Networks Lab 2: Design Draft

Brian Mackwan(22b0413), Ekansh Ravi Shankar(22b1032),
Shravan S(22b1054), Ved Danait(22b1818)

August 2024

Transmission and Reception

1. We maintain a set of three frequencies corresponding to three input values : $\mathbf{0} - 1000$, $\mathbf{1} - 4000$, $\mathbf{2} - 500$.

2. We use the input value "2" as a sort of buffer to separate adjacent bits. Our encoding is as follows :

$$bit[i] \xrightarrow{encode} bit[i] + \text{`2'}$$

   So for example, 10110 will become (12)(02)(12)(12)(02)

3. For each value in the encoded message, we play the corresponding frequency. For the sender, the **duration** for each value is **0.2** seconds. Meaning in a 0.4 second interval we can relay a single bit in the original message which has 0.2 seconds of the original message and 0.2 seconds of the buffer frequency.

4. In the receiver, we have tentatively kept the **duration** as **0.1** seconds. This decreases the probability of error since we break the tone into more windows, meaning that we have higher chances of catching each distinct tone.

5. We keep the **threshold** as **2 Hz** on either side of the corresponding frequencies.

6. We collect each and every discernible tone at the receiver which is within 2 Hz of any of our frequencies. We now need to post-process this string. This is simply done by iterating through the array and storing each value which is not the same as the previous value. Finally, just remove all '2's from the string to obtain the original message.

Error Correction

1. We use **Cyclic Redundancy Check** to correct the errors.
2. To perform CRC, we use an appropriate divisor polynomial, which when applied on messages with 20 or lesser bits, has a **Minimum Hamming Distance of 5** in the final message which is the initial message and the appended remainder.
3. Thus, when two error bits are introduced, we can simply brute force by **taking all possible pairs** as error bits, and flipping them and checking which one gives a **remainder of zero** when dividing by the divisor polynomial used.
4. This works because all the strings in our space have a minimum Hamming Distance of 5, and hence there will be a **unique pair which has to be flipped** to give the solution.
5. The **divisor polynomial** to be used is

   $$x^{10} + x^8 + x^6 + x^5 + x^4 + x + 1$$

   According to `https://users.ece.cmu.edu/~koopman/crc/` and local tests conducted, this polynomial works for all messages of sizes upto 20 bits, which means the final message transmitted will be upto **30 bits** (this is only the message and remainder), because the degree of the divisor is 10.