**CSE 561 Android Project Report**

Lilian de Greef & Calvin Loncaric

9 May 2014

**Modulation Scheme:** amplitude modulation

**Number of bits per symbol:** 1

**Data rate achieved:** 540 bits per second

**Bit error rate:** depends on distance, see below

**Maximum distance at which the BER is still less than 2%:** 17.5 cm

**Source repository:** <https://github.com/Calvin-L/glowing-llama>

**DESIGN DECISIONS**

**Signal Format**

Our signal consisted of 4410 Hz frequency sine waves and silence to represent *one* and *zero* bits, respectively. Packets contained a *preamble*, *header*, and *payload* in this respective order.

The *preamble* consisted of a total of 32 bits of alternating *one*’s and *zero*’s.

The *header* consisted of 16 bits representing the payload size. We stopped including the source and destination because the system only uses two phones.

The *payload* was capped to have a maximum length of 256 bits. Longer messages were split into multiple packets.

**Decoding Algorithm**

To decode a packet picked up from the microphone, after detecting the preamble, our algorithm

1. Takes the absolute value of the raw audio signal.
2. Applies a max-filter to this absolute value (with a window size of two wavelengths of our 4410 Hz signal) to smooth the results.
3. Detects the preamble by looking for alternating high and low points.
4. After a preamble is detected:
   1. Applies a threshold (based on the amplitude of the packet’s preamble).
   2. Breaks up the thresholded signal into bins, based the expected length of a bit, realigning at *one*-bits.
   3. Counts the number of high samples within each bin to determine if they contain a *one* or *zero* bit.

**EVALUATION**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Distance (cm)** | **Errors per Trial Run (out of 9856 bits sent)** | | | | | | | | | | **Average**  **Error Rate** |
| 2.5 | 0 | 91 | 0 | 4571 | 0 | 0 | 0 | 384 | 0 | 0 | 5.12% |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| 7.5 | 0 | 17 | 36 | 0 | 0 | 0 | 0 | 90 | 3 | 642 | 0.80% |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3668 | 0 | 0 | 3.72% |
| 12.5 | 31 | 775 | 705 | 5454 | 746 | 0 | 0 | 562 | 5359 | 4933 | 18.84% |
| 15 | 5465 | 1610 | 5965 | 2613 | 1308 | 385 | 1297 | 1305 | 2158 | 1367 | 23.82% |
| 17.5 | 3 | 0 | 82 | 7 | 36 | 385 | 14 | 18 | 65 | 0 | 0.62% |
| 20 | 89 | 580 | 18 | 55 | 674 | 2 | 1089 | 3 | 613 | 1 | 3.17% |
| 22.5 | 6114 | 1 | 101 | 20 | 26 | 207 | 4373 | 15 | 451 | 26 | 11.50% |
| 25 | 1 | 1 | 6462 | 7 | 4710 | 6466 | 6112 | 3677 | 6386 | 4955 | 39.34% |
| 27.5 | 6043 | 7769 | 7381 | 6112 | 6716 | 6112 | 6721 | 8747 | 9092 | 6043 | 71.77% |
| 30 | 57 | 3326 | 0 | 7380 | 9087 | 6043 | 4662 | 6720 | 7501 | 8740 | 54.30% |

Our error rate was lowest when the phones were around 5-7.5 cm apart. Between 12.5-15 cm we observed a marked decrease in detection accuracy that we cannot fully explain.

The primary sources of error were failure to detect the preamble and failure to correctly decode the header. A missed packet (or one with the wrong length) would cause all of the subsequent data to be shifted over, resulting in very low overall accuracy. Every error resulting in more than 1200 incorrect bits in the table above was a result of a missed packet (or a packet with an incorrectly interpreted length).

We have two main ideas for improving overall accuracy:

* Adaptively determine how many high samples should make a bin count as a *one* based on the preamble (we already adaptively select a threshold in this manner)
* Add an error-correcting code to the header to make that critical part of decoding more robust