Beacon-Beep: Light-Weight High-Accuracy Localization System for Smartphones

Mostafa Uddin
Old Dominion University
muddin@cs.odu.edu

Tamer Nadeem
Old Dominion University
nadeem@cs.odu.edu

ABSTRACT

In this paper, we design, implement and evaluate the Beacon-Beep localization system. The design goal of the Beacon-Beep system is to to develop a light weight, high accuracy, energy efficient and privacy preserving localization system for off-the-shelf smartphones. The Beacon-Beep system leverages both the acoustic interface and the WiFi interface of the smartphone at kernel level to achieve high accuracy in user's location estimation. This system is a combination of both ranging based and fingerprint based localization system. The Beacon-Beep system does not require any central controlling system to schedule collaboration between nearby devices. Furthermore In beacon-Beep system, each user's smartphone works autonomously to estimate it's location. We implement the complete Beacon-Beep system in a commercial off-the-shelf smartphone. Our result shows that Beacon-Beep system can achieve less then a meter of error for more than the 90% of the time for both indoor outdoor environments.

1. INTRODUCTION

In many indoor environments (e.g., airport terminal, railway station, shopping mall, and office building), knowing the location of the user would enable several interesting application and services. For example, accurate indoor guidance, efficient network management, generation of safety alerts, access to merchandise and promotion information, analyzing the popularity of different section in the store, movement of the passenger etc. Numerous academic works have been done in the area of indoor localization. Most of the localization research works have been based on Radio Frequency (RF)-based techniques that leverages signal strength of RF signal from different nearby RF sources or infrastructures (e.g., WiFi access point, cellular tower). Many researchers are combining multiple modalities such as sound with the WiFi to achieve high accuracy localization system [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. For example, the localization schemes [1, 2, 3, 4], utilize the acoustic based ranging [11, 12, 13] scheme and combine it with the RF at the application layer. Recently, researchers have also focused on developing indoor localization system for off-the-shelf smartphones [1, 2, 5, 6, 7, 8, 14]. Based on the literature review, we summarize the following problems for the existing methods in achieving accurate indoor localization system for smartphones:

- RF based approaches with sophisticated localization algorithm can only achieve limited accuracy with error range of 6-8m. Recent works show that existence of the same signature or fingerprint of RF signal at different distinct location prohibits to have high accuracy in indoor localization system [1, 2, 7].
- Those with high accuracy indoor localization system require customized hardware, which is not supported in smartphones. In addition, some systems require extra infrastructure hardware [3, 4, 15, 16].
- Recent proposed high accuracy indoor localization systems for smartphones [1, 2] require a central controller to schedule the collaboration between neighbor devices. This requirement hinders the usage of these localization systems in real practical environments. Besides, these systems overlook the privacy and energy consumption issues.

In this paper, we design and develop a light-weight high-accuracy indoor/outdoor localization system (*Beacon-Beep*) for off-the-shelf smartphones. The Beacon-Beep system leverages both the acoustic interface and the Wi-Fi interface of the smartphones at the kernel level to achieve high accuracy in user's location estimation.

The proposed system uses a combination of both ranging based and fingerprint based localization approaches. In the ranging-based approach, Beacon-Beep utilizes the Time Difference of Arrival(TDoA) between the acoustic and the radio-frequency (RF) signals by leveraging the slow propagation speed of the acoustic signal with respect to the RF signal to estimate relative ranges. In the fingerprint-based approach, since the internal structure of the indoor environments could significantly affect the propagation of the acoustic signals, Beacon-Beep identifies and extracts acoustic features as unique fingerprints for indoor locations.

The Beacon-Beep system does not require any central controlling point to coordinate between nearby devices. In this system, each user's device (i.e. smartphone) works autonomously to determine its location. Furthermore, this localization system does not require user's device to transmit any acoustic signal or RF messages. Such characteristics make the localization system energy efficient for the user's smartphone. In addition, it also preserve the privacy of the user.

We summarize our contribution of this paper as follow:

- Detailed study of the sound driver system for linux OS based smartphones (e.g Android, Maemo etc.).

- Leveraging both the acoustic interface and the WiFi interface of the smartphone to achieve high accuracy localization system.
- Detailed study of the sound propagation for LoS and NLoS scenarios.
- Implementing and evaluating the complete light-weight high-accuracy localization system using off-the-shelf smartphones.

To the best of our knowledge, Beacon-Beep is the first localization system that utilizes the acoustic system of the smartphone at kernel-level to improve the accuracy of the range estimation. In addition, we believe that the study of the sound signal propagation for both LoS and NLoS scenarios is a new contribution for localization systems.

The paper is structured as follows. We introduce the Beacon-Beep system with brief discussion about each component in the system in Section 2. In Sections 3-6, we describe in detail the different modules that construct client components of the Beacon-Beep system. We evaluate the performance of the system under different indoor/outdoor scenarios in Section 7. Then we discuss the related works in Section 8 and conclude with Section 9.

2. BEACON-BEEP SYSTEM

Figure 1(a) shows the different components of Beacon-Beep system. Similar to many localization systems, Beacon-Beep consists of two main components: infrastructure component that runs on infrastructure hardware, and client component that runs on the user's device (i.e. smartphone). In this paper, we refer to an infrastructure device running the Beacon-Beep infrastructure component by a *beacon device*. A beacon device periodically broadcasts a RF message (i.e. beacon frame), which we refer to as a *beacon*. In addition, a beacon device also generates an acoustic signal, *beep* following each broadcasted beacon message. In a practical scenario, a beacon device could be an Access Point (AP) with additional acoustic interface (i.e. speaker, mic and sound driver). A user's smartphone running Beacon-Beep client component (e.g., Beacon-Beep application) will capture the beacon messages and the corresponding beep signals from the surrounding beacon devices. Using the captured beacon messages and beep signals, Beacon-Beep application will infer the user's location.

In a typical usage scenario of Beacon-Beep, the Beacon-Beep application on user's smartphone will collect the beep signals from the surrounding beacon devices along their corresponding beacon messages. Each captured beep signal at user's smartphone will be classified either it is a Line-of-Sight (LoS) or a Non Line-of-Sight (NLoS) signal with respect to its corresponding beacon device. For the LoS signals, the application determines the relative ranges between the user's smartphone and the corresponding beacon devices. On the other hand, the application will predict a set of possible locations using acoustic fingerprint-database for the NLoS signals. Finally, the application combines the extracted information from the LoS and NLoS signals to estimate the actual user location.

The Beacon-Beep localization system has the following interesting features:

- The Beacon-Beep localization system exploits the multiple interfaces in the smartphones by utilizing both the RF interface and acoustic interface to design high-accurate localization scheme.
- The Beacon-Beep localization system does not require any central controlling component to coordinate between neighboring devices. In this system, user's smartphone calculate the location locally and no collaboration with neighboring devices is required.
- The Beacon-Beep application running by the users does not require any customized hardware. This enables any off-the-shelf smartphone to adopt and use the developed application.
- The Beacon-Beep system only requires one way transmission; the transmission of the beacon messages and beep signals by the beacon devices. This support the development of light-weight and energy efficient application for user's smartphones.
- The calculation of the user's location happen at the user's application. In addition, The user's application does not require to share any acoustic or WiFi information with any nearby device or access point. This enables to preserve and protect the user security and privacy.

In the following subsection, we provide a brief overview of the infrastructure component and the client component.

2.1 Infrastructure Component

The beacon device, which runs the infrastructure component of the Beacon-Beep system, periodically generates a RF beacon message followed by a beep signal. The beep signal is a sinusoidal signal with frequency above or equal to 18kHz. Such frequency is beyond the normal human hearing perception. In this work, we limit the frequency selection to be within the 18kHz-21kHz frequency range, which is perceptible to the most of the off-the-shelf smartphones [17]. To avoid interference in beep signal, similar to access point channel assignment , nearby beacon devices should select different frequencies for their beep signals in order to be able to detect different signal from different beacon devices in the same vicinity. From experiments, we found that if the frequency space between two adjacent beep signals is 250Hz, then it is sufficient to avoid the interference and detect the beep signals on client side using simple correlation technique. This frequency configuration enable us to have different 13 frequency values to be assigned concurrently to 13 overlapping access points, which is more than the typical number of overlapped access points in typical configuration. A centralized or distributed channel assignment scheme such as [18, 19, 20] could be used in beep signal assignment. Wi-Fi beacons generate by a beacon device will include information on the frequency value of the corresponding beep signal. In section 3.1, we describe more details about the beep signal detection technique. In the implementation, we use a smartphone (i.e. Nokia N900) with a speaker (i.e. Nokai MD-11) as a beacon device. In Section 5, we describe the implementation details of the beacon device using smartphones as well as the implementation details of the beep signal generation that follows each beacon message.

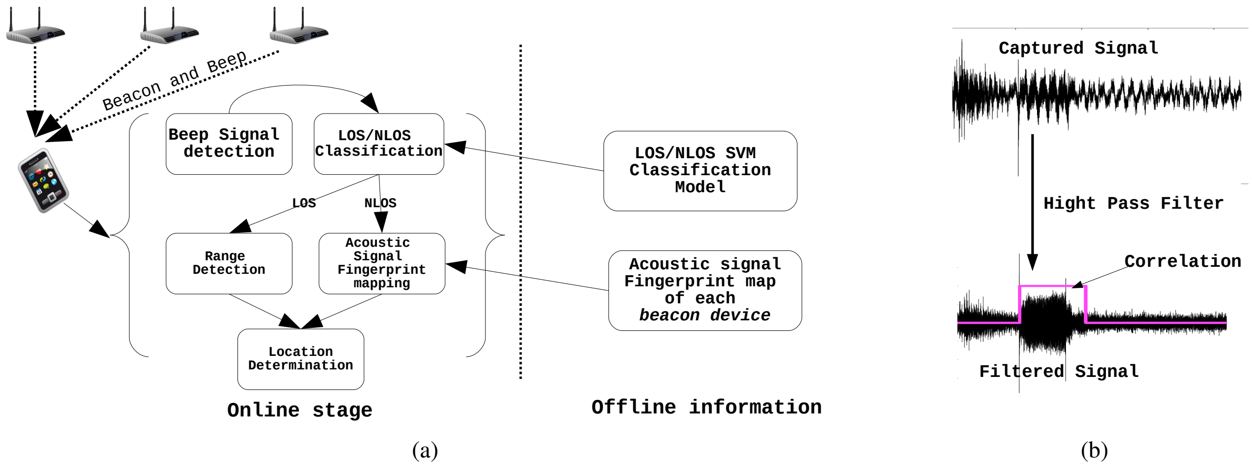

Figure 1: (a) Architecture overview of the different components of Beacon-Beep localization system (b) Beep signal detection technique

2.2 Client Component

In Beacon-Beep system, client component consists of four main modules described in the following:

- **Non Line-of-Sight (NLoS)/ Line-of-Sight (LoS) Detection module:** The NLoS/LoS detection module consists of a classifier and feature extraction components. We use a binary classification model through an offline training to classify and detect whether a received beep signal is LoS or NLoS. Once the Beacon-Beep application that is running on the user's smartphone receives a beep signal, it uses the feature extraction component to extracts the corresponding features from the beep signal to detect whether the beacon device corresponding to a received beacon is in the LoS or in the NLoS of the user's smartphone.

- **Acoustic-Fingerprint module:** In Beacon-Beep system, we build offline the acoustic fingerprint database using acoustic signal strength and the cepstrum coefficients [21]. Acoustic-fingerprint database is accessible, for example as a web service, to the Beacon-Beep application application on the user's smartphone. Given the location of a beacon device and the the layout of the indoor/outdoor surrounding, we determine the geometry of the NLoS area and the LoS area corresponding to that beacon device. Then, by collecting beep signals from that beacon device we build the corresponding acoustic-fingerprint database for the beacon device's NLoS area. This database is used, later, to predict the user's relative location to the corresponding beacon device.

- **Range Detection module:** For a LoS beep signal, range detection module is responsible to estimate the user's relative distance to the corresponding beacon device. This module uses the Time Difference of Arrival (TDOA) of the beacon message and the corresponding beep signal to estimate the relative distance between the user's smartphone and the beacon device.

- **Location Determination module:** The Location determination module estimates the user's actual location based on the outcomes of the range detection and the acoustic-fingerprint modules. If the range detection module provides the relative ranging of the user's smartphone for at least three different beacon devices (i.e., when the user is in LoS of at least 3 beacon devices), a triangulation technique is applied on these ranges to estimate the user location. When the outcome of the range detection module is less than three relative distances (i.e., the use is in the LoS of two or less beacon devices), we use the outcome of the acoustic-fingerprint module to provide a set of possible locations of the user for the NLoS beacon devices. Then, the location determination module uses both the relative ranges and the possible location set to calculate the user's location.

In the following, we describe each of the client component modules in more details.

3. NLOS/LOS DETECTION MODULE

In this section, we start with describing the scheme Beacon-Beep client uses to detect the beep signal from the captured account signal. Next, we analyze some common wavelet patterns of the beep signal under different LoS/NLoS conditions. In addition, we describe some anomalies that we have observed. Finally, we describe the binary classification model (LoS/NLoS) we apply on the beep signals.

3.1 Beep Signal Detection

Precise detection of the beep signal at the receiving side is crucial in the proposed system. Beacon-Beep system uses the beep signal detection scheme to: 1) classify whether the beep signal is from a LoS or a NLoS beacon device, 2) estimate the relative range between the user and the beacon devices for the LoS beep signals, and 3) estimate the possible locations of the user for the NLoS case.

Our detection scheme should satisfy the following two criteria: 1) precise identification of the first sample of the Beep sound, and ii) light weight implementation to cope with the user's smartphone capabilities. Figure 1(b) summarizes the steps followed in detecting the beep signal. Initially, a high pass filter over the sample data is applied to get rid of all ambient background noise. Then, we apply L-2 norm cross-correlation over the filtered signal. The correlation values above a minimum threshold belong to the beep signal.

We use a lightweight filtering method in time domain to satisfy the above second criteria. Given the input signal is $x[i]$ and output signal is $y[i]$, the first-order filter can be expressed as:

$$y[i] = (1-k)\cdot x[i] + k\cdot y[i-1]$$
$$y[i] = x[i] - y[i]$$
$$k = \exp - \frac{2\cdot\pi\cdot f_c}{f_s} \tag{1}$$

Based on the above equation, higher order high pass filter can be constructed. In our implementation, we use the 5th order high pass filter which embed 5 samples delay in detecting the starting event of the beep signal. Such delay can lead up to 4cm of error, which is negligible in our scenarios. In order to do the L-2 cross-correlation, we create a short sinusoidal signal of 25 samples with the same frequency as the beep signal. We select the length of the sinusoidal signal empirically from our previous observation. The correlation is done between the short sinusoidal signal with the filtered signal in a sliding window fashion with single sample increment. The index number of the captured sample with a correlation value above a threshold value indicates the beginning of the Beep sound. In the implementation, we set the threshold value to 0.9 which is enough to detect the direct path (i.e., LoS) signal. Although the first arrival of a NLoS signal might be weaker than the rest of the signal due to the multi-path effect, the threshold value of 0.9 is still enough to detect the starting of the beep signal for multi-path(i.e. NLoS) scenarios. In addition, the frequency of the multi-path signal usually get little distorted due to the phase shift. However, the frequency distance between any consecutive beep signals (i.e., 250Hz) takes care of the frequency offset due to the phase shift. Therefore, it is possible to detect two beep signals from two different beacon devices even if their arrival times overlap with each other.

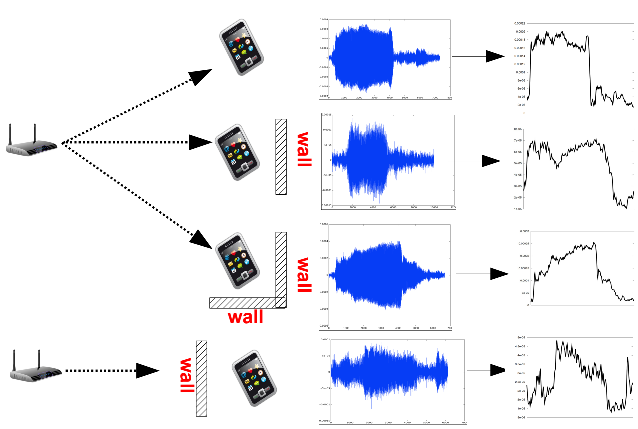

Figure 2: Detected beep signal under different LoS/NLoS conditions.

3.2 Beep Signal under LoS/NLoS Conditions

In this section, we describe the different observed patterns of the beep signal for different LoS and NLoS scenarios. Figure 2 shows typical arrangement configurations for both beacon device and the user's smartphone. In the first arrangement, the smartphone is in the LoS of the beacon device with no wall around the user's smartphone. In this configuration, the detected beep signal shows a flat with sharp start/ending pattern. For the second arrangement with a wall right behind the user's smartphone, the beep signal shows two spike; one from the direct path signal and the other from the reflected path signal of the wall. In the third case, where the user's smartphone is at the corner of two walls the beep signal shows a monotonous increment of the amplitude due to the aggregation of two reflected signal from the walls. All these patterns of the LoS beep signal have been verified by repeating the experiments over multiple times. Unlike LoS case, we observer that in NLoS case the beep signal has high variations pattern with multiple number of spikes. In addition, the amplitude of these multiple spikes also changes over time for NLoS cases. However, none of the patterns in the NLoS case matches with any of the LoS cases.

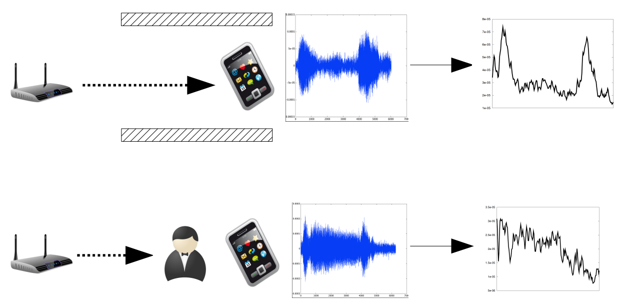

Figure 3: Detected beep signal under different anomaly conditions

Figure 3 shows some anomaly patterns for LoS/NLoS cases. For example, the first pattern is corresponding to placing the user's smartphone in a long narrow corridor. In this scenario, the beep signal shows two spikes with short distance between them. One pike is corresponding to the direct path signal while the other one is corresponding to the reflected path signal from the surrounding walls. Another anomaly scenario is when the user body blocks the direct path signal from the beacon device. In this scenario, the pattern of the beep signal is not consistent. We have observed several different patterns under this scenario. Based on these observed patterns, we describe in the next subsection the classification model we built to detect whether the received beep signal is corresponding to NLoS or LoS scenarios.

3.3 Classification of LoS/NLoS Signals

This section describe both the training and testing phase of the classification model we developed to detect whether the beep signal is corresponding to a LoS or a NLoS scenario. In training the classification model and select the corresponding features, we apply the following steps:

- The beep signal is a time series data with both positive and negative values. Therefore, we start with taking the absolute values of the time series data of the beep signal. Since the amplitude of the beep signal reduces along the distance, we normalize the beep signal's amplitude to minimize the effect of the distance. Let $R = \{r_1, r_2, ..., r_n\}$ is the normalized samples of the beep signal of length n. Note that the length of the beep signal is not fixed and can vary from one signal to another due to the multi-path effect. In R, we assume r_i are independent and identical distributions.
- Next, we find the initial values of the Gaussian mixer model, $\theta_j = \{\alpha_j, \mu_j, \sigma_j\}$ using the expectation maximization algorithm [22] for $j = 1, ..., m$, where m is the number of the gaussian models, α_j is the weight vector, μ_j is the mean vector, and σ_j is the variance vector. We use the normalized time series beep signal (R) for estimating the likelihood parameters of the Gaussian models.
- After estimating the gaussian models, we compute the following $R(i, j)$ which is a $n \times m$ matrix,

$$R(i,j) = \frac{1}{\sigma_j \sqrt{2\pi}} exp(\frac{1}{2}(\frac{r_i - \mu_j}{\sigma_j})^2) \tag{2}$$
$$i = 1, ..., n$$
$$j = 1, ..., m$$

  where n is the length of the beep signal samples.
- Next, we calculate the probability density function for the i-th normalized sample of the beep signal given the above gaussian model

$$P(r_i|\Theta) = \sum_{i=1}^{m} \alpha_j R(i,j) \tag{3}$$
$$i = 1, ..., n$$
$$j = 1, ..., m$$

  where Θ is the gaussian mixer model parameters.
- Finally, we reduce the variable length time series of the beep signal to a single size features using the following log-likelihood equation,

$$F = \sum_{i=1}^{n} \log P(r_i|\Theta) \tag{4}$$

The features values obtained from each input time series samples are used as input dataset for training and testing our SVM classifier model [23]. The SVM model basically performs binary classification to detect whether the received beep signal is corresponding to LoS or NLoS scenario. We experimented with different number of gaussian models to verify the performance of our SVM model as shown in Figure 4. We collected 200 LoS beep signals and 100 NLoS signals for our dataset. We used 10-fold cross validation over the dataset to calculate the overall performance (Figure 4).

4. ACOUSTIC-FINGERPRINT MODULE

The acoustic-fingerprint module consists of two components; a fingerprint database and a location mapping. The

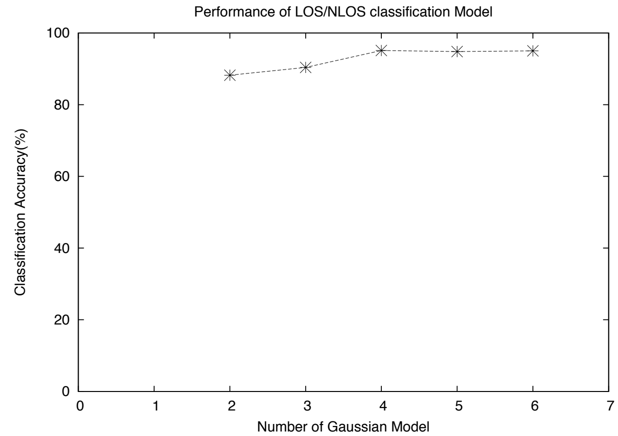

Figure 4: Classification Performance of our LoS/NLoS Detection Technique

fingerprint database is build offline based on the collected location annotated acoustic signal fingerprint. The location mapping is an online component that runs on user's smartphone. This component collects the signature pattern from the received beep signal and map to a possible set of locations using the fingerprint database .

4.1 Acoustic-Fingerprint Database

Fingerprint based localization scheme requires careful selection of fingerprints that shows sufficient spatial differences and minimum temporal variations. We use the acoustic signal strength and the cepstrum coefficients (1, 2 and 3) features for building the fingerprint database. We create a fingerprint-database for the corresponding NLoS area of each beacon device. For example, Figure 5 shows two beacon devices at two different corners of the building corridors (red and green circles) with the NLoS area corresponding to each beacon device along the corridors only (red and green boxes). We split each NLoS area to 25 blocks of size $1m \times 1m^2$ numbered from 0 to 24 with the smaller numbers corresponding to blocks closer to the beacon device. In the following section, we explain the selection of acoustic-fingerprint for the setup in figure 5.

4.2 Fingerprint Selection

The acoustic signal strength (power) of a *beep* signal is not always stationary. The change in the surrounding environment and/or the movement can change the value of the signal strength. In creating the fingerprint database, we collected number of samples at a particular location and use the mean and the variance values of the acoustic signal strengths of these samples as the fingerprint of that location. Figure 6 and 7 show the mean and the variance acoustic-fingerprint maps corresponding to the NLoS blocks in the figure 5. In these figures, both the mean and the variance values are normalized to (0,1) using the same range value (min-max normalization). Although a single sample of acoustic signal strength is not fixed, we verified through repeating the same experiment several time that the mean values are fixed over time. As shown in Figure 7, the variance values in the variance map are minimum. We see a continuous decrease of

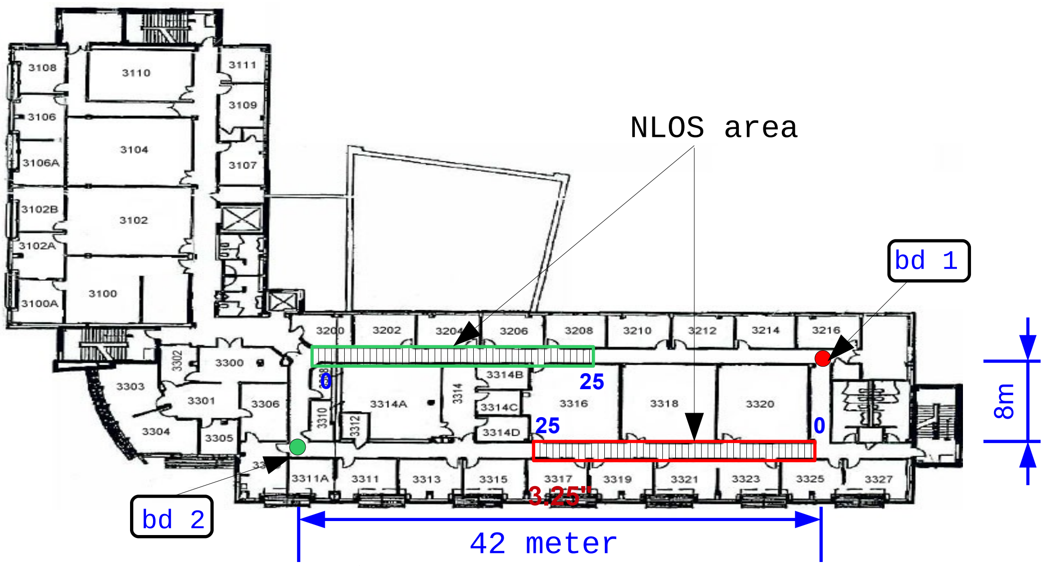

Figure 5: The 3rd floor map we used for the experiments

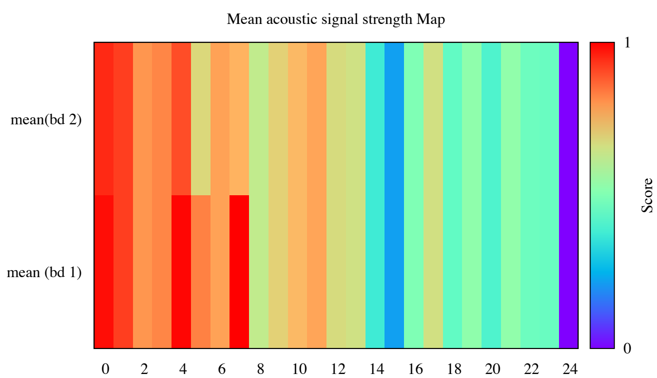

Figure 6: Acoustic Signal Strength Mean Values Map

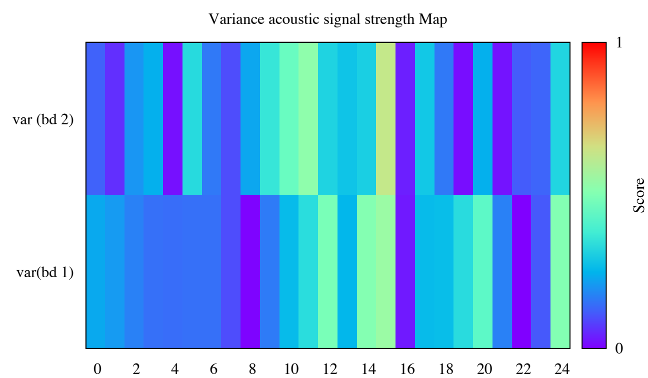

Figure 7: Acoustic Signal Strength Variance Map

the acoustic signal strength from near to far for the NLoS areas similar to the Wi-Fi signals. Unlike the mean map, the variance map suffers from a relatively high values in the middle of the map as shown for both beacon devices. This observation could be explained as the NLoS blocks closer to the beacon device receives strong acoustic signal strength of the beep signal and hence the effect of the surroundings are minimized due to the proximity. Similarly, for the far blocks, the beep signal has such low strength and the effect of the surrounding is not observable. On the other hand, the effect of surrounding has more impact on the signal strength of the beep signal that translates into high variance values for the NLoS blocks in the middle of the map.

We also use the cepstrum coefficient features of the beep signal to build the fingerprint database. The figure 8 shows the cepstrum coefficient fingerprint map for the beacon devices *bd_1*. In the figure 8, higher order cepstrum coefficients show less spatial variance. In contrast, the lower order cepstrum coefficients show changes over space (e.g., cepstrum coefficients 1, 2, and 3 in figure 8). The value of the cepstrum coefficient reflects the variation pattern of the signal over time. Each cepstrum coefficient corresponds to a different fluctuating component of the signal at time domain. The 0-order cepstrum coefficient represents the DC component of the signal which is the total power of the signal. Thus we didn't use the 0th-order cepstrum coefficient . On the other hand, we found higher order cepstrum coefficients are less effective fingerprints. We select the coefficient that are more consistent and uniquely distinguishable over space. From the fingerprint map 8, we select the cepstrum coefficient 1, 2, and 3 for the fingerprint database.

4.3 Location mapping

The LoS/NLoS detection module sends the received beep

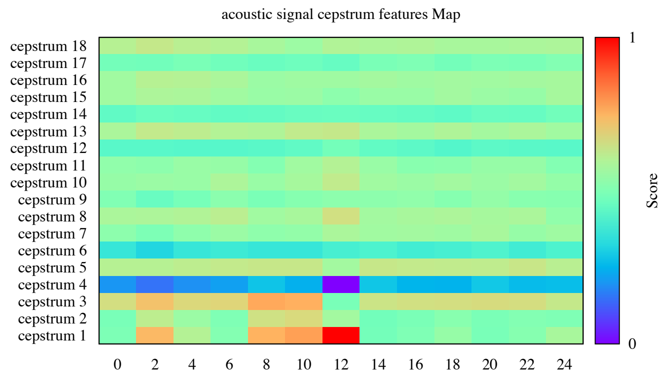

Figure 8: Acoustic Signal Cepstrum map

signal to the location mapping component if it is classified as NLoS signal. By knowing the used acoustic frequency by the beacon device, location mapping component determines the beacon device corresponding to the beep signal. Then, the location mapping component extract the features (acoustic signal strength and cepstrum coefficients 1, 2, and 3) from the beep signal. Finally, we apply the following steps to map the features with the fingerprint database to find the possible set of locations.

- Given a NLoS area, we have m location annotated acoustic signal fingerprint s for a beacon device. Then the fingerprint database will have the following information for the m locations,

$$\mu_i,\ i = 1, 2, ..., m.$$
$$\sigma_i,\ i = 1, 2, ..., m$$
$$C_i = \{c_{i,1}, c_{i,2}, c_{i,3}\},\ i = 1, 2, ..., m \quad (5)$$

  where μ_i and σ_i are the mean and the variance of the signal strength at location i respectively. C_i is the set of cepstrum coefficient (1,2 and 3) for the location i.

- Let the features extracted from the beep signal be:

$$p,\ S = \{s_1, s_2, s_3\} \quad (6)$$

  where p is the strength of the signal and S is the set of cepstrum coefficients for the beep signal.

- Then, we calculate the following metrics:

$$X_i = \frac{1}{\sigma_i\sqrt{2\pi}} exp(\frac{1}{2}(\frac{p - \mu_i}{\sigma_i})^2)$$
$$d_i = \sum_{j=1}^{3} |s_j - c_{i,j}|^2$$
$$q_i = \frac{X_i}{d_I}$$
$$i = 1, 2, ..., m \quad (7)$$

- Finally, we find a set of possible locations by selecting first couple of maximum values from the list $q_i,\ i = 1, 2, ..., m$, where the values of the i map to the possible set of locations.

5. RANGE DETECTION MODULE

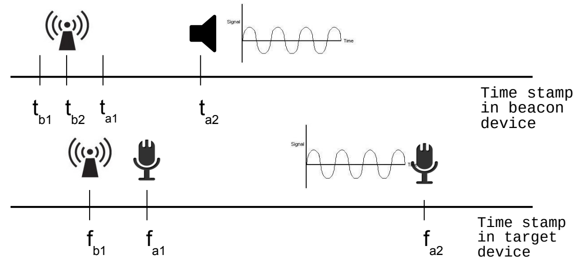

Figure 9: System overview of Range Detection Module.

In this section we describe in details the of range detection module. We start by describing the module overview, followed by the challenges and how we address those challenges in implementing the module. In measuring the relative ranging distance, beacon device is the frame of reference and the user's smartphone measure the range to the beacon device.The user's smartphone record the timestamps of the reception of both the RF beacon message and the acoustic beep signals received from the beacon devices. These timestamps help the user's device to infer the relative distance to the corresponding beacon device.

5.1 System overview

The range detection module is based on the Time Difference of Arrival(TDOA) technique that utilizes the relative velocity of two different signals; RF and acoustic. Figure 9 gives an overview of the range estimation mechanism and shows the time-line of the different event's timestamps for both the beacon device and the user's smartphone. These timestamps are:

- t_{b1}, Time when the beacon device puts the Wi-Fi beacon message into the transmission buffer of the firmware.

- t_{b2}, Time when the last bit of the beacon message is transmitted from the beacon device.

- t_{a1}, Time when audio driver starts writing audio frames into the hardware buffer.

- t_{a2}, Time when speaker starts to generate the beep sound from the beacon device.

- f_{b1}, Time when the user's smartphone receives the last bit of the RF beacon message.

- f_{a1}, Time when microphone of the user's smartphone is turned on to capture the audio data in the audio driver buffer.

- f_{a1}, Time when the user's smartphone detects the starting of the beep sound from the captured data.

In this range detection module, the user's smartphone only receives the RF beacon message and beep signal. Such flexibility enables the user's smartphone to calculate the relative ranging distance from beacon device locally. As we will show later, this range estimation mechanism does not require any time synchronization between the beacon device and the user's smartphone.

A typical Time-of-Arrival system uses the propagation speed of the signal to infer the distance between the transmitter and the receiver. The signal with high speed needs to have high precision in determining the travel time from sender to receiver. Since the sound has lower propagation speed compared to RF, acoustic signal has lower relative ranging error corresponding to small timing error. For example, a millisecond error of TOA estimation could result up to 30 centimeter of error in estimating the relative range of an acoustic signal. In order to limit the relative ranging error to few centimeter, it is enough to maintain the time precision in millisecond level in the range detection module. In figure 9, all event's timestamp is considered to be in millisecond precision.

Given the high propagation speed of the Wi-Fi signal and the small length of the Wi-Fi beacon, it takes less then a millisecond to transfer a beacon messgae from a beacon device to a user's smartphone. Even though the value of t_{b2} and f_{b1} might be different in two devices, we approximate both timestamps t_{b2} and f_{b1} in representing the same event on the two different timelines. Given the speed of the sound in air is s_a and the distance between the *beacon device* and user's smartphone is, D then we can write the following equation from figure 9

$$\begin{aligned} D &= s_a.(t_{a2} - f_{a2}) \\ &= s_a.(t_{a2} - f_{a2} + t_{b2} - t_{b2}) \\ &= s_a.((t_{a2} - t_{b2}) - (f_{a2} - f_{b1})) \\ &= s_a.(\Delta t_{ab} - \Delta f_{ab}) \end{aligned}$$

In Equation (8), Δt_{ab} represents the time difference between the last bit of the beacon message transmitted and the speaker started to generate the beep signal from the beacon device. Similarly, Δf_{ab} represents the time difference between the last bit of the beacon message received and the start event of receiving the beep signal at the user's smartphone. Both Δf_{ab} and Δt_{ab} values are the local time difference at the user's smartphone and the beacon device respectively. Therefore, it is worth to pint that no synchronization is required between the two device in the calculation of Δt_{ab} and Δf_{ab} values. Moreover, In range detection module, as we will show later, we manage to eliminate the uncertainty in Δt_{ab}. Therefore, the user's smartphone does not require any information from the beacon device and the range detection module doesn't require any collaboration between the two devices.

5.2 Uncertainty Challenge

In TDoA, accuracy of the ranging method highly depends on the precision and accuracy of measuring the arrival time of two different signals. Typically, TDOA based approach requires to track the timestamps of the different transmission and reception signals. As a consequence, such approach requires time synchronization between the receiver and the transmitter. To address this synchronization requirements, a periodic broadcast of special message is required by a centralized node [24, 25]. However, this approach introduces number of uncertainties that has been described in Fikret et al. [26]. These uncertainties reduce the chance of maintaining precise time synchronization between different devices. Recently, to overcome the requirement of time synchronization number of relative ranging schemes have been proposed [13, 1, 12, 2, 11]. Such schemes utilize the time difference of two local event's timestamp rather then using a single local even's timestamp value to measure the relative range. The range detection module, as shown in Equation (8), utilizes the same trick to tackle the time synchronization problem.

Authors In [13] highlighted some of the uncertainties in acoustic based high accuracy relative ranging estimation. For example, one of the major uncertainties is the high variation delay between the actual emission of acoustic signal by the speaker and the signal transmission command generated by the application. As we described earlier, in the implementation of our Beacon-Beep system, we implement the beacon devices using smartphones. Therefore, it is important in our system to understand and eliminate the uncertainties in transmitting and receiving beep signals using smartphones. Figure 10 plots the CDF of the delay associated with the transmission of the acoustic signal (solid line) from Nokia N900 smartphone for 1000 samples. It is clear from the figure that the delay varies significantly between 2ms and 6ms, which lead to few meters of error in measuring the relative range module.

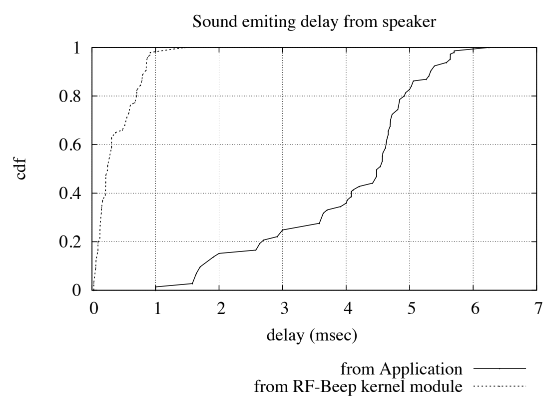

Figure 10: Sound sending delay uncertainty between normal application sending sound signal and our range detection module sending the sound signal. We have run 1000 times get the delay values for both normal application and our range detection module. In both cases we send 4400 samples of sound data for 44100Hz sampling rate

By analyzing the smartphone's sound driver actions in details as we will describe in the following subsection, we identified that the total delay in acoustic signal transmission consists of 3 main delays that are correspond to the following three actions: i) Power up the playback stream, ii) Data transfer from application to sound driver, and iii) the DMA data transfer from sound driver to actual sound hardware. However, Implementing the range detection module at the sound driver level provides us the flexibility in controlling the delay of powering up the playback stream. Moreover, executing the transmission of the acoustic signal at the driver level help us to get rid off the delay of data transfer from application to sound driver. This leaves our system with only one source of uncertainty corresponding to the DMA data transfer from sound driver to actual hardware. However, the impact of this uncertainty is less then 1 millisecond as shown in Figure 10 (dotted line). In order to guarantee that the user's smartphone captures the beep signal from the very beginning, we add a fixed delay before sending the beep signal.

At the receiving side, user's smartphone has two uncertainties in receiving the *beep* signal. i) Power up the capture stream, and ii) Detect the starting event of the *beep* signal. Figure 11 plots the delay of powering up the capture stream of the *target device*. As shown, the delay of powering up the capture stream is between 2-2.5msec maximum. Note that, this delay measurement allows us to determine how much delay (Δt_{ab}) we should consider when sending the *beep* signal from the *beacon device*. Both the power up delay and the delay of determining the starting event of sound are exclusive, so we can write the Δf_{ab} as:

$$\Delta f_{ab} = \mu + \frac{n_{ab}}{f_s}. \tag{8}$$

where μ is the delay to power up the capture stream, n_{ab} is the starting sampling number of the captured beep signal;, and f_s is the sampling frequency of capturing event. Note that, $\frac{n_{ab}}{f_s}$ is the delay of receiving the beep signal. Detecting the n_{ab} in the captured sound data is a challenging task. In order to have high precision in measuring the relative ranging, it is critical to precisely detect the n_{ab} value. In Section 3.1, we described the approach of detecting beep signal in more details.

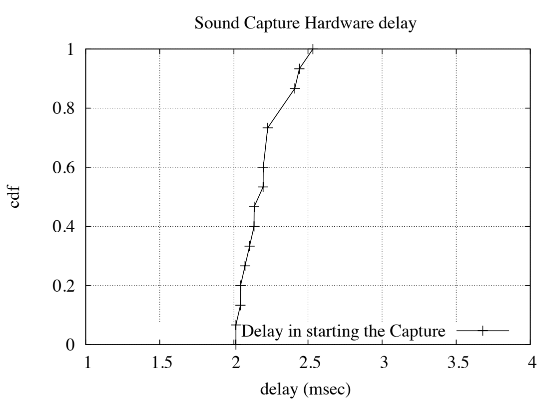

Figure 11: Audio capture starting event delay

5.3 Operation of Sound Driver

In the range detection module, it is important to have accurate timestamp information for both playing and recording sounds. Such requirement motivates us to understand more details about the operation of the linux sound driver in smartphones. This section describes about the sound driver of the smartphones that are linux based(e.g. Android phones, N900, Maemo 5 phone etc.). In this section, we describe how the sound samples transfer from the user-space to the actual hardware through the sound driver, while playing a sound. In addition, we explain how sound samples transfer from the actual hardware to the user-space through the sound driver, while capturing or recording the sound.

In smartphones, the sound operation consists of two main components, i) the kernel-level sound driver and ii) the sound library. User applications can only access the sound library to communicate with sound driver for playing and recording sounds. The sound driver within the kernel has three layers: 1) the low-level layer that is responsible for accessing the hardware and is implemented as callback functions, 2) the middle-level layer that supports the common routines for different sound hardware components, and 3) the top-level layer that is the entry point for the sound library.

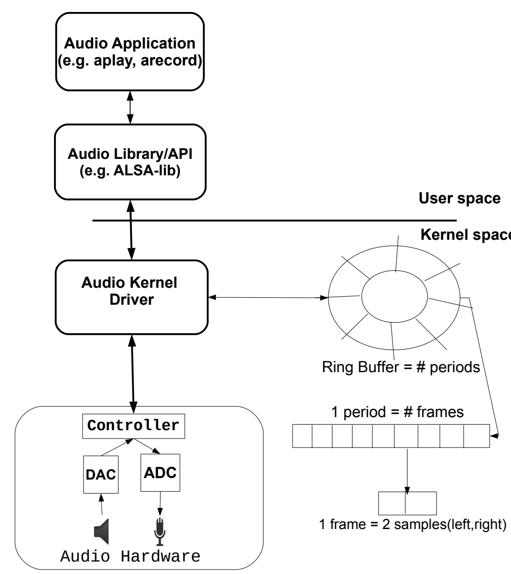

Figure 12: Operation flow of playing and capturing sounds in smartphones.

The sound library typically transfers the sound samples from the user-space application buffer to the DMA buffer within the kernel-space. This DMA buffer is also shared between the sound driver and the sound hardware. The DMA buffer implements a "ring buffer" in which when it reaches the end of the buffer, it restarts automatically from the beginning. The DMA "ring buffer" has two pointers, the application pointer and the hardware pointer. In major sound hardwares, the DMA buffer is divided into several "epoch"s. The pointers (hardware and application) of the DMA buffer via, usually, the invocation of interrupts get updated by the end of each "epoch". The DMA buffer size is equal to the number of epochs multiplied by the "epoch size". The "epoch size" is the number of frames that is corresponding to one

”epoch”. The frame is the smallest unit of sound samples.

Each of the playback stream (i.e. while playing sound) and the capture stream (i.e. while recording sound) has its own DMA buffer. In playing a sound, the application writes the sound samples to the DMA buffer within the kernel-space. After writing one epoch size of samples, sound driver initiates an interrupt operation to start the DMA data transfer from the DMA buffer to the hardware buffer. Following the completion of the interrupt, the sound driver updates the pointers of the DMA buffer thru the corresponding call-back function. By recording the timestamp of this update operation, we capture the timestamp actual transmission of the sound signal event. For the sound recording operation, the hardware interrupts the sound driver to update the buffer pointers after each filling of the DMA buffer with one epoch size of sound samples. By recording the timestamp of this interrupt, we will be able to predict the actual time of receiving the sound signal.

5.4 Temperature Estimation

In the range detection module, we use the sound speed to calculate the relative range. The actual sound speed depends highly on the environmental factors such as temperature, relative humidity and atmospheric pressure. Among these environmental factors, relative humidity and atmospheric pressure has the least impact on sound speed in the air [3]. In contrast, the sound speed is very sensitive to the changes in the temperature. For example, a change of one Celsius degree in the temperature could cause an approximately of 18% change in the sound speed. This sound speed sensitivity forces us to measure the air temperature prior to ranging estimation . In Beacon-Beep system user’s smartphone estimates the temperature of the atmospheric air before calculating its location.

Figure 13: Position of the speaker and microphone

Now-a-days, smartphones come with internal temperature sensor to protect the internal circuit for getting over heated. Such sensor are not suitable to measure the air temperature of the surrounding environment. To over come this problem, we use the acoustic hardware of the smartphone to measure the temperature of the surrounding air. In doing this, we identify the smartphone’s speaker and microphone with the farthest distance in between. For example, Figure 13 shows Nokia N900 smartphone that has two speakers; one at the top(right) and the other one at the bottom(left). The microphone is also located near the bottom of the phone. As shown in the figure, the top speaker and the bottom mic has the farthest distance, which is 10cm. We turn on both the speaker and the microphone of the smartphone. Later we generate a sinusoidal tone from the top(right) speaker and capture that sound by the microphone. We utilize the sound driver to control the tone generation from the top(right) speaker. As shown in Figure 14, we record both the timestamps of the actual transmission of the tone from the speaker, and the reception of that tone at the microphone. In the experiments, we use the following model [27] to calculate the air temperature:

$$\theta = \frac{\frac{d}{t_b - t_a} - 331.3}{0.6} \tag{9}$$

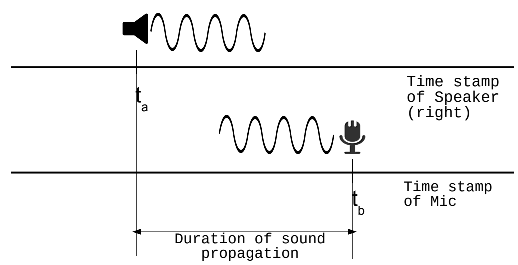

Figure 14: Timeline of sound propagation between speaker and microphone

where θ is the air temperature in Celsius, d is the distance between the speaker and the mic (i.e 10cm from Figure 13), t_a is the timestamp (in microsecond) of tone transmission from speaker, and t_b is the timestamp (in microsecond) of the tone reception by the microphones. Based on the above equation and with t_a, t_b, d measurements, we estimate the temperature at different indoor/outdoor environments. For each environment, we have collected 20 samples to measure the temperature. Figure 15 shows the statistics of our estimated temperature measurement using Equation (9). From the figure, the estimated temperature usually is within 4-5° Celsius of the actual temprature, which has minor effects on the sound propagation speed in the air.

6. LOCATION DETERMINATION MODULE

The location determination module is responsible to estimate the user’s final location based on both the range estimation for the LoS beep signals, and the location mapping using fingerprint database for the NLoS beep signals. If the user is in the LoS of at least three beacon devices, location determination module applies the triangulation technique to estimate the user’s actual location. However, in scenarios where the users is in the LoS of two or less beacon devices, location determination module infer the user location using the following steps:

- Using the acoustic-fingerprint module, we select the

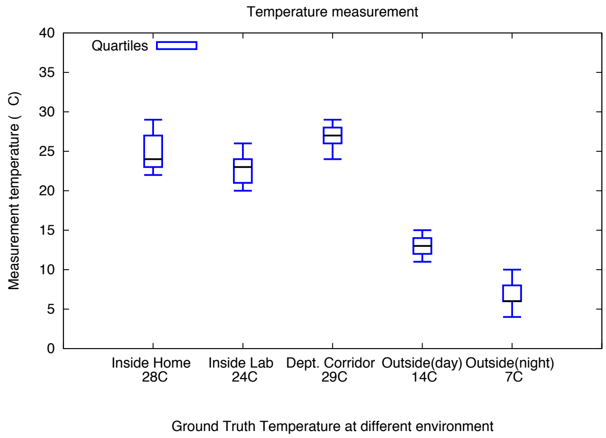

Figure 15: Estimated Temperature over different indoor/outdoor environment

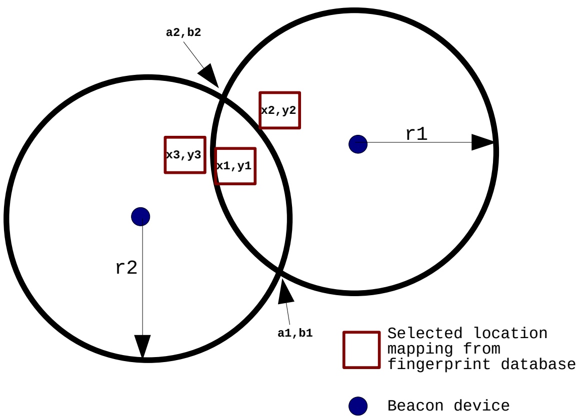

Figure 16: Location determination mechanism for LoS-NLoS case

best f location blocks from the acoustic-fingerprint database for each of available m NLoS beep signals as described in section 4. Figure 16 shows example of such location blocks (red boxes), where each location block is represented by its central coordinate (x, y). In our implementation, the size of each location block is $1m \times 1m$. The union set of those location blocks is defined as:

$$L_i = \{l_{i_1}, l_{i_2}, ..., l_{i_f}\}$$
$$LB = L_1 \bigcup L_2 ... \bigcup L_m$$
$$i = 1, 2, ..., m. \tag{10}$$

- Using the range detection module, from the available n (i.e., $n < 3$) LoS beep signals we estimate the relatives ranges $r_1, r_2, ..., r_n$ between the users's location and the corresponding beacon devices.

- For the scenarios where the number of LoS signals is two (i.e., $n = 2$) and given we have N number of location blocks in the set LB, we apply the following calculation to infer the user's location d_{min}:

$$l_i \equiv (x_i, y_i) \quad i = 1, 2, .., N$$
$$D_{d_1} = \sum_{i=1}^{N} \sqrt{(x_i - a1)^2 + (y_i - b1)^2}$$
$$D_{d_2} = \sum_{i=1}^{N} \sqrt{(x_i - a2)^2 + (y_i - b2)^2}$$
$$d_{min} = \min_{d=\{d_1, d_2\}} D_d \tag{11}$$

where $(a1, b1)$ and $(a2, b2)$ the intersection points of the two circles centered around the two LoS beacon devices with radiuses corresponding to the calculated ranges (i.e. r_1 and r_2). Figure 16 shows an example for this scenario with two LoS beacon devices and their corresponding ranges.

- For the scenarios we have a single LoS (i.e., $n = 1$), we use the following calculations to infer user's final location d_{min} by trying to locate the closest location block to the circle's perimeter centered around the beacon device corresponding to LoS beep signal with radius r_1:

$$l_i \equiv (x_i, y_i) \quad i = 1, 2, .., N$$
$$d_{min} = \min_{l_i} |r_1 - \sqrt{(a - x_i)^2 + (b - y_i)^2}|$$

where a, b are the center coordinate of the beacon devicel.

7. PERFORMANCE EVALUATION

7.1 Experiment Equipment

In the experiments we use the Nokia N900 smartphone as a beacon device. In addition, we add a mobile speaker Nokia MD-11 with the N900 smartphone to generate the *beep* signal. In the experiments, we were able to detect the beep signal up to 35m (on average) from the beacon device for both indoor and outdoor environments. This acoustic range fits with the 30m typical Wi-Fi ranges in the indoor environments [7]. Since the Wi-Fi will have larger range in outdoor, we can also extended the range of the beep signal by using more powerful speaker. In our experiments, we use the same speaker Nokia MD-11 with the same transmission power of 3000mW for both indoor and outdoor environments. We use the same smartphone model for implementing the Beacon-Beep client application. N900 runs Maemo 5 linux based OS powered by TI OMAP3 processor which can support up to 1 GHz [28]. The phone has 256MB of high performance RAM with 1GB of VM support, two speakers laid out at the top and the bottom surface of the phone, and a single microphone located at the bottom of the front surface (figure 13). The audio features of the device is provided by the TI TWL4030 chip supported by ALSA SoC driver. The Wi-Fi chipset of the N900 phone is TI WL1251 supported by wl12xx driver. Beacon-Beep application requires some modifications of the wl12xx and the ALSA SoC drivers which are both part of the linux kernel open source code.

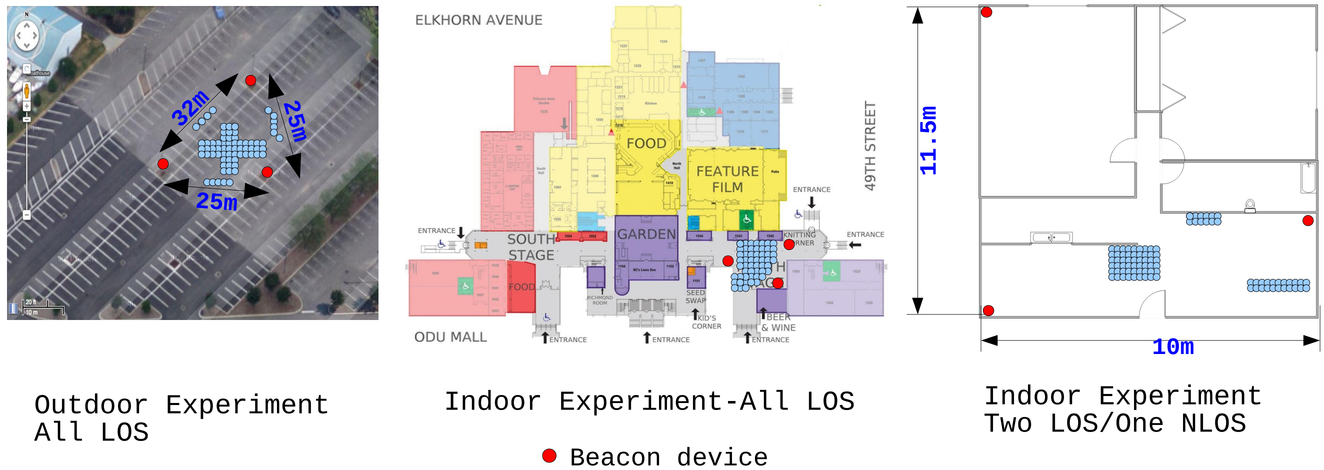

Figure 17: Outdoor and Indoor layout of our evaluation experiments

7.2 Experiment Scenarios

We evaluate the Beacon-Beep localization system under the following different scenarios:

- **Outdoor Environment-Three LoS**: This experiment is taken place in a open space parking lot. Figure 17 (left) shows the location of the three *beacon devices*.
- **Indoor Environment-Three LoS**: In this scenario, we conducted our experiment inside the WebCenter building in our campus. Figure 17 (center) shows the location of the three *beacon devices*.
- **Indoor Environment-Two LoS/One NLoS**: In this indoor scenario, we ran our experiment inside a two bedroom apartment. Figure 17 (right) shows the location of the three *beacon devices*. In this case, we place one *beacon device* in NLoS and two in LoS to the user.
- **Indoor Environment-One LoS/One NLoS**: Figure 5 shows the setup for this scenario with two beacon devices. In this experiment, we evaluate our system for the locations with one LoS and one NLoS beacon devices.

For each experiment setup, we have repeated the location estimation for different 500 samples from 50 distinguish locations.

7.3 Experiment Results

Outdoor Environment-Three LoS: In Figure 17 (left), blue circles shows the locations of the collected samples. Plot 18(a) shows the distribution of the distance error in cm for this experiment setup. The plot shows that the error in location estimation is less then 1meter for more than 90% of the samples collected in this outdoor experiment.

Indoor Environment-Three LoS: Figure 17 (center), shows the approximate locations of the collected samples (i.e., blue circles). This experiment was done in a public place with a lot of surrounding students. In addition, the place has several furnitures such as tables, chairs and desktop PCs. The distance ranges between the beacon devices in the range of 20-22m. Plot 18(b) shows the error distribution of the estimated location with an error less then 1.5 meter for more than 90% of the time.

Indoor Environment-Two LoS/One NLoS: In this experiment, we placed one beacon device (i.e., smartphone) in the NLoS (i.e., shown at the top left corner of Figure 17 (right)) while the other two beacon devices in the LoS of the collected samples that are show as blue circle in Figure 17 (right). In this experiment, we select the highest three scored possible locations (based on Equation (7)) from the acoustic-fingerprint database corresponding to the NLoS beacon device. Plot 18(c) shows the distributions of the distance error with error less then 1meter for more than 90% of the samples.

Indoor Environment-One LoS/One NLoS: In this experiment, we have collected the location samples from the green and the red box in figure 5. The plot 19 shows the distributions of the distance error with error less then 1.1 meter for more than 95% of the samples.

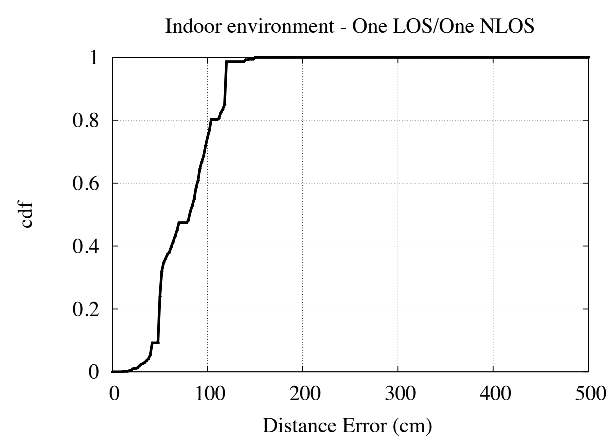

Figure 19: Distribution of the indoor experiment localization error for One LoS and One NLoS

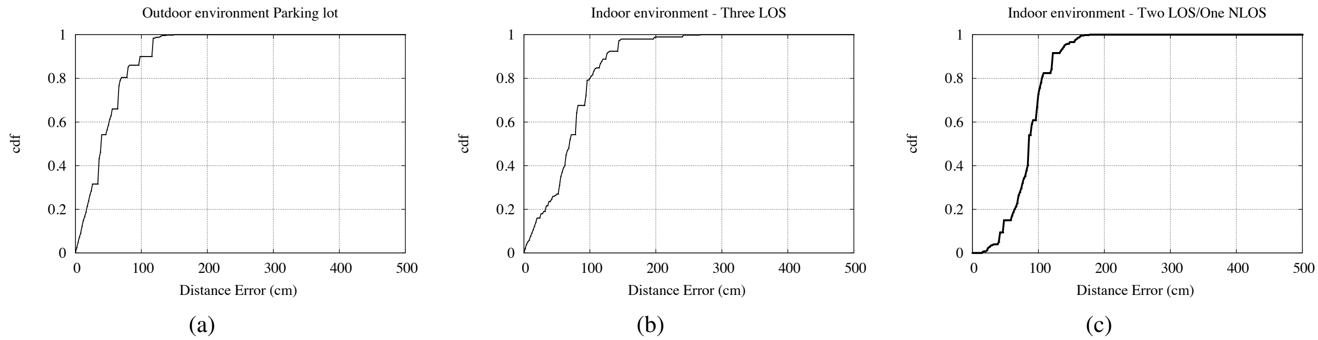

Figure 18: (a) Distribution of the outdoor experiment localization error for Three LoS (b) Distribution of the indoor experiment localization error for Three LoS (c) Distribution of the indoor experiment localization error for Two LoS and One NLoS

8. RELATED WORK

Most of the localization research works have been based on Radio Frequency (RF)-based techniques that leverages signal strength of RF signal from different nearby RF sources or infrastructures (e.g., WiFi Access point, Cellular Tower). Such RF based approaches with sophisticated localization algorithm can achieve reasonable accuracy with error range of 6-8m. Recent work shows that existence of the same signature or fingerprint of RF signal at different distinct location prohibits to have high accuracy in indoor localization system [15, 4, 14, 29]. Recently, researchers are combining multiple modalities such as sound with the WiFi to achieve high accuracy localization system [1, 2, 30, 3]. For example, the localization schemes [1, 2, 12, 11], utilize the acoustic based ranging [13] scheme and combine it with the RF at the application layer. In this section we highlight some of the recent and most relevant location based systems that use both sound and RF for localization.

The BAT [30] system is an indoor localization system that utilizes the time-of-flight of ultrasound signal. BAT system requires tightly controlled and centralized architecture. Moreover, the receiver in the BAT system needs to be synchronized using the broadcast message from RF base station. However, such technique has number of uncertainties that has been discussed in Fikret et al [26]. Beside, BAT system requires their own customized hardwares to implement. Cricket [3] is a indoor localization system that utilizes the combination of RF and ultrasound to determine the distance of a targeted device. It uses concurrent transmissions of radio and ultrasound signals and their corresponding difference of arrival times to the target device to infer the distances. Although the cricket uses two different signals similar to our Beacon-Beep system, but it requires customized hardware for time stamping and to ensure concurrent transmissions of both RF and acoustic signal. Moreover, unlike our Beacon-Beep system, the Cricket system targeted to achieve a room-level granularity of accuracy in determining the location of the target devices.

In most recent work [1], author have used both the acoustic signal and the WiFi (RSSI) to achieve high accuracy. For example, In Centaur author have used RF signal strength and Acoustic based ranging to achieve high accuracy indoor localization system. Even though Centaur can achieve high accuracy but the system requires a centralized control system to schedule communication between the nearby devices. As a result, implementing such system requires to maintain a backend server or controlling system. It also requires the user's device to communicate with the neighboring user's devices, which make the user to sacrifice their privacy. Similarly, [2] also requires a centralized controlling system to communicate between nearby devices. In [10],the proposed localization system only uses the acoustic features of the background sound for indoor localization. The background sound of a room is not a robust features for indoor localization purpose. Moreover this systems only targeted to achieve room level location accuracy. In addition, some localization system uses multiple modalities of the smartphone including sound to determine logical location of user[9].

9. CONCLUSION

This paper shows how the sound system and the RF interface in the off-the-shelf smartphone can be utilized to develop a light weigh high accuracy localization system. In this paper, we leverages the unique characteristics of the acoustic signal to address both the LoS and NLoS scenarios. The proposed localization system Beacon-Beep does require any central coordination module for the devices. Moreover, our system does not require any custom made hardware for the user's devices. Beacon-Beep localization is also energy efficient for the user's device. In addition, the localization system preserve the user privacy during location estimation. We evaluate our system under different indoor/outdoor environment. Throughout the experiment, we were able to achieve location estimation error less or equal to 1.5m for more than 95% of the time.

10. REFERENCES

[1] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: locating devices in an office environment." in *Mobicom '12: Proceedings of the 18th annual international conference on Mobile computing and networking*, new York, NY, USA: ACM Press, 2012, pp. 281-292.

[2] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones." in *MobiCom 2012: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, istanbul, Turkey, August 2012.

[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system.” in *MobiCom 00: Proceedings of the 6th annual international conference on Mobile computing and networking*, new York, NY, USA, 2000.
[4] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, “Walrus: wireless acoustic location with room-level resolution using ultrasound,” in *MobiSys 05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, new York, NY, USA: ACM Press, 2005, pp. 191203.
[5] A. Matic, A. Popleteev, V. Osmani, and O. Mayora-Ibarra, “Fm radio for indoor localization with spontaneous recalibration.” *Pervasive Mob. Comput.*, vol. 6, December 2010.
[6] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka., “Spot localization using phy layer information.” in *In MobiSys 12, 2012.*
[7] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, “Fm-based indoor localization.” in *In Proceedings of The 10th International Conference on Mobile Systems, Applications and Services, (MobiSys 12)*, lake District, UK, June 25-29, 2012.
[8] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. W. I, “Indoor location sensing using geo-magnetism.” in *In MobiSys 11, 2011.*
[9] M. Azizyan, I. Constandache, and R. R. Choudhury, “Surroundsense: mobile phone localization via ambience fingerprinting.” in *MOBICOM 2009*, pages : 261-272.
[10] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum.” in *MobiSys 2011*, pages : 155-168.
[11] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda., “Swordfight: Enabling a new class of phone-to-phone action games on commodity phones,” in *MobiSys’12: Proceedings of ACM MobiSys*, low Wood Bay, Lake District, UK, 2012.
[12] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, “On the feasibility of real-time phone-to-phone 3d localization.” in *SenSys 11: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, new York, NY, USA: ACM, 2011, pp. 190203.
[13] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “Beepbeep: A high accuracy acoustic ranging system using cots mobile devices.” sydney, Australia, ACM SenSys 2007, Nov 2007.
[14] M. Youssef and A. Agrawala, “The horus wlan location determination system.” in *MobiSys 05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, new York, NY, USA: ACM Press, 2005, pp. 205218.
[15] P. Bahl and V. N. Padmanabhan, “Radar: An in-building rf-based user location and tracking system,” in *INFOCOM 00: the 19th Annual IEEE Conference on. Computer Communications*, tel-Aviv, Israel: IEEE Infocom, March 2000.
[16] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, “Pinpoint: An asynchronous time-based location determination system.” in *MobiSys 06: Proceedings of the 4th international conference on Mobile systems, applications and services*, new York, NY, USA, 2006.
[17] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, “Detecting driver phone use leveraging car speakers.” in *International conference on Mobile computing and networking, 2011.*
[18] C.-F. Wong, S.-H. G. Chan, and J. Chen, “Paca: Peer-assisted channel assignment for home wireless lans,” in *Proceedings of the Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, 27 November - 1 December 2006*. IEEE, 2006.
[19] X. Yue, C.-F. Wong, and S.-H. G. Chan, “A distributed channel assignment algorithm for uncoordinated wlans,” in *Proceedings of the 7th IEEE conference on Consumer communications and networking conference*, 2010.
[20] C. Schurgers, G. Kulkarni, and M. B. Srivastava, “Distributed on-demand address assignment in wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 10, Oct. 2002.
[21] Wiki, “Mel-frequency cepstrum,” http://en.wikipedia.org/wiki/Mel-frequency_cepstrum.
[22] Y. Athavale, S. Krishnan, and A. Guergachi, “Pattern classification of signals using fisher kernels,” *Hindawi Publishing Corporation Mathematical Problems in Engineering*, 2012.
[23] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
[24] S. Ganeriwal, R. Kumar, and M. B. Srivastava., “Timing-sync protocol for sensor networks.” in *In SenSys,*, pages 138149, 2003.
[25] J. Elson, L. Girod, and D. Estrin., “Fine-grained network time synchronization using reference broadcasts.” in *OSDI, 2002.*
[26] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey.” in *IEEE Network*, pages 4550,2004.
[27] Wiki, “Speed of sound,” http://en.wikipedia.org/wiki/Speedofsound.
[28] ——, “Nokia n900,” http://wiki.maemo.org/Nokia_N900.
[29] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, “Ariadne: a dynamic indoor signal map construction and localization system,” in *MobiSys06: Proceedings of the 4th international conference on Mobile systems, applications and services*, new York, NY, USA, 2006.
[30] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, “The anatomy of a context-aware application.” in *MobiCom 99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, new York, NY, USA, 1999.