BattMan: Acoustic short-range communication leveraging ultrasound

A Project Report Submitted
in Partial Fulfillment of Requirements
for the Degree of

Bachelor of Technology

by
Amit Panghal (100050016)

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai 400076, India

Abstract

Short-range communication systems have been largely dominated by RF-based systems like 802 and bluetooth . An alternative could be using inaudible sound for communication. Using sound has clear advantages since it doesn't require additional hardware. It requires a speaker and a microphone, easily available in all cellphones. In this thesis, we present BattMan, an android application for short-range communication using inaudible frequencies. We explore the acoustic channel and do bit-error rate analysis at various frequencies to come up with operational frequencies.

Honor Code

I, Amit Panghal, certify that I have properly cited any material taken from other sources and have obtained permission for any copyrighted material included in this report. We take full responsibility for any code submitted as part of this project and the contents of this report.

Certificate

It is certified that the B. Tech. project "Acoustic short-range communication leveraging ultrasound" has been done by Amit Panghal (100050016) under my supervision. This report has been submitted towards partial fulfillment of B. Tech. degree requirements.

Bhaskaran Raman
Associate Professor, Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai-400076, Maharastra, India

Contents

Chapter 1

Introduction and Motivation

In this section, we will get familiar with some terms and then motivate the need for battMan.

1.1 Defintions

1.1.1 Crowdsourcing

Crowdsourcing is a process used for subdividing a hard problem by obtaining contents, services or ideas from a undefined group of people(online committee in this context) rather than traditional employees or suppliers, thus simplifying the problem in hand. When such communities contribute sensory information to a central body of knowledge, it is called participatory sensing. A smartphone is pleothera of sensors. It has wide range of sensors like GPS, accelerometers, gyroscope etc. Since smartphones are in widespread, everyday use and are always connected, they offer a great platform for extending existing web-based crowdsourcing applications to a larger contributing crowd.

1.1.2 ITS

ITS(Intelligent Transport System) are set of techniques using sensors and wireless communication to alleviate the problem of road traffic congestion. Such techniques aims to provide real-time information about road traffic congestion and public transportation systems, and enable its users to plan the travel over a less congested route, avoid peak hours, or use a public transportation systems in an informed manner, etc. Using smartphones as the sensing devices can help simplify the problem. GPS location of commuters can be the information crowdsourced. For example, the real-time information such as current route of a bus or number of commuters in the bus can help users plan their travel.

1.1.3 GPS Duty-Cycling

GPS Duty-Cycling is the process of turning GPS on and off depending on some policy. These processes can be divided into two broad categories, *static duty-cycle* (SDC) approaches, which turns GPS receivers on/off at regular intervals and *dynamic duty-cycle* (DDC) approaches, which adjust GPS cycles based on some model or events triggered. Both ADC and DDC have advantages and disadvantages. For example, DDC require additional sensors as triggers and look up tables to handle such events while SDC are implemented blindly without considering spatial and temporal correlation of signal reception.

1.2 Motivation

Crowdsourcing smartphones application are limited by the incentives it can provide participating users. Negatives factors such as high-battery usage by an application can deincentivize the particpation. GPS being a power hungry sensor, is major limitation for applications dependent on real time GPS information. This work is part of the project CARTS, Communication Assisted Road Transport System [1], which looks at the problem of road traffic congestion in developing regions from a broader perspective. The goal is to amortize smartphone power usage by sharing GPS load among commuters within a bus, using short-range communication.

1.2.1 GPS Lock Mechanism and Accuracy

The GPS module when turned on, attempts to acquire lock and is indicated by a flag in the output data packet. Once it has the lock, the GPS reports horizontal dilution of precision (HDOP), which is an estimate of error in the horizontal plane. This value tends to start high, perhaps several tens of metres, and falls over consecutive samples to a few metres. The process of obtaining lock and an estimate with acceptable HDOP is referred to as obtaining a GPS position lock [2] . Whenever the GPS is powered off, the position uncertainty grows progressively with time until it acquires the next lock.

1.2.2 Tradeoff: Energy and Accuracy

There is a tradeoff between energy efficiency of a MLSA and Location Accuracy, and is dependent on error tolerance limit in the estimate. For CARTS [1], this information is used to determine real-time traffic scenario and help commuters to plan their journey. In developing countries like India where traffic behaviour can be very random, the acceptable error should be less compared to similar system in developed region. So, the error-acceptance threshold must be a parameter of the application and can be determined by experimentation. Campus bus service can be used as the testbed to determine this threshold, as part of the future work.

1.2.3 GPS duty-cycling algorithm

The key ideas for this GPS-duty cycling algorithm are taken from ADC with radio-ranging [2]

The figure 1.1 shows the algorithm that will be running on each device. U is location uncertainity, AAU is the location uncertainity threshold, Sc is the current speed and s is the assumed speed of the device at a given time. Clearly, the parameter that governs the algorithm at any moment is location unceratinity. Effect of sharing the GPS load can be derived by sharing the location uncertainty. This process of sharing the information is explained in next section.

Figure 1.1: Algorithm from [2]

```
loop {t}
    Choose speed()
    if GPS == LOCK then
        position ← GPS position
        U ←Ugps
        lastlocktime = now()
        GPS = OFF
    else
        U ← U + (t * Sc )
        If GPS == OFF and U+(t*s)> AAU then
            GPS = ON
        end if
    end if
end loop
```

Figure 1.2: location uncertainity estimate sharing mechanism

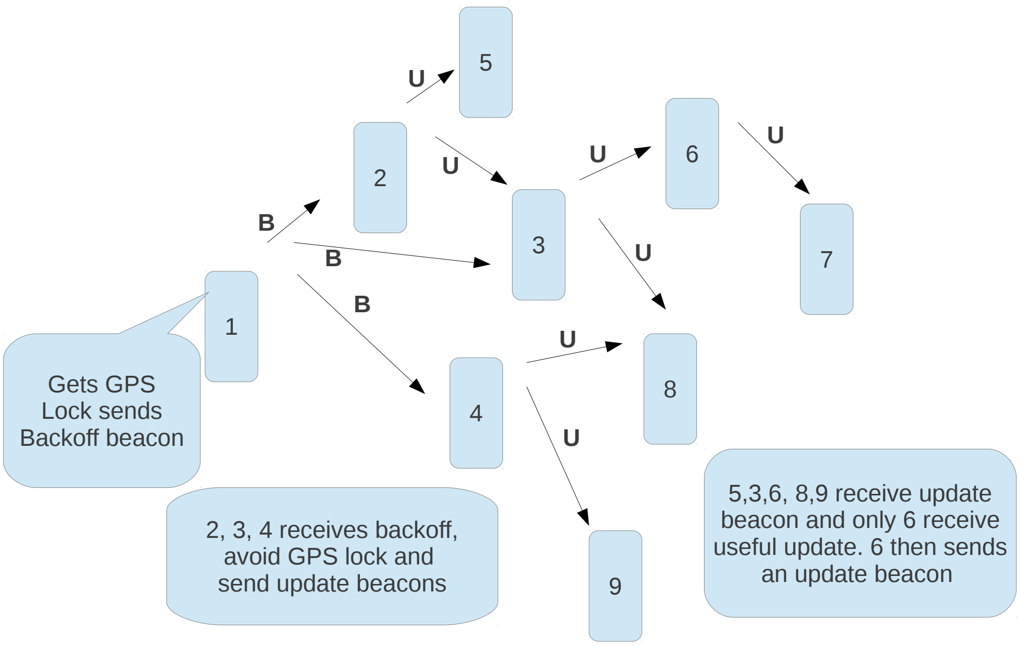


Table 1.1: Comparison of various technologies for contact logging

Features	WiFi	Bluetooth	Sound
hardware	yes	yes	yes
range	35-50m	10-15m	0.5-1.5m
bitrate	5-50mbps	1 mbps	80-150 bps
connection-oriented	yes	yes	no
power-consumption	highest	high	lowest

1.2.4 Participatory Sensing

If nearby nodes can exchange their location information, the uncertainity can be reduced and they can keep their GPS off for longer duration. For example, Node A has some uncertainty say Ua, and receives a beacon from Node B with uncertainity Ub. if $Ua > Ub + range$, then node A updates it location and uncertainity. Thereby increasing off time and hence energy savings. Range depends on technology we use, in our case it is 1-2m. The beacons can be of two types:

- back-off beacon
- update beacon

back off beacon can be triggered when a node is attempting a GPS lock, asking others to wait for a time before attempting a GPS lock; update beacon is triggered when

- a useful update beacon is received
- a GPS lock attempt is successful

1.2.5 Technology for short-range communication

Hardware support available in the smartphones makes various technologies feasible for the task. Some of the pros and cons of using them for our purpose are shown in table 1.1. Though WiFi and bluetooth seem to be a good alternative compared to sound if we look at range and bitrate. The problem with WiFi and bluetooth is that they are connection-oriented and the time consumed in discovering access points or devices take time. Moreover, keeping WiFi and bluetooth always on is a costly affair. Given that we are only transferring only a few bytes of data and the range is within our application requirements, low power consumption and connection-less nature of acoustic communication become the deciding metrics. Moreover, hardware requirement for acoustic communication is minimal and is easily available in all kinds of mobile-phones. Hence, we choose sound as a technology for short-range communication.

This report aims to build an short-range communication system using sound as the medium which could serve as a building block for participatory GPS-Duty cycling application for amortizing battery power.

Chapter 2

Previous work

Using audible signals to transmit information between electronic devices has been popular in the past. Researchers have build on these ideas to come up with coding-schemes that allow data to be transmitted and received through air using speakers and microphones.

Context-Aware Computing with sound [3] proposes audio-networking using ubiquitously available sound hardware for low-bandwidth, localized wireless networking. They have implemented various audio data transmission schemes like Dual-Tone Multi-Frequency (DTMF), On-Off Keying(OOK), Inaudible Data Transmission and melodic data transmission. For inaudible data transmission they have used a variant of OOK and used 21.2 kHz, greater than human hearing threshold and less than nyquist frequency of standard sound cards, as carrier frequency. They have achieved a throughtput of 8 bits/second with packet-receipt rate of 95%.

Dhwani [4] presents a novel, acoustic-based NFC system that uses microphone and speaker on mobile phones. It suggests use of frequencies from 6kHz-12kHz for a peer-to-peer acoustic NFC application. The paper mentions that, for frequencies above 6kHz, interference by ambient noise is close to phone's noise floor and hence, can be neglected. It discourages the use of frequencies above 12kHz since attenuation increases with increase in frequency. Dhwani provides information-theoretic security with help of their unique feature, Jam-secure. Jam-secure thwarts eavesdroppers by jamming the signal it is receiving and uses Self-Inteference Cancellation (SIC) to decode the signal. They are able to achieve data rates of upto 2.4 Kbps, sufficient for most NFC applications.

Short-range ultrasonic communications in air using quadrature modulation [5] studies Quadrature Phase Shift Keying as modulation scheme for Short-range ultrasonic communication system. They implement it on experimental communication system and show the feasiblity of such system over 1-3 meters, using frequencies in the 200-400 kHz range.

Chirpcast [6] broadcasts WiFi access keys using ultrasonics. They experiment and study various modulation techniques and achieve data-rates of upto 200 bits/ second using differential phase-shift keying at 90% accuracy.

Idea of using ultrasonic channel for data-transfer has been recently gaining popularity among various startups. Zoosh [7] uses ultrasound for payment using smartphones. In Infosound [8] technology, ultrasonic frequencies are used to send short information codes from speakers that are received by the microphones in mobile terminals. A mechanism used in conjunction with a server, converts these codes or acoustic IDs into URL information and users in the vicinity of the speaker are able to pick up coupons or access e-commerce sites.

Chapter 3

Acoustic Channel

In order to design an communication system, we must first study the characterstics of the communication medium or channel. In this chapter, we look at key features like acoustic propagation, acoustic noise and acoustic channel.

3.1 Modulation Schemes

Let's discuss various modulation schemes that could be used in our application.

3.1.1 Frequecy Shift Keying

This scheme passes information using discrete changes in frequency of carrier wave. FSK (BFSK) uses two frequencies to transmit 0 and 1. Let these frequencies be f1 and f2. This scheme offers greater noise immunity than the simpler On-Off Keying, since the absence of both f1 and f2 during a transmission indicates that an error has occurred.

3.1.2 Phase Shift Keying

It conveys digital data by modulating phase of the carrier wave. PSK uses a finite number of phases and maps them to a unique pattern of bits or symbol. The demodulator uses same symbol set as the modulator, determines the phase of received signal and maps it to the matching symbol to recover original data. There are two fundamental ways to utilize phase of the signal to convey information.

1. Using phase itself to convey information, requires modulator to have reference signal to compare the received signal against.

2. Using change in the phase to convey information, i.e differential scheme, doesn't require any reference signal at demodulator.

Following sections schemes using above techniques respectively.

Binary Phase Shift Keying

It is simplest form Phase Shift Keying and uses two phases separated by 180 degrees. This is the most robust of all PSK's since it require highest level of noise or distortion to make a modulator reach an incorrect decision. Since it only modulates 1 bit/symbol, it is unsuitable for high-data rate applications.

Differential Phase Shift Keying

It is based on mapping between the phase difference of carrier wave and symbols. For example, a logical one may correspond to adding pi to the current phase and a logical zero may correspond to adding 0 to the current phase.

For sake of simplicity we have chosen Frequecny Shift Keying(FSK) as our modulation technique. FSK uses two type of frequencies, one for transmitting a '0' bit and other for transmitting '1' bit.

3.2 Frequency Selection

Selecting the appropriate frequencies for battMan is an important design decision. First of all, all frequencies must be compatible with microphone/speaker's selectivity. Since most of the phones support acoustic samping rate of up to 44 kHz, we can choose a operating frequency of value upto 22 kHz. Selecting a frequency between 6-12kHz as suggested by Dhwani, audible to human adults, has a fundamental issue with respect to requirements of the application. The audible random noise will decincentivize the use of application. We decided to go ahead with frequencies inaudible to human adults. We found out that frequency above 16kHz are hardly audible to human adults and frequencies above 20KHz suffer from high attenuation to be of any good. We analyzed following frequencies:

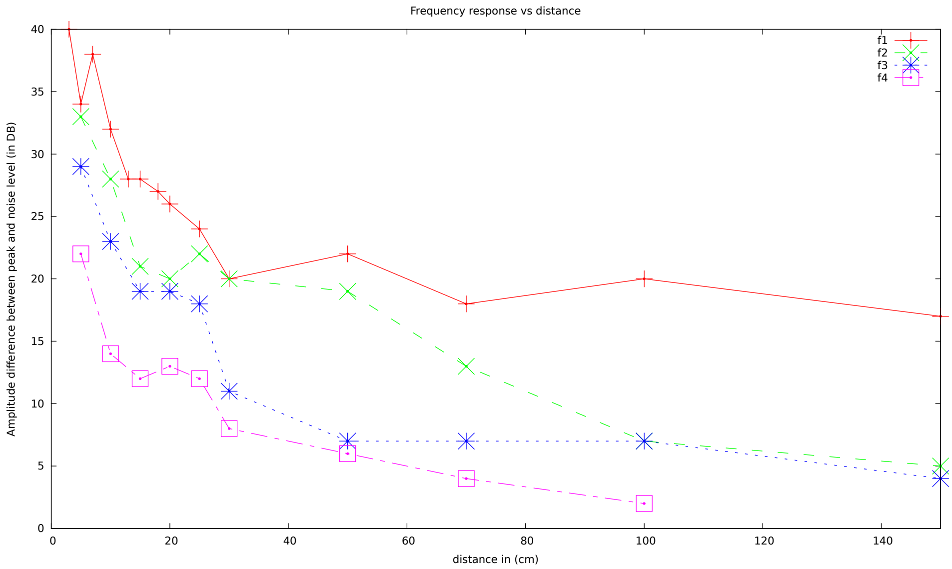
- 16.5 kHz
- 17.5 kHz
- 18.5 kHz
- 19.5 kHz

We carried out an experiment on these frequencies to determine the maximum acceptable range between nodes for minimum signal loss. We sent tones at frequencies stated above from a Karbbon Titanium S5 smartphone to a lenovo ideapad and used audacity for recorded signal analysis. We have plotted a graph 3.1 between the amplitude of peak above noise floor and distance between devices. Looking at graph we can see that attenuation increases as distance increases and the increment is more for higher frequencies.

3.3 Bit-error Analysis

Bit-error study is required to get proper estimate of operational range of the frequencies to be used in battman. We carried out this bit-error rate experiment for audible frequencies 12.5 kHz , 13.5 kHz , 14.5 kHz and 15.5 kHz apart from the inaudible frequencies mentioned in above section. The study at audible frequencies was done for range-comparison and exploration into possible extension of current work, i.e using audible frequencies in battMan for better data rates. The experimental set-up includes a Google nexus pad and Lenovo Ideapad Z580. We modified the battMan for experiment and sent continous stream of symbols to receiver at various distances. The receiver on Lenovo Ideapad Z580 laptop is a desktop application written in java that is used to analyze received signal and calculate bit-error at various distances for given frequencies. The audible and inaudible

Figure 3.1: Peak difference vs distance graph for 4 frequencies used in battman



frequencies are analyzed and plotted separately. At receiver, fft of received samples equivalent to a bit is done and then amplitude at given frequencies is calculated. If amplitude of frequency being analyzed is not maximum, we increment the number of bits-in-error and finally a bit-error % is calculated at that particular distance for the frequency.

The graph 3.2 is plotted for audible frequencies , 12.5 kHz (S), 13.5 kHz (0), 14.5 kHz (1) and 15.5 kHz (E). As expected,we can observe that bit-error % increases as distance between the sender-receiver is increased except for few dips. The bit-error % for higher frequencies is more compared to lower. If we consider 10 % bit-error acceptable then the maximum range for communication using 12.5 kHz, 13.5 kHz ,14.5 kHz and 15.5 kHz is 5 m, 3.75 m, 3.5 m and 1.2 m respectively.

Similarly, the graph 3.3 shows bit-error plot for inaudible frequencies 16.5 KHz (S), 17.5 KHz (0), 18.5 KHz (1) and 19.5 KHz (E). The graph shows behaviour similar to previous graph. Going by 10 % bit-error acceptance, the maximum range for 16.5 KHz, 17.5 KHz, 18.5 KHz and 19.5 KHz are 1.2 m, 1 m, 0.8 m and 0.75m respectively.

Finally, after carrying out bit-rate analysis and keeping in mind our application requirement we decided to go with frequencies 16.5 KHz and 17.5 KHz for sending '0' and '1' respectively.

Figure 3.2: bit-error vs distance graph for audible frequencies (12.5 kHz, 13.5 kHz, 14.5 kHz, 15.5 kHz)

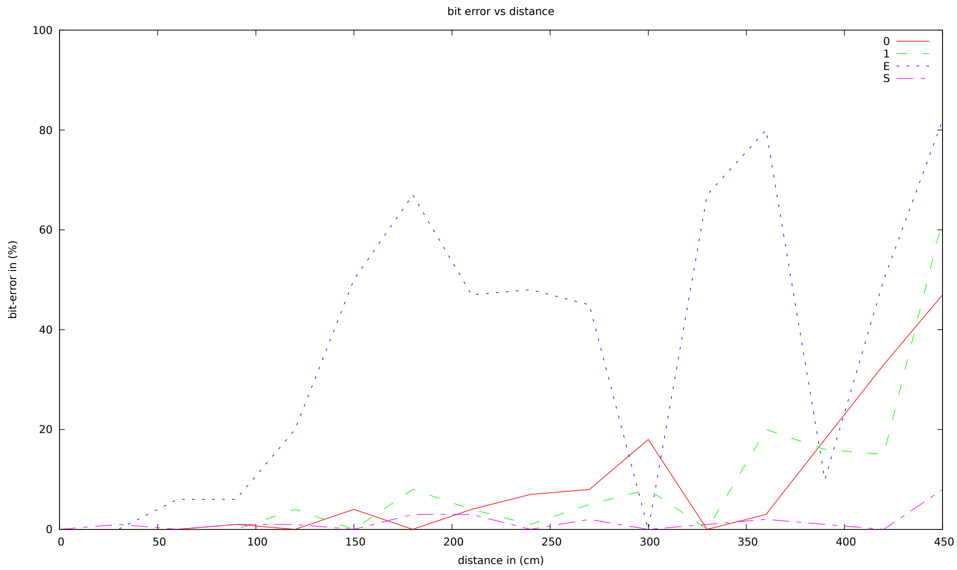
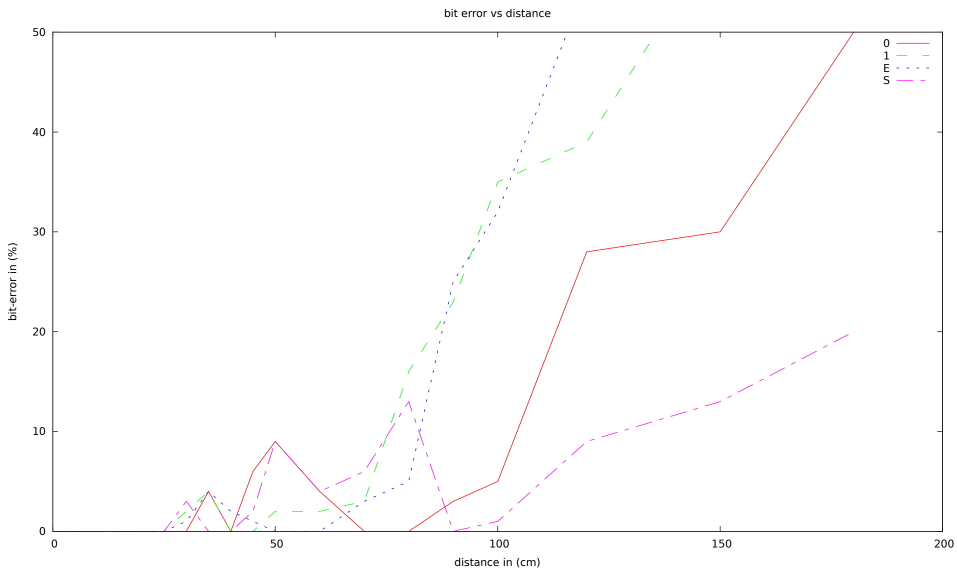


Figure 3.3: bit-error vs distance graph for inaudible frequencies (16.5 kHz, 17.5 kHz, 18.5 kHz, 19.5 kHz)



Chapter 4

BattMan : SmartPhone Application

We have built an android application for sending messages from one device to another using inaudible frequencies. BattMan has two major components, the sender and the receiver.

4.1 BattMan : Sender

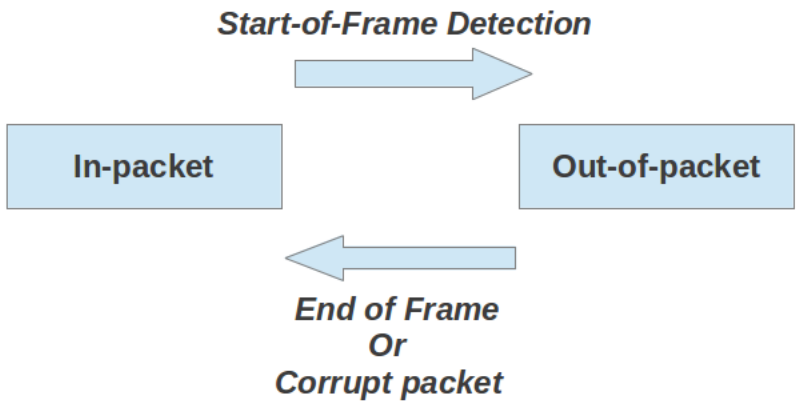
The sender sends the data inputted by user. The data is converted into bit-stream and then divided into fixed size packets according to maximum packet size fixed (300 bits). Figure 4.1 shows the structure of a typical packet. A packet consists of a header, data and a CRC value calculated over entire packet added at the end of packet to detect transmission errors. Packet header consist of a packet-length field and a CRC value calculated over the header at the end. This packet is then sent to physical layer for transmission. Following section is detailed on how transmission takes place.

4.1.1 Physical Layer : Packet Transmission

The modulation scheme used is Frequency Shift Keying (FSK), 16.5 kHz and 17.5 kHz are operational frequencies used for transmitting '0' and '1' respectively. The sample rate of the transmitter is 40960 Hz and data rate is 10 bits per second. Each packet is prefixed with a short preamble symbol ('0101') for start of packet detection on receiver side. Each bit is transmitted by sending a tone at the mapped frequency for bit-duration (0.1 seconds).

Figure 4.1: packet structure used in battMan

Figure 4.2: State machine used in receiver



4.2 BattMan : Receiver

The tasks of receiver is to correctly receive and decode the packet transmitted by sender. The networking stack at receiver includes physical layer and application layer for receiving and decoding the packet respectively. We will look at the detailed fuctionalities of these two layers in following sections.

4.2.1 Physical Layer

The physical layer tasks include start of frame detection and bit synchronization to receive data packets. The receiver samples at 40960 Hz and in one iteration 4096 samples are read, i.e equivalent to 1 bit since bit-duartion is 100 millisecond. FFT of these samples is carried out to detect the dominant frequency and hence the bit. Now we move on to how start-of-frame is detected and bit-synchronization is implemented in battMan.

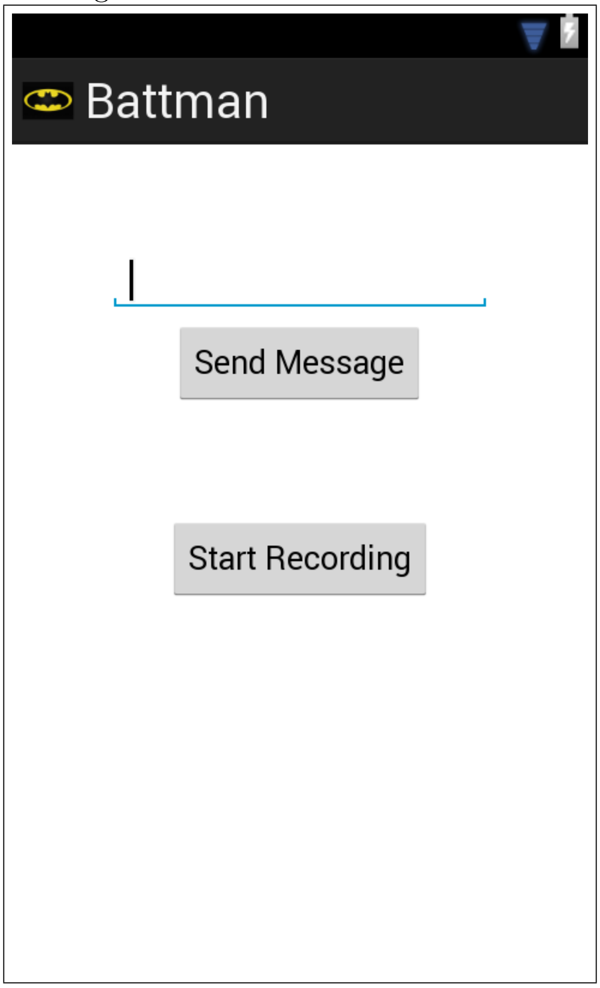
Start-of-Frame Detection

As we know sender sends a short preamble symbol i.e '0101' before every packet, the task is to detect this symbol. We maintain a state machine 4.2 with two states at receiver, out-of-packet and in-packet respectively. Initially, we are in out-of-packet state. After detecting the preamble signal, we change the state to in-packet and change back to out-of-packet once we have either successfully received a packet or packet-header/packet is corrupt. For detecting preamble, we maintain an window of 4N samples, where N corresponds to 1-bit worth of samples i.e 4096. For every fresh N samples read, we have N possible 4N windows and for every window we check these two conditions,

1. if the window corresponds to preamble symbol i.e "0101"
2. first half and second half of the window are highly correlated.

For second condition, we use pearson-correlation function. It gives values between -1 and +1, -1 for total negative correlation, 0 for no correlation and +1 for total positive

Figure 4.3: screenshot of BattMan



correlation. We consider given two windows to be correlated if the value of pearson-correlation cofficient is greater than 0.95.

If both conditions mentioned above are satisfied, we have successfully detected the preamble and can change the state to in-packet.

Bit-Synchronization

Bit-Synchronization is ensured by start-of-frame detection and fft-sample size corresponding to a bit. Start-of-frame detection ensure we don't cross the bit boudary and correctly receive rest of the frame.

4.2.2 Application Layer

Application layer processes the packet, check for transmission errors and decode the packet to get data. The header is processed to get packet length and CRC value in header is cross-validated to ensure packet-header is not corrupt. If values match, we get the CRC value for whole packet at the end using packet-length and validate it against the calculated value. If either of packet-header or packet is corrupt, we drop the packet and move to out-of-packet state. Else, we process furthur to get the data.

Chapter 5

Future work

5.1 Improvements in BattMan

The current version of battMan sends message from one device to another. If two devices in vicinity were to send at the same time, it might lead to inteference since there is no inherent support for medium-sharing in protocol stack. We need to add CSMA-like support for our audio-networking stack for making it compliant with our requirements. The bit-rates achieved with simple modulation scheme like Frequency Shift Keying are very low. So, we need to try alternatives like phase-shift-keying and differential-phase-shift keying for better rates. Alternatively, we can go ahead with audible frequencies as suggested by dhwani and try melodic data transmission, improving data-rates without the irritating random noise. One can easily snoop on the transmissions and get the data since the current version of battMan has no support for information security. For starters, the jam-secure technique proposed by Dhwani [4] can be implemented and added to battMan.

5.2 Integration with CARTS

Integration of BattMan with ongoing CARTS [1] project can be next logical step. It will be added to crowd-source application [9]. The combined application can be deployed for test on IITB campus's bus network. Experiments can be carried out to measure power-gain by smartphone due to BattMan and effect of location error estimates on the overall application.

Chapter 6

Conclusion

This report explores the benefit of using sound as a medium for short-range communication. We have done range experiments and bit-error analysis on inaudible frequencies to validate their compatibility in our scenario. We have come up with battMan, an android application and implemented the basic version which sends message from one device to another using ultrasound.

Acknowledgement

I would like to thank my guide, Prof. Bhaskaran Raman for the regular discussions and directions, without which I would not have managed it this far. I would like to thank Aman mangal and Bo Wei (phd student, UNSW, Austrailia) for their inputs on filters and signal processing. I would also like to thank Dr. Prasant Mishra(Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore) for inputs on acoustic communication.

References

[1] Communication assisted road transport system. http://www.cse.iitb.ac.in/silmaril/br/doku.php?id=proj:carts.

[2] Raja Jurdak, Peter Corke, Dhinesh Dharman, and Guillaume Salagnac. Adaptive gps duty cycling and radio ranging for energy-efficient localization. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 57–70, New York, NY, USA, 2010. ACM.

[3] Anil Madhavapeddy, David Scott, and Richard Sharp. Context-aware computing with sound. In *IN PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING*, pages 315–332, 2003.

[4] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. Dhwani: secure peer-to-peer acoustic nfc. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, SIGCOMM '13, pages 63–74, New York, NY, USA, 2013. ACM.

[5] Chuan Li, D.A. Hutchins, and R.J. Green. Short-range ultrasonic communications in air using quadrature modulation. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 56(10):2060–2072, October 2009.

[6] Francis Iannacci and Yanping Huang. Chirpcast: Data transmission via audio, 2010.

[7] Zoosh. http://www.verifone.com/.

[8] Yamaha infosound. http://research.yamaha.com/network/infosound/.

[9] Nikhil Patil, Vivek Paharia, Bhaskaran Raman, and K.K Ramakrishnan. Content centric information crowdsourcing and dissemination for better public transportation.

[10] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware design experiences in zebranet. In John A. Stankovic, Anish Arora, and Ramesh Govindan, editors, *SenSys*, pages 227–238. ACM, 2004.

[11] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 87–101, New York, NY, USA, 2007. ACM.

[12] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 299–314, New York, NY, USA, 2010. ACM.

[13] Mikkel Baun Kjærgaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjær. Entracked: Energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 221–234, New York, NY, USA, 2009. ACM.

[14] A. Madhavapeddy, R. Sharp, D. Scott, and A. Tse. Audio networking: the forgotten wireless technology. *Pervasive Computing, IEEE*, 4(3):55–60, 2005.

[15] Ted Tsung-Te Lai, Chun-Yi Lin, Ya-Yunn Su, and Hao-Hua Chu. Biketrack: Tracking stolen bikes through everyday mobile phones and participatory sensing. In *Proceedings of 2nd International Workshop on Sensing Applications on Mobile Phones (Colocated with SenSys 2011)*, Seattle, WA, 2011.

[16] Chao-Lin Wu, Yu-Te Huang, Cheng-Lung Wu, Hao hua Chu, Polly Huang, and Ling-Jyh Chen. An adaptive duty-cycle scheme for gps scheduling in mobile location sensing applications. In *Proceedings of 2nd International Workshop on Sensing Applications on Mobile Phones (Colocated with SenSys 2011)*, Seattle, WA, 2011.

[17] Lu Hong, Yang Jun, Liu Zhigang, Lane Nicholas D., Choudhury Tanzeem, and Campbell Andrew T. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, New York, NY, USA, 2010. ACM.

[18] Chirp.io. `http://chirp.io`.