accurately detected by the Receiver.

2.2 Preamble and Message Structure

To get a reliable reception of the message, a preamble is transmitted before the actual message. The preamble is a known sequence of bits that is giving us the length of the original message which also allows the Receiver to synchronize with the incoming signal and accurately detect the start of the message. The message structure includes the preamble followed by the message bits, and then the CRC bits for error detection.

3 Error Detection and Correction

Error detection and correction are critical for ensuring the integrity of the transmitted message. In our system, we will be using a 10-bit CRC algorithm to generate redundant bits that are appended to the original message. These CRC bits allow the Receiver to detect and correct errors that may have occurred during transmission.

3.1 CRC Algorithm

The CRC algorithm operates by dividing the message by a predefined polynomial (in this case, we are using 11-bit polynomials). The remainder from this division is the CRC value, which is appended to the message. During reception, the Receiver performs the same division to verify the integrity of the received message. If the recalculated CRC matches the received CRC, the message is considered error-free.

However, if the recalculated CRC does not match the received CRC, it indicates that there are bit errors in the transmitted message. To handle these errors, we will maintain a precomputed table that maps possible remainders (when the message is divided by the polynomial) to the indices of the error bits. Specifically, this table will store the mappings of $\binom{m+k}{2}$ possible remainders, where

m is the message length and k is the length of the CRC remainder bits.

When a mismatch occurs, the Receiver will look up the recalculated remainder in this table. The table will provide the indices of the bit errors, allowing the Receiver to correct them and recover the original message. This approach utilizes the fact that certain patterns of errors produce specific remainders, which can be used to get the exact location of the errors in the message.

4 Reception and Decoding

The Receiver side is responsible for capturing the sound signal, decoding the bits, and verifying the message's integrity using the CRC bits. The reception process involves the following steps:

4.1 Sound Reception

The sound signal is captured using the microphone, and the signal is sampled at a high rate of 44100 i.e. after obtaining 44100 samples of frequency it will conclude it as a single interval and then perform Fourier Transform to get the maximum amplitude frequency to ensure accurate detection. The sampled signal is then processed to extract frequencies corresponding to each packet.

4.2 Frequency Detection

To determine the value of each bit, the Receiver side will perform a Fourier transform on segments of the received signal using the fft module of the scipy library. The frequency with the highest amplitude in each segment is identified, and the corresponding decimal value is determined based on the detected frequency via the encoding which then is converted to binary string.

4.3 CRC Verification

Once the message bits are decoded, the CRC bits are extracted, and the CRC value is recalculated from the received message. If the recalculated CRC matches the received CRC, the message is correct. If there is a mismatch, the Receiver will identify the bit errors and correct them.

5 Conclusion

Using this system will provide a reliable method for transmitting binary messages using sound signals. The use of frequency-based encoding, Fast Fourier Transform for getting dominant frequencies and CRC error detection ensures that messages are accurately transmitted and received, even in the presence of bit errors in the transmitted message.

References

[1] S. Babaie et al. "Double Bits Error Correction Using CRC Method". In: *2009 Fifth International Conference on Semantics, Knowledge and Grid*. Zhuhai, China, 2009, pp. 254–257. DOI: 10.1109/SKG.2009.77.

[2] Vivien Boussard et al. "Enhanced CRC-based correction of multiple errors with candidate validation". In: *Signal Processing: Image Communication* 68 (2018), pp. 170–181. DOI: 10.1016/j.image.2018.05.004.

[3] Wikipedia. *Cyclic Redundancy Check (CRC)*. Accessed: 2024-08-25. 2024. URL: https://en.wikipedia.org/wiki/Cyclic_redundancy_check.