RF-Beep: A Light Ranging Scheme for Smart Devices

Mostafa Uddin
Department of Computer Science
Old Dominion University, USA
email: muddin@cs.odu.edu

Tamer Nadeem
Department of Computer Science
Old Dominion University, USA
email: nadeem@cs.odu.edu

***Abstract*—In this paper, we design, implement and evaluate *RF-Beep* - a high-accuracy, one-way sensing, energy efficient and light-weight ranging scheme for smart devices. *RF-Beep* is based on the well known Time-Difference-of-Arrival (TDoA) scheme that utilizes the different propagation speeds of both the acoustic and the radio-frequency (RF) signals. Unlike the previous works, *RF-Beep* utilizes both the audio interface (i.e., microphone, speaker and sound driver) and the RF interface (i.e., WiFi) at the kernel-level of commercial-off-the-shelf smart devices. Implementing the scheme at lower levels enables us to understand and address the challenges related to the timing uncertainties in transmitting and receiving the acoustic signal. Moreover, *RF-Beep* does not require any special hardware or infrastructure support. In this paper, we describe the complete implementation of *RF-Beep* at the kernel space of Linux OS. We evaluate *RF-Beep* under different indoor and outdoor real scenarios. Results show that the error in the estimated range is less than 50cm for more than 93% of the time.**

I. INTRODUCTION

Accurate indoor/outdoor ranging schemes [1], [2], [3], [4], [5], [6] enable many useful and interesting applications such as accurate navigation directions [4], [2], efficient network management [7], face-to-face multiuser gaming applications [5], photo sharing [8], [9] and video viewing application [10], driver's phone detection[11], advertising applications [12], etc. Recently, several acoustic based ranging schemes [1], [13], [14], [2], [3], [4], [5] were proposed as high-accuracy ranging schemes (i.e., ranging error is in centimeter level). In general, each of these schemes exploits the slow propagation speed of the acoustic signal to achieve the high accuracy ranging.

In this paper, we design, implement and evaluate *RF-Beep* - a high-accuracy light-weight ranging scheme for commercial-off-the-shelf (COTS) smart devices. *RF-Beep* is based on the Time-Difference-of-Arrival (TDoA) scheme that utilizes the different propagation speeds of both the acoustic and the radio-frequency (RF) signals. Unlike previous works, *RF-Beep* scheme utilizes both the audio interface (i.e., microphone, speaker and sound driver) and the RF interface (i.e., WiFi) on smart devices at the kernel-level. Implementing the scheme in the kernel space enables us to understand and address the challenges related to the timing uncertainties in transmitting and receiving the acoustic signal.

Our experiments show that the delay between issuing the transmission command of the acoustic signal by the application and the actual emission of the acoustic signal by the speaker varies from 2ms to 6ms (Figure 3). However, by issuing the transmission of the acoustic signal at the kernel space instead, RF-Beep reduces the corresponding timing uncertainty to the order of microseconds (i.e., less than 1ms).

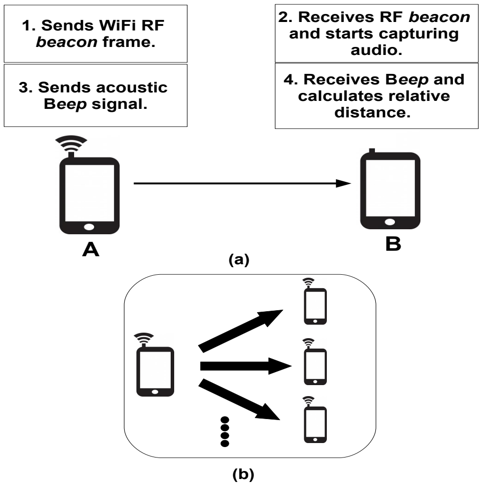

Figure 1: (a) Basic use case of *RF-Beep*, (b) *RF-Beep* for one-to-many scenario.

Figure 1(a) illustrates the basic use case of *RF-Beep*. In this illustration, device 'A' could be any smart device that is capable of generating RF and acoustic signals through RF (i.e., WiFi) and audio (i.e., speaker) interfaces respectively. Smart device 'A' could play different roles according to the usage scenario such as: a gateway/base node in ad hoc/sensor networks, an access point in enterprise networks, a Group Owner (GO) in WiFi Direct network [15], a peer node in ad hoc networks, etc. Similarly, device 'B' is a smart device that wants to estimate the relative distance to device 'A'.

In *RF-Beep*, device 'A' broadcasts a small RF frame, which we refer to as *RF beacon*, followed by an acoustic signal (*Beep*) after a short delay. The *Beep* sound is a high frequency sinusoidal acoustic signal with a single frequency. When device 'B' receives both the RF beacon and the *Beep* sound from device 'A', it will be able to calculate the distance to device 'A'. The time difference between the reception of both the RF beacon and the *Beep* sound is used to calculate the range between the two devices. Since the calculation is done locally at the receiver device, *RF-Beep* could be used by multiple devices concurrently as shown in Figure 1(b). For example, in the WiFi infrastructure mode of A2PSM [16], each WiFi client could estimate its relative distance to the access point concurrently to be used in calculating the exact time to wake up in order to minimize its power consumption.

As shown in Figure 1(a), only device 'A' transmits both signals while device 'B' (and other devices in Figure 1(b)) need only to receive the transmitted signals. Therefore, we classify *RF-Beep* as a one-way sensing scheme in estimating the range. In addition, since device 'B' only needs to receive the transmitted signals to calculate the range with no additional transmission, device 'B' consumes low power when using RF-Beep. Moreover, in *RF-Beep*, device 'B' calculates the range locally without requiring any collaboration from device 'A'. Hence, *RF-Beep* preserves the privacy of device 'B'. Furthermore, this non-collaborative nature of *RF-Beep* eliminates the need for any central server or additional infrastructure support, which makes the system practically applicable under different conditions and scenarios (e.g., adhoc scenarios).

We summarize our contributions in this paper as follows:

- Design and develop *RF-Beep* - a high-accuracy lightweight ranging scheme for smart devices. *RF-Beep*, to the best of our knowledge, is the first to integrate both the audio and the WiFi interfaces of the smart devices at the kernel-level. This low-level integration allows us to understand and address the timing uncertainties in transmitting and receiving the acoustic signal.
- Implement *RF-Beep* on COTS smartphones. *RF-Beep* utilizes the unique integration of the speaker, microphone and WiFi hardware components of the smartphone.
- Evaluate the implemented scheme under different indoor and outdoor realistic scenarios.

The rest of this paper is organized as follows: in Section II, we list the related work, and highlight the differences between our scheme and those related work. We describe in details *RF-Beep* and its corresponding challenges in the Section III. In Section IV, we describe the architecture of *RF-Beep* and the implementation details of the proposed modules. Evaluation of *RF-Beep* under different realistic indoor/outdoor scenarios is presented in Section V. Finally, we conclude and highlight our future work in Section VI.

II. RELATED WORK

Many localization schemes leverage the signal strength of multiple RF signals from different nearby RF sources or infrastructures (e.g., WiFi Access point, Cellular Tower). These RF based schemes accompanied with sophisticated localization algorithms could achieve reasonable accuracy with error of 6-8m. Recent work [4] shows that the existence of the same signature or fingerprint of the RF signal at different distinct locations prevents us from having good accuracy especially for indoor localization schemes [17], [18], [19], [20]. Recently, researchers combined multiple modalities such as sound with the WiFi to achieve higher accuracy localization scheme [2], [4], [21], [22]. For example, localization schemes in [2], [4], [3], [5] utilize an acoustic-based ranging scheme (i.e., BeepBeep [1]) in combination with the RF to improve the localization accuracy. In this section we highlight the recent and most relevant ranging techniques as well as the localization systems that use ranging techniques.

BAT: BAT [21] is an indoor localization system that utilizes the time-of-flight of ultrasound signals. BAT system consists of a RF base station, number of ultrasound receivers, and a customized transmitter carried by personnel. The RF base station orchestrates the activity of the receivers by broadcasting a message to them. Then, the transmitter sends out an ultrasound pulse that is received by the ultrasound receiver elements. Each receiver determines the time interval between the RF message and the ultrasound signal to infer its distance to the transmitter. BAT system can achieve high accuracy localization with large number of deployed ultrasound receivers. In BAT system, the ultrasound receiver needs to be synchronized with the RF base station thru the broadcast message from the base station. However, such technique has a number of uncertainties that have been discussed by Fikret et al. [23]. In addition, BAT system needs special customized hardware.

Cricket: Cricket [22] is an indoor localization system that utilizes the combination of RF and ultrasound to determine the distance between the target device and the transmitter. It uses the difference in arrival times of concurrent transmissions of radio and ultrasound signals at the target device to infer the distance. Although Cricket uses two different signals similar to *RF-Beep*, it requires special customized hardware for time stamping and to ensure concurrent transmissions of both the RF and the acoustic signals. Cricket system also requires infrastructure to place the transmission devices within.

PinPoint: PinPoint [6] uses the Time-of-Arrival (ToA) of multiple radio signals for location estimation. It uses a mathematical approach to compensate for the clock difference between different nodes. Although PinPoint does not require synchronization between nodes, it requires two-way sensing between nodes to estimate the relative distances. Moreover, PinPoint requires custom made expensive hardware. Recently, authors demonstrated in [24] how to implement a modified PinPoint using COTS 802.11 hardware. However, the proposed scheme requires collaboration between devices and it only achieves moderate accuracy of the meter level (i.e. 3m) for simple Line-of-Sight scenarios.

Whistle: Whistle [25] is an acoustic based Time-Difference-of-Arrival (TDoA) localization scheme. Whistle consists of several receiving devices and a source device. In whistle, the source device is to be located and the receiving devices are at known locations. The source device generates an acoustic signal followed by another acoustic signal from one of the receiving devices. All of the receiving devices sense these two signals and count the samples between the two signals to estimate the relative distance between that receiving device and the source device. In this scheme, both the source device and a receiving device have to generate an acoustic signal. In addition, Whistle requires collaboration between the receiving devices to estimate the location of the source device.

BeepBeep: BeepBeep [1] is an acoustic based high accuracy ranging system for smartphones. To calculate the range between two phones in BeepBeep, each of the two phones generates an acoustic beep sound. Then, the phones estimate the time difference in receiving both beep sounds (the one from itself and the one from the other phone) to calculate the relative distance between the phones. BeepBeep intelligently avoids the time synchronization requirement in estimating the relative distance. In addition, the uncertainty of sending and receiving acoustic signals has also been addressed. However, in Section III-B we explain how BeepBeep system overlooks the timing uncertainty of sending and receiving the acoustic signal.

BeepBeep is a two-way sensing ranging scheme. It requires both devices to collaborate in calculating the distance. As shown by the authors in [1], the relative distance estimation equation requires the dimension information of both phones as well as the reception timestamps of the two acoustic signals at both phones. Therefore, using BeepBeep in high accuracy localization system [4], [2] requires a central controller to coordinate the collaboration between the receiver and the transmitter. This requirement hinders the usage of the BeepBeep ranging scheme in localization systems.

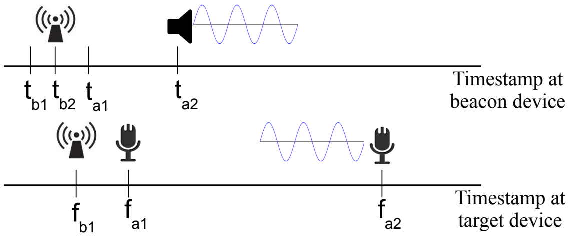

Figure 2: Overview of RF-Beep

III. RF-BEEP

In this section, we describe in details the proposed *RF-Beep*. We start by describing the scheme overview, followed by the implementation challenges, and then how we address those challenges. A device in *RF-Beep* could play one of the following two roles:
i) *Beacon device*: In this role, the device (e.g., device 'A' in Figure 1(a)) is responsible for transmitting both the RF beacon and the Beep sound periodically. The *beacon device* is used as the point of reference in measuring the relative distance. An example of a *beacon device* could be a smartphone in ad-hoc mode, a leader node among a group of sensor nodes in a wireless sensor network, an access point in an enterprise wireless network, etc.
ii) *Target device*: In this role, the device measures the range to the *beacon device*. The *target device* records the reception timestamps of both the RF beacon and the Beep sound from the *beacon device*. These timestamps help the *target device* to infer the relative distance to the *beacon device*. An example of a *target device* could be any smart device or a sensor node in a wireless sensor network.

A. RF-Beep Overview

RF-Beep is a ranging scheme based on the Time-Difference-of-Arrival (TDoA) technique that utilizes the relative velocity of two different signals; RF and acoustic. Figure 2 gives an overview of *RF-Beep* and shows the time-line of timestamps corresponding to different events at both the *beacon device* and the *target device*. Note that, all timestamps in Figure 2 are measured in milliseconds. These timestamps are summarized in the following table:

Timestamp Parameters	
t_{b1}	Time when the *beacon device* puts the RF beacon into the transmission buffer of the RF driver.
t_{b2}	Time when the last bit of the RF beacon is emitted from the *beacon device*'s RF hardware.
t_{a1}	Time when the audio driver starts writing audio frames into the audio hardware buffer.
t_{a2}	Time when the speaker starts to generate the Beep sound from the *beacon device*
f_{b1}	Time when the *target device* receives the last bit of the RF beacon.
f_{a1}	Time when the microphone of the *target device* captures the audio samples in the audio driver buffer.
f_{a2}	Time when the *target device* detects the starting of the Beep sound from the captured audio samples.

In *RF-Beep*, the *target device* only receives both the RF beacon and the Beep signal in which it does not require to share any information with the *beacon device*. Such flexibility enables the *target device* to calculate the relative distance to the *beacon device* locally. As we will show later, *RF-Beep* does not require any time synchronization between the *beacon device* and the *target device*.

A typical Time-of-Arrival (ToA) / Time-Difference-of-Arrival (TDoA) scheme uses the propagation speed of the signal to infer the distance between the transmitter and the receiver. The precision in determining the travel time is proportional to the propagation speed of the signal. For example, a RF signal requires high precision in calculating the travel time due to its high speed. On the other hand, since the sound has lower propagation speed compared to RF signals, an acoustic signal requires relatively lower precision in determining the travel time. For example, a millisecond error in ToA/TDoA estimation of the acoustic signal results in range estimation error up to 30cm. In order to limit the ranging error to few centimeters, we managed to maintain the error in *RF-Beep* time precision to less than 1 millisecond, as we will describe later.

Given the high propagation speed of the RF signal and the small length of the RF beacon, it takes less than a millisecond to transmit the RF beacon from the *beacon device* to the *target device*. Therefore, although the values of t_{b2} and f_{b1} are different on the two devices' timelines, we approximate both timestamps t_{b2} and f_{b1} to represent the same event on both timelines. Given the speed of the sound in air is s_a and the distance between the *beacon device* and the *target device* is D, we can write the following equation using Figure 2:

$$\begin{aligned} D &= s_a.(t_{a2} - f_{a2}) \\ &= s_a.(t_{a2} - f_{a2} + t_{b2} - t_{b2}) \\ &= s_a.((t_{a2} - t_{b2}) - (f_{a2} - f_{b1})) \\ &= s_a.(\Delta t_{ab} - \Delta f_{ab}) \qquad (1) \end{aligned}$$

In equation 1, Δt_{ab} represents the time difference between the transmission of the last bit of the RF beacon and the actual emission of the Beep signal by the speaker of the *beacon device*. Similarly, Δf_{ab} represents the time difference between the reception of the last bit of the RF beacon and the beginning of receiving the Beep signal at the *target device*. Both Δf_{ab} and Δt_{ab} values are measured locally at the *target device* and the *beacon device* respectively. Therefore, it is worthwhile to point out that there is no need for any type of synchronization between the two devices in order to calculate Δt_{ab} and Δf_{ab}. Moreover, in *RF-Beep*, as we will show later, we manage to limit the uncertainty in Δt_{ab} calculation to less than a millisecond. Thus, by fixing the Δt_{ab} value to a constant that would be known a priori, the *target device* does not require any additional information from the *beacon device*. Consequently, *RF-Beep* does not require any collaboration or exchange of information between the two devices.

B. Challenges

In TDoA, accuracy of the ranging method highly depends on the precision and the accuracy of measuring the arrival times of two different signals. Typically, TDoA based approach requires one to track the transmission/reception timestamps of two different signals. Therefore, this approach requires a tight synchronization between both the receiver and the transmitter. In order to address this synchronization requirement, most of the current schemes broadcast a periodic message from a centralized/infrastructure node [26], [27]. However, this approach introduces a number of uncertainties that have been described by Fikret et al. [23]. These uncertainties reduce the chance of maintaining precise time synchronization between multiple devices. To overcome the requirement of time synchronization, recent proposed ranging schemes [1], [2], [3], [4], [5] try to utilize the time difference between two local events in the range calculation instead of using a single local event. *RF-Beep*, according to equation 1, utilizes the same trick to tackle the time synchronization challenge.

Authors in [1] highlighted different types of uncertainties in any acoustic-based ranging scheme. One significant uncertainty is the high variation in the time delay between issuing the transmission command of the acoustic signal by the application and the actual emission of the acoustic signal by the speaker. Furthermore, the reception time of the acoustic signal at the *target device* depends on this transmission uncertainty at the *beacon device*. Figure 3 plots the CDF of the time delay between issuing the transmission command at the application-level and the actual emission of the acoustic signal (solid line) based on 1000 sample runs. It is clear from Figure 3 that the delay varies significantly within the range of 2ms-6ms, which leads to multiple meters error in estimating the range.

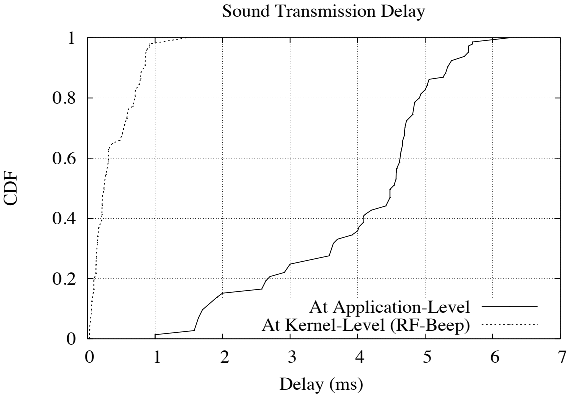

Figure 3: CDFs of the time delay between the issuing the transmission command of the acoustic signal and the actual emission of the acoustic signal when the transmission command is issued at the application-level (solid line), and at the kernel-level as in *RF-Beep* (dotted line). CDFs are calculated based on 1000 sample runs in which each consists of transmitting 4400 samples of sound data at 44100Hz sampling rate

Through careful study of the sound driver operation in details for Linux-based smart devices (e.g., Nokia N900 smartphone), we have figured out that the delay in the acoustic signal transmission consists of three main components that correspond to the following three actions: i) Powering up the playback stream, ii) Data transfer from the application to the sound driver, and iii) Direct Memory Access (DMA) data transfer from the sound driver to the actual sound hardware. Most of the portable smart devices have a sophisticated Dynamic Audio Power Management (DAPM) to minimize the power consumption of the audio system. DAPM makes power-switching decisions based on the audio stream (capture/playback) activity. Before playing the audio stream, DAPM takes some time to power up the playback subsystem of the audio system. Implementing *RF-Beep* within the sound driver provides us the flexibility in controlling this initial delay of powering up the playback stream. In addition, executing the transmission of the acoustic signal at the driver-level help us to get rid off the delay of transferring the audio data from the application space to the sound driver space. Thus, *RF-Beep* has only one source of uncertainty component that corresponds to the DMA data transfer from the sound driver to the actual hardware. However, experiments show that the impact of this source of uncertainty is less then 1 millisecond as shown in Figure 3 (dotted line). In Section IV, we will explain in details the implementation of the Beep sound transmission mechanism in *RF-Beep* for Linux-based sound drivers.

At the receiving side, *RF-Beep* has two uncertainty delays in receiving the Beep sound corresponding to: i) Powering up the capture stream, and ii) Detecting the starting event of the captured Beep sound. Figure 4 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 2-2.5ms maximum. Note that, this delay measurement allows us to determine how much delay (Δt_{ab}) we should consider before sending the Beep sound from the *beacon device*. Both the power up delay and the delay of detecting the starting event of the sound are exclusive, so we can write the Δf_{ab} as:

$$\Delta f_{ab} = \mu + \frac{n_{ab}}{f_s}. \qquad (2)$$

where μ is the delay to power up the capture stream, n_{ab} is the starting sampling number of the captured Beep sound, and f_s is the sampling frequency of the capture event. Note that, $\frac{n_{ab}}{f_s}$ is the delay in receiving the Beep sound. Detecting the n_{ab} in the captured sound data is a challenging task. In order to have high precision in measuring the range, it is critical to precisely detect the n_{ab} value. A number of reasons can make it challenging to perform such a task. For example, in Non-Line-of-Sight (NLoS) situation, multipath effect could make it ambiguous to detect the starting of the Beep sound properly. Moreover, the hardware (microphone and speaker) creates some large waveform distortion while generating/receiving the acoustic signal. In the implementation section, we address the challenges of detecting the Beep signal in more details.

IV. IMPLEMENTATION

RF-Beep utilizes both the acoustic interface and the RF interface at the kernel-level of the commercial available smart devices. In our implementation, we use the WiFi hardware of the smart devices as the RF interface and the WiFi beacon message as an example of the RF beacon. We implement *RF-Beep* at the kernel space of the Linux operating system in Nokia N900 smartphones. Although kernel-level implementation provides more flexibility, it is a challenging task to combine both the acoustic and the WiFi interfaces at the kernel-level in smart devices. More specifically, it is challenging to complete a sequence of operations within certain time constraint at the kernel-level in smart devices. In this section, we describe the architecture of *RF-Beep* in which we address the above challenges. We also describe in more details the generation and detection processes of the Beep sound.

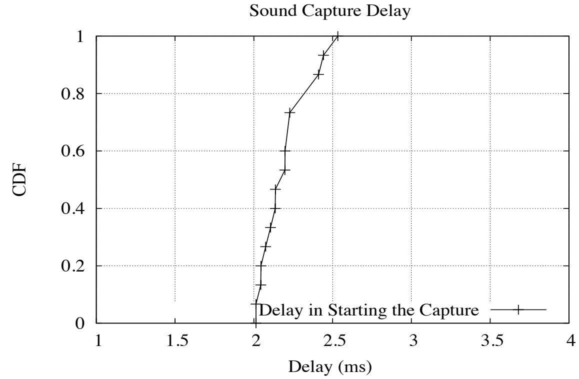

Figure 4: CDF of the time delay in powering up the capture stream based on 1000 sample runs

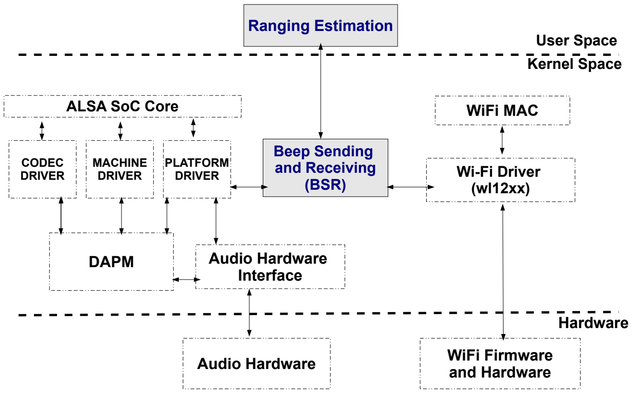

Figure 5: RF-Beep architecture for Linux-based smart devices.

A. RF-Beep Architecture

Figure 5 shows the different modules of *RF-Beep* (the shaded boxes) along with the standard modules of the sound driver and the WiFi driver (the white boxes). Most of COTS smart devices have Advanced Linux Sound Architecture (ALSA) SoC driver. ALSA SoC basically splits the embedded audio system into three main components: i) Codec Driver, ii) Platform Driver, and iii) Machine Driver. Among them, the platform driver is responsible for the DMA data transfer between the sound driver and the audio hardware. In Figure 5, the Beep Sending and Receiving (BSR) module is the main kernel module in *RF-Beeb*. This module is responsible for creating the Beep signal and transmitting it to the platform driver for further DMA transfer to the actual hardware. In addition, this module also processes the captured sound signal from the platform driver for detecting the Beep signal. In the *beacon device*, whenever the WiFi interface transmits a RF beacon frame, a notification signal is sent to the BSR module. The BSR module then powers up the playback stream, generates the acoustic Beep signal, and transmits it to the platform driver. In case of the *target device*, whenever the WiFi interface receives a beacon frame, a signal to the BSR module is sent to power up the capture stream and starts recording the audio samples in the allocated buffer. Finally, the BSR module detects the starting event of the Beep signal from the captured acoustic samples and transfers all the timestamp information to the *Ranging Estimation* module. The *Ranging Estimation* module, finally, calculates the range based on the collected information from the BSR module.

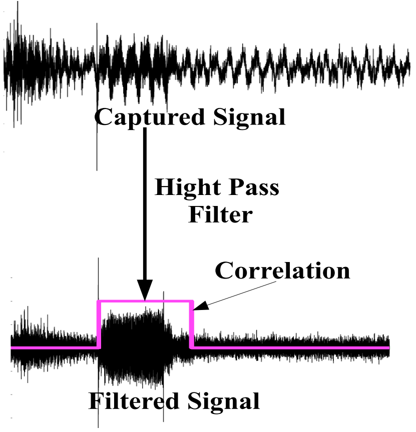

Figure 6: Beep signal detection mechanism.

B. Beep Signal Generation

In *RF-Beep*, we timestamp the interrupt event of the WiFi driver (wl12xx) that corresponds to a successful transmission of the RF beacon. This interrupt event approximates the transmission of the last bit of the beacon frame. Following the interrupt event, the WiFi driver sends a command to the BSR module to send the Beep sound after a fixed time delay. In our implementation, we set this fixed delay to 5ms. In addition, we use the 18kHz frequency to generate the Beep signal in the implementation. This frequency is closer to the upper limit of human hearing range that makes it is hard to be perceived by a person. On the other hand, COTS smartphones are sensitive enough to capture such high frequency sound. Moreover, given most of the background ambient noise is below 7-8kHz, the generated Beep sound using 18kHz will be robust to most of the background noises.

The duration of the Beep signal is another important parameter for the detection of the signal at the *target device*. For example, large duration of the Beep signal might create several multipath signals that will overlap with the original signal and make it harder to be detected. On the other hand, small duration of the Beep signal make it hard to be noticed at the receiving side. Empirically, we found that Beep signals with length of 2205 samples, which is equivalent to 50ms duration given the sampling rate of 44100Hz, yield the best performance. In Figure 5, the BSR module is responsible for generating the sound samples of the Beep signal and for transferring them to the corresponding buffer. Once the buffer is filled up with the samples, the platform driver initiates the DMA transfer between the buffer and the audio hardware.

C. Beep Signal Detection

Precise detection of the Beep signal at the receiving side is crucial for accurate range estimation. Our detection mechanism should satisfy the following two features: 1) Precise identification of the first sample of the Beep sound, and ii) Light-weight implementation. Figure 6 shows the steps in detecting the Beep signal. First, we apply a high pass filter over the sample data to get rid of all ambient background noise. Following the high pass filter, we apply L-2 norm cross-correlation over the filtered signal. The correlation values that exceed a certain threshold indicate the existence of the Beep signal.

We use a lightweight filtering method in the time domain to satisfy the second feature mentioned above. In defining the high-pass filter in the time domain, we label the input signal as $x[i]$ and output signal as $y[i]$. In general, the first-order filter [28] can be expressed as:

$$\begin{aligned} y[i] &= (1-k)\cdot x[i] + k\cdot y[i-1] \\ y[i] &= x[i] - y[i] \end{aligned} \tag{3}$$

where $k = \exp - \frac{2\cdot\pi\cdot f_c}{f_s}$, f_c is the cutoff frequency for the high pass filter, and f_s is the sampling frequency of the capture event.

From the above equation, we can construct a higher order high pass filter in the time domain. In our implementation, we use 5th order high pass filter which embed 5 samples delay in detecting the starting event of the *Beep* signal. This delay can lead up to 4cm of error, which is negligible in our scenarios. In order to use the L-2 cross-correlation, we use a short duration sinusoidal signal that consists of only 25 samples with the same frequency as the Beep signal. We select the duration length of the sinusoidal signal empirically to be of 50ms duration as discussed in Section IV-B. The correlation is done between the short sinusoidal signal and the filtered signal in a sliding window fashion with one sample increment. We identify the beginning of the Beep sound as the first index number of the captured sample that has: i) a correlation value above the specified threshold, and ii) the correlation values of the next captured 25 samples or more are above the threshold too. In the implementation, we set the threshold value to 0.8 to detect the beep signal. In an indoor environment, reflections from the surrounding walls might create multipath signals that overlap with the direct Line-of-Sight (LoS) signal. Although the combination of multipath signals may cause stronger signal than the direct path signal at the receiving side, the direct path signal always comes earlier than the other multipath signals. Therefore, we detect the beep signal by locating the earliest correlation value that exceeds the threshold.

V. EXPERIMENTS AND EVALUATION

A. *Equipment*

To evaluate *RF-Beep* performance on real testbed, we implement *RF-Beep* on two Nokia N900 smartphones running Maemo 5 Linux-based OS [29]. We use one phone as the beacon device while the other acts as the target device. Nokia N900 phone has two speakers laid out at the top and at the bottom surface of the phone, and a microphone located at the bottom part of the front surface. The audio features of the device are supported by the ALSA SoC driver. The WiFi chipset of the Nokia N900 phone is TI WL1251, supported by the wl12xx driver. In our implementation, the BSR module interacts with both the wl12xx driver and the ALSA SoC driver.

B. *Scenarios*

We evaluate *RF-Beep* under the following three scenarios:

- Indoor-quiet scenario: In this scenario, we conduct the experiments in our research lab in a quiet environment. Figure 7 shows the different positions of the *target device* and the *beacon device* within the lab layout.
- Indoor-noisy scenario: We conduct the experiments inside the Student Union Center during the lunchtime. During this time, too many students were moving around and chatting. Such activities create a very noisy and dynamic environment for our experiment.
- Outdoor open-space scenario: In the scenario, we chose an open area parking space inside our campus.

Since the sound speed varies with the actual temperature of the air [30], we measure the temperature of the environment before conducting the experiment using the TinkerKit thermostat sensor. In the experiments, we use the following model for calculating the speed of the sound, $s_{air} = 331.3 + 0.6 \cdot \theta$, where θ represents the temperature of the air in Celsius [30]. The measured θ during our indoor-quiet, indoor-noisy, and outdoor experiments was 28, 22, and 23 Celsius respectively. During the experiments, we place both the *beacon device* and the target device in an orientation where the phone's front surface is facing up. We run *RF-Beep* 100 times at each location to measure the range between the *target device* and the *beacon device*.

In the following experimental evaluation, we use the box plot to present the error statistics of the estimated range using *RF-Beep*. The lower bound of the box plot defines the 25th percentile of the error values while the upper bound defines the 75th percentile of the error values. The horizontal line inside the box plot represents the median error. The upper and the lower limit of the vertical line show the max and min error values, receptively.

C. *Results*

Figure 7 shows the indoor experiment setup for *RF-Beep* ranging scheme. The circle symbol with 'S' indicates the position of the *beacon device* while the other five square boxes show the different positions for the *target device*. In the figure, the positions 'A', 'B', 'C', 'D' and 'E' are 460, 585, 360, 930 and 700cm away from the *beacon device* respectively. The statistical presentation of the ranging error for each position is shown in Figure 8. We observe, from the figure, that the median error is quiet stable at the different positions. Moreover, 75% of the ranging errors are below or equal to 50cm for each position in the experiment.

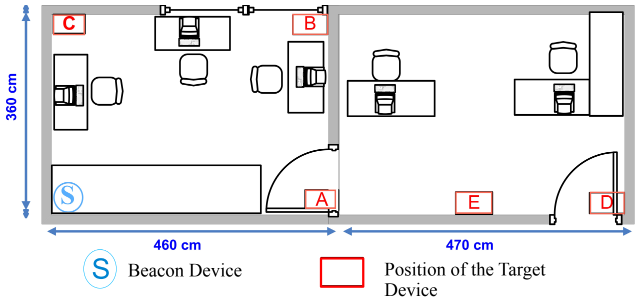

Figure 7: Lab layout setup of the indoor-quiet scenario for evaluating *RF-Beep*.

Figure 8 shows that the error of some runs are greater than or equal to 1m at some positions. However, the total number of these runs with high error is only 0.8% of the total runs. Note that the small heights of the quartile boxes in Figure 8 indicates the consistency of the range calculation at different positions. Such results conclude that *RF-Beep* performance is consistent over the different setups of the indoor-quiet scenario. We also notice that the median error in Figure 8 is higher than the other scenarios as shown later. The primary reason for the high median error in this scenario is due to the multipath effect caused by the closer surrounding walls of our lab.

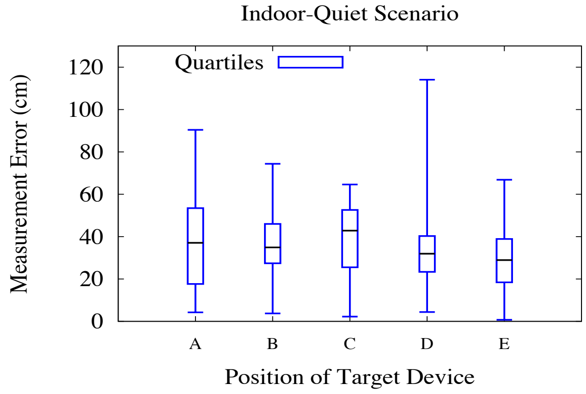

Figure 8: Error values of *RF-Beep* for indoor-quiet scenario. X-axis positions correspond to red squares in Figure 7.

Figure 9 shows the statistics of the ranging error for different distances between the *beacon device* and *target*

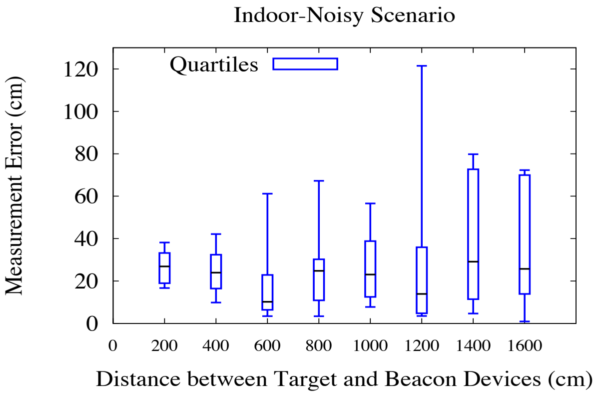

Figure 9: Error values of *RF-Beep* for indoor-noisy scenario.

device for the indoor-noisy scenario. As shown in the figure, shorter distances show relatively less variation of the error compared to longer distances. From the figure, 75% of the error values are less than 40 cm for all the distances below 14 meters. For the 14 and 16 meters experiments, we observe that 75% of the error values are less than 70 cm. However, the median error for these higher distances (14m and 16m) is as low as 30cm. These results show that *RF-Beep* is robust and highly accurate even in noisy and dynamic scenarios.

Despite the high dynamics of indoor-noisy scenario, Figure 9 shows lower median error compared to the other indoor scenario in Figure 8. We can justify this behavior by two observations. First, most of the human voices and background noises are below the 4kHz, which is far below the frequency we use for the Beep signal. Therefore, the students' chatting and the background noises are not interfering with the generation and the detection processes of the Beep signal and, consequently, not affecting the accuracy of *RF-Beep*. Second, given that the wavelength of the acoustic signal is larger than the size of the small chair/table and human body, these objects do not reflect acoustic wave. Since the experiment was conducted in a relatively open space compared to the previous indoor experiment, existence of multipath signals is minimal compared to the previous scenario. Hence, the detection of the Beep signal is more accurate, which results in lower ranging error.

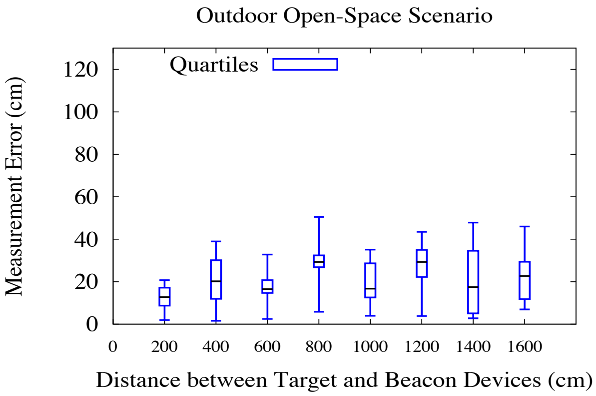

Figure 10: Error values of *RF-Beep* for outdoor scenario.

Figure 10 shows the ranging error at different distance for the outdoor scenario. From the figure, the error variation as well as the median error increases as the distance increases. In Figure 10, relatively short height of the quartile box indicates a consistent ranging error compare to the indoor scenarios. In outdoor scenario, we observe that 75% of the runs show ranging error less than 35cm. This result indicates that *RF-Beep* has better accuracy in outdoor scenario. This is because of the higher accuracy of the Beep signal detection process in the outdoor scenario due to the rare multipath occurrence compared to the indoor scenario. Consequently, Figure 10 shows lower median error. For all the above experiments, the ranging error is less than 50cm for more than 93% of the runs.

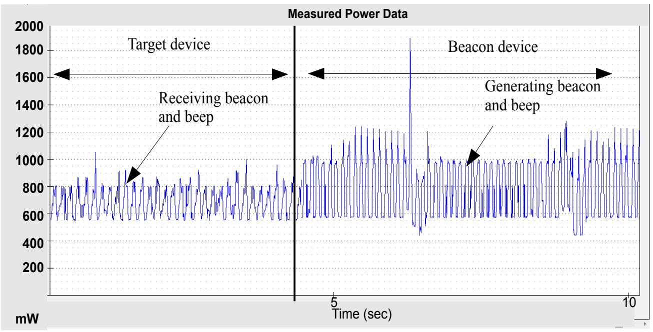

Figure 11: Energy consumption of *RF-Beep* for both devices.

In order to evaluate the power consumption of *RF-Beep*, we connect both the beacon device and the target device (i.e., Nokia N900 smartphones) to the Monsoon Power Monitoring tool [31]. During power monitoring, we turn off all radio interfaces (e.g. Cellular, Bluetooth) except the WiFi interface for both the devices. In addition, we make sure that both devices have the same configurations as well as the set of active applications. Figure 11 plots the power consumption for both devices during the transmission and the reception of both the RF beacon and the Beep sound. In the figure, each energy consumption spike at the beacon device represents the event of generating the RF beacon message and the Beep sound. On the other hand, for the target device, each spike represents the reception of the RF beacon message and the Beep signal. From the figure, the spike for the target device jumps from 600mW to 800mW on average, whereas for the beacon device power consumption jumps from 600mW to 1000mW. Therefore, the beacon device consumes more power compared to the target device. Thus, RF-Beep for the target device (i.e. user's smart device) is power efficient.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have designed a light-weight high-accuracy ranging scheme for smart devices - *RF-Beep*. *RF-Beep* is a one way sensing scheme that eliminates the time synchronization requirement between the peer devices. In addition, RF-Beep utilizes the RF interface and the audio interface on the smart devices at the kernel-level to address the delay uncertainties in transmitting/receiving acoustic signals. Moreover, RF-Beep does not require any special or additional hardware to use. Finally, we evaluate our system under real different indoor/outdoor scenarios. Experiments show that, while the median ranging error is less than 40cm for all the different scenarios, the ranging error is less than 50cm for more than 93% of the runs. In future work, we would like to address the impact of the target device's orientation (e.g. horizontal, vertical etc.) and position (e.g. in handbag, pocket etc.) on the range estimation using *RF-Beep*. Furthermore, we will study how to enhance the accuracy of detecting the starting event of the Beep signal under different indoor/outdoor scenarios especially in multipath-rich scenarios. In addition, we will explore other approaches in designing the Beep signal in order to increase the robustness of the signal detection under different scenarios and conditions.

REFERENCES

[1] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices," in *Proc. of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, Sydney, Australia, 2007.

[2] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: Locating Devices in an Office Environment," in *Proc. of the 18th International Conference on Mobile Computing and Networking (Mobicom '12)*, Istanbul, Turkey, 2012.

[3] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, "On the Feasibility of Real-Time Phone-to-Phone 3D Localization," in *Proc. of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys 11)*, Seattle, WA, USA, 2011.

[4] 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 *Proc. of the 18th International Conference on Mobile Computing and Networking (Mobicom '12)*, Istanbul, Turkey, 2012.

[5] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, "SwordFight: Enabling a New Class of Phone-to-Phone Action Games on Commodity Phones," in *Proc. of the 10th international Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, Low Wood Bay, Lake District, UK, 2012.

[6] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "PinPoint: An Asynchronous Time-Based Location Determination System," in *Proc. of the 4th international Conference on Mobile Systems, Applications and Services (MobiSys '06)*, Uppsala, Sweden, 2006.

[7] T. Nadeem and L. Ji, "Location-Aware IEEE 802.11 for Spatial Reuse Enhancement," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1171–1184, 2007.

[8] D. Frohlich, A. Kuchinsky, C. Pering, A. Don, and S. Ariss, "Requirements for Photoware," in *Proc. of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, New Orleans, LA, USA, 2002.

[9] S. Counts and E. Fellheimer, "Supporting Social Presence through Lightweight Photo Sharing On and Off the Desktop," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*, Vienna, Austria, 2004.

[10] G. Shen, Y. Li, and Y. Zhang, "MobiUS: Enable Together-viewing Video Experience Across Two Mobile Devices," in *Proc. of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, San Juan, Puerto Rico, 2007.

[11] 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 *Proc. of the 17th International Conference on Mobile Computing and Networking (MobiCom '11)*, Las Vegas, NV, USA, 2011.

[12] Wimbeep, "Wimbeep," http://www.wimbeep.com/about/about-wimbeep.

[13] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke, "A High-Accuracy, Low-Cost Localization System for Wireless Sensor Networks," in *Proc. of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, San Diego, CA, USA, 2005.

[14] M. Kushwaha, K. Molnr, J. Sallai, P. Vlgyesi, M. Marti, and kos Ldeczi, "Sensor Node Localization using Mobile Acoustic Beacons," in *Proc. of the IEEE 2nd International Conference on Mobile Adhoc and Sensor Systems (MASS '05)*, Washington, USA, 2005.

[15] W.-F. Alliance, "P2P Technical Group, Wi-Fi Peer-to-Peer (P2P) Technical Specication v1.0," 2009.

[16] M. Uddin and T. Nadeem, "A2PSM: Audio Assisted Wi-Fi Power Saving Mechanism for Smart Devices," in *Proc. of the 14th Workshop on Mobile Computing Systems and Applications (HotMobile '13)*, Jekyll Island, GA, USA, 2013.

[17] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *Proc. of the 19th IEEE Conference on Computer Communications (INFOCOM 00)*, Tel-Aviv, Israel, 2000.

[18] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "WALRUS: Wireless Acoustic Location with Room-Level Resolution using Ultrasound," in *Proc. of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, Seattle, Washington, 2005.

[19] M. Youssef and A. Agrawala, "The Horus WLAN Location Determination System," in *Proc. of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, Seattle, WA, USA, 2005.

[20] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: A Dynamic Indoor Signal Map Construction and Localization System," in *Proc. of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, Uppsala, Sweden, 2006.

[21] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context-Aware Application," in *Proc. of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, Seattle, WA, USA, 1999.

[22] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, MA, USA, 2000.

[23] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: a survey," in *IEEE Network*, pages 4550, 2004.

[24] M. Mah, N. Gupta, and A. Agrawala, "Pinpoint Time Difference of Arrival for Unsynchronized 802.11 Wireless Cards," in *Proc. of the 29th IEEE Conference on Computer Communications (INFOCOM 10)*, San Diego, CA, USA, 2010.

[25] B. Xu, R. Yu, G. Sun, and Z. Yang, "Whistle: Synchronization-Free TDoA for Localization," in *Proc. of the 11st International Conference on Distributed Computing Systems (ICDCS '11)*, 2011.

[26] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, Los Angeles, CA, USA, 2003.

[27] J. Elson, L. Girod, and D. Estrin, "Fine-grained Network Time Synchronization using Reference Broadcasts," in *Proc. of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, Boston, MA, USA, 2002.

[28] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*, 3rd ed. Springer, 2004.

[29] Wiki, "Nokia n900," http://wiki.maemo.org/Nokia_N900.

[30] ——, "Speed of Sound," http://en.wikipedia.org/wiki/Speedofsound.

[31] M. S. Inc., "Monsoon Power Monitoring Tool," http://www.msoon.com/LabEquipment/PowerMonitor/.