Ultrasonic Time Synchronization and Ranging on Smartphones

Patrick Lazik Niranjini Rajagopal Bruno Sinopoli Anthony Rowe
Electrical and Computer Engineering Department
Carnegie Mellon University
{plazik,niranjir,brunos,agr}@ece.cmu.edu

***Abstract*—In this paper, we present the design and evaluation of a platform that can be used for time synchronization and indoor positioning of mobile devices. The platform uses the Time-Difference-Of-Arrival (TDOA) of multiple ultrasonic chirps broadcast from a network of beacons placed throughout the environment to find an initial location as well as synchronize a receiver's clock with the infrastructure. These chirps encode identification data and ranging information that can be used to compute the receiver's location. Once the clocks have been synchronized, the system can continue performing localization directly using Time-of-Flight (TOF) ranging as opposed to TDOA. This provides similar position accuracy with fewer beacons (for tens of minutes) until the mobile device clock dirfts enough that a TDOA signal is once again required.**

Our hardware platform uses RF-based time synchronization to distribute clock synchronization from a subset of infrastructure beacons connected to a GPS source. Mobile devices use a novel time synchronization technique leverages the continuously free-running audio sampling subsystem of a smartphone to synchronize with global time. Once synchronized, each device can determine an accurate proximity from as little as one beacon using Time-Of-Flight (TOF) measurements. This significantly decreases the number of beacons required to cover an indoor space and improves performance in the face of obstructions. We show through experiments that this approach outperforms the Network Time Protocol (NTP) on smartphones by an order of magnitude, providing an average $720\mu s$ synchronization accuracy with clock drift rates as low as $2ppm$.

I. Introduction

Positioning systems have already revolutionized how smartphones interact with the surrounding world. Indoor location-based services would enhance a wide variety of applications including: augmented reality, pervasive computing, advertising, social networking and asset management. Up to this point, location-aware applications primarily work outdoors by using the Global Positioning System (GPS). Unfortunately, satellite-based approaches do not work indoors since these weak signals do not easily penetrate through building walls. Various techniques have been proposed and developed for indoor positioning such as broadcast-based technologies (WLAN, RFID etc.) and motion-based technologies, such as inertial sensing-based positioning. Many technology companies have been racing to develop indoor location solutions like iBeacon [1] and Gimbal [2] that provide proximity-based services. RF-based solutions are notoriously unable to provide accurate ranging due to multipath caused by indoor clutter. They also suffer from the barrier problem where a node placed nearby on one side of a wall can incorrectly localize a user in the neighboring room. In order to compensate for poor ranging technologies, the number of beacons can be increased and the transmit power reduced to improve spatial resolution. However, this higher density of beacons increases setup time along with hardware and management costs.

Ultrasound-based localization systems have been shown to outperform RF-based systems in terms of accuracy. For example, the MIT cricket system [3] can range to within $5cm$, has a boundary detection accuracy of $1cm$ and can compute 3D position to within $10cm$ and orientation within 3 degrees. Ultrasound is comparatively much slower than light, which makes it easy to measure signal TOF and perform TDOA measurements. It also does not suffer from the barrier problem since ultrasound does not penetrate walls nearly as easily as RF. The main drawback to ultrasonic approaches is that they require custom hardware and infrastructure. Our goal is to bring many of the benefits of ultrasonic localization to current smartphones.

In this paper, we propose a platform capable of providing positioning and time synchronization to most modern mobile devices that have a microphone input. The platform consists of a small number of wireless nodes that are equipped with GPS receivers with a Pulse Per Second (PPS) output, positioned near windows in order to distribute time synchronization messages to the interior nodes using 802.15.4. The beacons then synchronously generate ultrasonic data streams that can be used for identification and ranging. The beacons also contain ultrasonic MEMs microphones that can be used to perform inter-beacon ranging to aid in infrastructure configuration. By computing the distance between each Line-Of-Sight (LOS) beacon, the system can solve instances of a graph realization problem to determine relative node positions.

In [4], the authors present a modulation approach that uses ultrasonic chirps just outside of the human hearing to uniquely identify and estimate distance to beacons from a mobile device. Most modern mobile devices such as smartphones and tablets contain $48kHz$ audio codecs and microphones that are sensitive to the near ultrasound frequency spectrum, allowing them to function as receivers for ultrasound based ranging systems. Since all beacons transmit simultaneously, the authors show that using the TDOA of each signal, along with beacon location data requires $N+1$ transmitters to localize a mobile receiver in N dimensions. This is often impractical in indoor environments that are sparsely provisioned with beacons. In order to determine the 3D location of a receiver, TDOA requires at least four transmitters while TOF would require only three transmitters. In addition, TOF can perform fairly accurate localization with less then N transmitters when coupled with known physical constraints imposed by the indoor layout. In realistic environments, this has a significant impact on deployment density and performance. For example, if nodes are deployed down a long hallway or if beacons are blocked by obstructions, the TDOA system often fails, while a TOF system is more likely return a correct result. The trade-off is that TOF requires that the receiver is time synchronized with the infrastructure.

We propose a novel technique by leveraging the free-running clock within a smartphone's audio sub-system to perform time synchronization. As part of the TDOA ranging calculation, it is possible to compute the time offset between each received ultrasonic pulse and the initial transmit time. This same approach is what allows GPS receivers to accurately time synchronize with the satellites. Since our beacons are fed by GPS PPS signals, the mobile device can precisely estimate the starting point of the last UTC second within its audio buffer. After capturing two transmit cycles, the mobile device can determine both the time and the skew rate of the audio clock source. Since the audio clock remains running freely, it will continue to record and count samples without any additional jitter that would normally be introduced by the OS and networking stack. Smartphones typically have multiple highly accurate time sources at their disposal including GPS, cellular tower time and NTP. Unfortunately, applications cannot directly access these timing sources without going through a relatively indeterministic call stack. Furthermore, synchronizing with an NTP server over WiFi is not scalable in large building environments, where the access points consistently service high traffic loads resulting in high time synchronization error. We experimentally evaluate that NTP time synchronization on a phone is often only accurate to within $10 - 20ms$. At a $48kHz$-sampling rate, the audio clock has a granularity of $22\mu s$, which on iOS devices can be processed with less than $1ms$ of jitter. Audio events (like new beacon signals) can be time stamped with the precision of the audio sampling granularity of $22\mu s$ while application events can be time stamped within $1ms$ due to jitter accessing the audio buffer. In both cases, this significantly outperforms existing time sources and allows the synchronization to take place solely with one-way communication to the receiving device.

Being able to tightly synchronize time between two devices is fundamental to distributed real-time systems. Though we utilize time synchronization for its ability to improve ranging, it can also be used for event ordering, determining event consistency and coordination of communication. One immediate application of tight time synchronization within the context of localization is the ability for devices to perform direct TOF ranging from beacons instead of TDOA ranging. After a node has heard from four or more nodes within a single area, the mobile device can synchronize with global time and then perform TOF ranging for any successive beacons. We experimentally evaluate that the skew rate of the audio clock on select mobile phones ranges anywhere from 7 to $40ppm$. With clock rate correction over multiple samples, we are able to achieve a stability of less than $2ppm$. This corresponds to $2.7m$ of drift per hour of ultrasonic ranging error. We envision indoor applications where in larger spaces there would be excess beacons that would allow the mobile devices to synchronize and then estimate TOF ranges that slowly degrade as the clock drifts until it reaches another region with four or more transmitters within range. Global time synchronization at this level of granularity can also be a powerful mechanism to enhance mobile-to-mobile cooperative localization.

The main contributions of this paper are (a) a hardware platform that is able to transmit time synchronized omni-directional ultrasonic pulses, (b) an improved data transmission scheme that can operate on devices with less bandwidth than what was presented in [4], (c) a time synchronization approach that utilizes the audio input system of a mobile device and (d) a rigorous evaluation of the ability of the platform to perform time synchronization and ranging.

A. System Architecture

Figure 1 shows the main components of our system including: a GPS receiver, a *Network Master*, a group of *Ultrasonic Beacons* and a mobile device with a microphone. One or more GPS units need to be placed in a location where their antennas have a clear view of the sky. A *Network Master* device rebroadcasts the Pulse-Per-Second (PPS) signal from the GPS over 802.15.4 within the indoor space. Each beacon has a power amplified 802.15.4 radio (described in Section V) such that the majority of the network is reachable within a single hop. One could imagine extending coverage using various existing multi-hop time synchronization approaches. The beacons also include a microphone that can be used along with the 802.15.4 radio for inter-beacon ranging. Though not discussed in detail in this paper, inter-node ranging can aid in network configuration as well as help reduce timing error derived from incorrect beacon location measurements. Since each *Network Master* device is GPS time synchronized, multiple master nodes can be placed around the perimeter of a building to improve coverage as necessary. Each *Ultrasonic Beacon* has an on-board 802.15.4 radio transceiver along with the hardware required to generate and transmit ultrasonic signals. Beacons transmit a signal (described in Section IV) with pre-defined release intervals, i.e. Time-Division-Multiple-Access (TDMA), that encode a unique beacon identification number. The mobile device, typically a smartphone, records audio at a high-sampling rate (e.g. $48kHz$) for a period of time to collect available beacon messages. The smartphone then demodulates any received signals, and uses the TDOA approach described in Section III to compute the offset from the start of the TDMA cycle in order to synchronize with the global GPS time. At this point, any application with access to the audio buffer can map audio sample count to global time using the audio DAC as an auxiliary clock that is synchronized with global time. After observing three or more transmitters within one TDMA cycle, the mobile device can perform direct TOF ranging until its clock drifts too far from global time (approximately 2.7 meters per hour assuming $2ppm$).

II. Related Work

This paper touches upon topics related to ultrasonic communication, ranging and time synchronization. All of these areas have large bodies of research, so our discussion will focus primarily on closely related work in the mobile computing space.

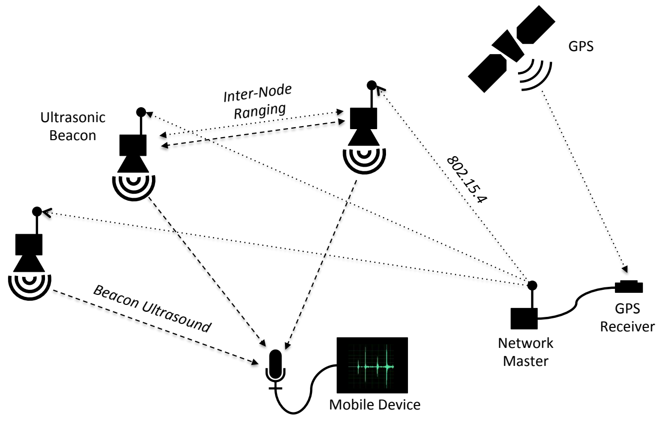


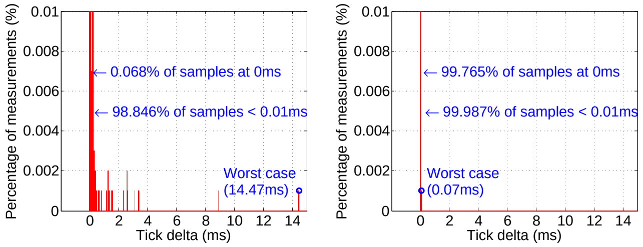
Fig. 1. System overview

A. Localization and Ranging

Research on the topic of localization can be broadly classified into two main categories of range-based approaches [3], [5], [6] and range-free approaches [7], [8], [9], [10]. Range-free approaches typically attempt to match either synthetic or naturally occurring signatures to a particular location. Range-based approaches on the other hand use measured distances or angular estimates between known anchor points to compute a position. We focus on range-based technologies including Time-of-Arrival (TOA), TDOA and TOF. For a more detailed general overview, we refer to [11].

TOF [12] systems compute distances based on how long it takes for a signal to propagate from a sender to a receiver. For example, [13] and [14] both compute distances by measuring the Round-Trip-Time-of-Flight (RTOF) by recording a signal's departure and the return time divided by the propagation speed. This assumes that the receiver will retransmit a return signal within a fixed amount of time. BeepBeep [14] uses this approach on cellular phones to compute inter-device ranges. Even though smartphones are typically sensitive to ultrasonic sound, their speakers are highly directional in those frequencies, which lead [14] to use audible frequencies. The authors also focused on peer-to-peer ranging rather than infrastructure to device ranging.

TDOA systems can remove the requirement of knowing exactly when a signal was transmitted by using what is known as *pseudo-ranging*. Pseudo-ranging computes distances by looking at the relative differences between the arrival of several signals, assuming they were all transmitted simultaneously or at known offsets. As compared to TOA and TOF approaches, this requires one additional transmitter to allow the common distance from all broadcasting devices to be estimated. GPS [5] is the most popular example of this ranging approach. Similar approaches have been applied towards ultrasonic communication [15], [16], [4]. The Dolphin [15], [16] system adopts a pseudo-ranging approach using using a $50kHz$ carrier with Direct-Sequence-Spread-Spectrum (DSSS) modulation. While extremely accurate, this approach requires custom hardware and is not applicable to standard smartphone devices. In [17], the authors expand upon Dolphin (while still requiring custom hardware) by adding a self-training deployment approach based on filtered motion within the space. This work in part inspired our inter-node ranging capability. In [18] the authors identify the location of a cellular phone in a car using ultrasonic pseudo-ranging from the car's audio speakers. This approach used fixed frequency tones in an extremely controlled environment where data transfer was not required.



(a) iPhone 4

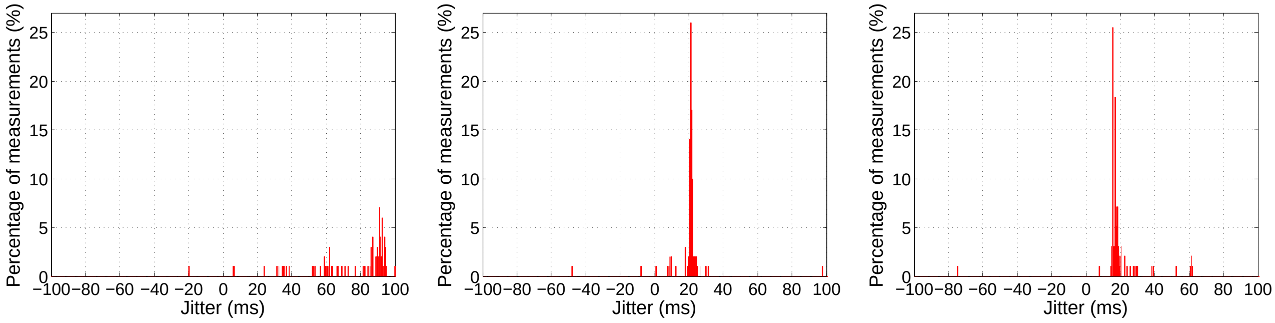
(b) iPhone 5S

Fig. 2. Delay in consecutive OS system time calls

In [4], the authors introduce an ultrasonic TDOA ranging approach that is able to perform ranging between speakers distributed in the environment and mobile devices. The system utilizes standard speakers and is evaluated on previous generations of smartphones with a wider frequency range above $20kHz$ than the current generation (iPhone 4 as opposed to 5-6). The approach also only supports pseudo-ranging and not TOF. In this paper, we extend concepts in this work in four ways: First, we introduce and evaluate an approach for time synchronizing the phones with the infrastructure that allows TOF ranging from a single beacon. Second, we improve on the original ultrasonic modulation technique by adding TDMA support in order to run on devices with less available audio bandwidth. Third, we create an embedded stand-alone hardware platform with an omni-directional transducer, which is critical for practical deployment scenarios. Finally, we add hardware support for inter-node ranging which can enhance system deploy-ability.

B. Synchronization

A significant amount of work from the distributed systems community has focused on time synchronization [19], [20], [21], [22]. The most commonly adopted of these approaches is the Network Time Protocol (NTP) that uses round-trip message delay averaging to set times. We show through our experiments that due to the asymmetric and lossy nature of modern wireless communication channels, it is extremely difficult to reach sub milli-second levels of accuracy. To the best of our knowledge, this is one of the first efforts to explore how tightly smartphones can synchronize with global time at the application level. Celltower and GPS synchronization exist within their own subsystem but are usually isolated from the main system clock or updated at a coarse granularity (seconds). There has been significant work related to message passing based approaches from the sensor networking community [23], [24], [25], [26] that can be applied towards infrastructure timestamping. Eventually these approaches could find their way into mobile phones. In our system, the beacons are synchronized with GPS time such that in the future when high-precision synchronization is ubiquitous, smartphones can leverage these sources to directly compute TOF ranges.



(a) Remote stratum 1 server over WAN via LTE

(b) Remote stratum 1 server over WAN via WiFi

(c) Local stratum 1 over LAN via WiFi

Fig. 3. NTP second tick deviation from UTC second tick

	Jitter (number of samples at 48kHz)		
	0	1 ($22\mu s$)	2 ($44\mu s$)
iPhone 4	66%	34%	0%
iPhone 5S	0%	87%	13%

TABLE I. AUDIO SAMPLING JITTER

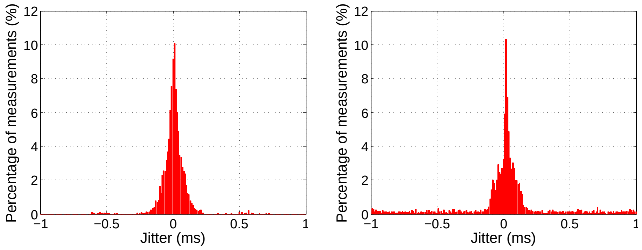
III. TIME SYNCHRONIZATION

Time synchronization is extremely valuable in a wide variety of applications. It enables event ordering, coordinated actuation, energy-efficient communication, low-power duty cycling and the ability to measure distances. In this section, we explore the limits of time synchronization on mobile platforms and propose a new approach that synchronizes the audio recording subsystem to external infrastructure in order to provide applications with the ability to perform precise time-stamping (especially of audio events) isolated from indeterminism within the operating system and network.

A. Network Time Synchronization

Benchmarking time synchronization accuracy on a smartphone is difficult because the current platforms typically do not expose low-level I/O and the operating systems are optimized for energy-efficiency rather then timing performance. In order to understand the nature of timing and synchronization on smartphones, we ran a set of timing experiments that examine key time-synchronization performance characteristics as described below.

1) Clock Granularity: We first need to establish the granularity of the clock on our mobile device in order to bound the minimal synchronization accuracy. This can be achieved by calling os_get_time_of_day() continuously in a tight loop and inspecting the tick increments. Figure 2 shows the resulting distribution of tick values for two iOS devices. The histograms show that the system clock has a granularity of 1-microsecond and that jitter associated with OS delays and context swaps is quite low. The faster of the two platforms (iPhone 5S) exhibits almost no jitter from background tasks. A similar experiment on an Android Galaxy Nexus 4 showed high levels of jitter for most reads, on the order of milliseconds and greater due to differences in Android's task scheduler. For this reason, we perform the rest of our experiments using iOS. Also, currently iOS has a lower latency audio subsystem as compared to Android. Round-trip audio times computed by looping test sounds back from input to output using [27] and are typically below $10ms$ on iOS devices.



(a) iPhone 4

(b) iPhone 5S

Fig. 4. Audio buffer OS time-stamping jitter

2) ADC Timing Performance: In order to compare the smartphone's clock against a reference source, we need to establish a low-latency input or output mechanism. Typical smartphone I/O includes: UART, Bluetooth, WiFi, LTE, the display, audio, external buttons and various sensors like light and acceleration. With the exception of audio and the UART, the interfaces exhibit milliseconds or greater amounts of timing uncertainty. The audio interface is particularly appealing since it contains its own continuous sampling clock that can be used for relative time stamping and has the ability to configure buffer sizes and sampling rates.

In order to estimate audio latency, resolution and clock drift, we connect the PPS output from a uBlox 7 GPS receiver through a level shifter into the microphone input of a smartphone. The GPS PPS output provides a highly stable clock reference with less then 25 nano-seconds of jitter. In our first experiment, we collect 1 hour of PPS input audio signals. Table I shows the jitter between the time that the PPS signal transitions as compared to the expected time based on the audio sampling rate. For example, a jitter value of 1 means that the PPS pulse period was 1 audio sample ($22\mu s$) different than expected over that period of time due to sampling jitter or clock drift. From these values we can compute that the clock drift-rate over an extended period on the iPhone 4 and iPhone 5S are $7.17ppm$ and $23.56ppm$ respectively as compared to the GPS reference. We see a worst-case sampling jitter between two PPS edges as 2 audio samples.

3) OS Timing Performance: In order to measure timing performance based on OS time stamping, we use the same experimental setup from the previous section and also timestamp the arrival of each audio buffer segment using the OS clock ($1\mu s$ granularity) and compute the relative PPS edge in the audio stream. Figure 4 shows that the worst-case jitter between OS tick time and a PPS tick is $724\mu s$ with an average jitter of $53.2\mu s$. Since the PPS signal is regular, this allows us to bound the OS timestamping jitter to within $1ms$.

4) NTP Timing Performance: We can now use our OS timestamping bounds to benchmark the performance of NTP running on the phone as compared to the GPS input. Using ios-ntp [?], we capture NTP timestamps along with the system time and our audio PPS input. For each NTP sample, we allow the server to synchronize for 200 seconds before comparing against the OS and PPS time. During this time the NTP process performs clock rate adjustment. Figure 3 shows the jitter between the NTP clock and the PPS timestamps over 100 different NTP synchronizations. Since NTP is negatively impacted by jitter and asymmetry in communication channels, we ran experiments using LTE, campus WiFi and an idle WiFi router directly connected to a Stratum 1 NTP server fed by a dedicated GPS clock. We see that NTP using LTE has an average jitter between synchronization attempts of $47ms$ (max $466ms$), while NTP over normal WiFi has an average jitter of $30ms$ (max $326.5ms$) and even in the isolated ideal case, there is an average of $19.3ms$ (max $74ms$) of jitter. For measuring audio TOF, a time of $1ms$ corresponds to a distance of $0.33m$ while the distance equivalent to $47ms$ is more than $15m$. This is not accurate enough to be useful for most indoor localization applications. In the next section, we describe our acoustic TDOA synchronization approach that improves timing accuracy to $720\mu s$ on average in practice.

B. TDOA Synchronization

As part of the TDOA calculation, it is possible to estimate the instant when each signal was originally transmitted. We use this approach to synchronize the audio stream with respect to global time, which can then be used as a reference for application-level time-stamping. Time-stamping of audio events based on their position in a buffer completely removes any sources of delay from the operating system or networking stack. Given the relatively small amount of jitter seen when sampling audio, it also stands as a reasonable alternative for synchronizing other events, for example to perform cooperative ranging between two mobile phones.

Figure 5 shows the layout of three transmitters and a receiver in 2-D space, and their corresponding notions of time. We consider the receiver's clock to be offset by T_{offset} from the transmitter's clock. Synchronization is achieved by estimating this offset. Typically this time offset is not estimated since the TDOA equations are used to directly estimate the position of the receiver [28]. However, the time offset can be obtained easily once the position has been estimated, as explained below.

(X_i, Y_i) denotes the position of transmitter i for $i = 1, 2, 3$ and is assumed to be known. The position of the receiver (x, y) is unknown. d_i is the distance between transmitter i and the receiver and is given by:

$$d_i(x, y) = \sqrt{(X_i - x)^2 + (Y_i - y)^2}$$

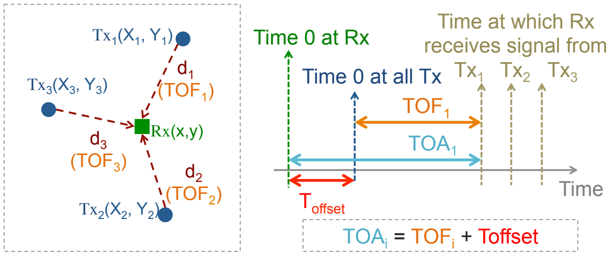


Fig. 5. Timings for estimation of T_{offset}

The TOF of the audio signal from transmitter i to the receiver is given by:

$$TOF_i = \frac{d_i(x, y)}{V}$$

where V is the speed of sound.
The corresponding arrival time of the signal measured by the receiver is the TOA_i, given by:

$$TOA_i = TOF_i + T_{offset}$$
$$TOA_i = \frac{d_i(x, y)}{V} + T_{offset}$$

The receiver needs to estimate T_{offset} given TOA_i and (X_i, Y_i) for $i = 1, 2, 3$. To estimate the T_{offset}, we first estimate the position of the receiver. To estimate (x, y), we use the standard multilateration technique [28] by eliminating T_{offset} and arrive at the TDOA equations. We then find the (x, y) that minimizes the sum of squares of error (ξ) in TDOA.

$$\begin{aligned}
Measured\, TDOA_{ij} =& TOA_i - TOA_j \\
True\, TDOA_{ij} =& \frac{d_i(x, y) - d_j(x, y)}{V} \\
\xi_{TDOA_{ij}}(x, y) =& [TOA_i - TOA_j - \frac{d_i(x, y) - d_j(x, y)}{V}]^2 \\
(\hat{x}, \hat{y}) =& \operatorname*{argmin}_{x,y} \sum_{\substack{(i,j) \\ 1 \leq j \leq N \\ j \neq i}} \xi_{TDOA_{ij}}(x, y)
\end{aligned}$$

We next estimate T_{offset} from $(\hat{x}, \hat{y})$ and the TOA by:

$$\hat{T_{offset}} = \frac{1}{3}\left(\sum_{i=1}^{3}\left(TOA_i - \frac{d_i(\hat{x}, \hat{y})}{V}\right)\right)$$

IV. Communication

The ultrasonic signals generated by the system must be able to provide accurate ranging information, support multiple access and encode data to identify the transmitter. In [4], data symbols are encoded as rate adjusted ultrasonic chirps, which rely on different rates of increasing/decreasing frequency over time to be identifiable by matched filtering. Chirps benefit from an effect known as Pulse Compression, which increases the SNR at the receiver by a factor $T\Delta f$ over a sinusoidal signal at equal transmission power, where T is the signal duration and Δf is the bandwidth. This improves both the range resolution as well as the SNR of the data symbols for better detection. This encoding scheme is no longer attractive due to the more limited frequency response of current iOS devices ($-35dB$ falloff between $19kHz$ and $22kHz$). To achieve the same

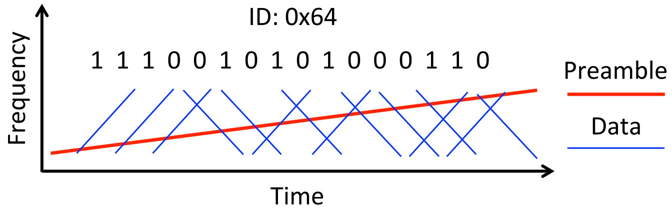
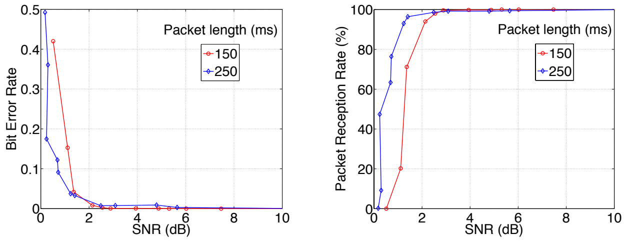


Fig. 6. Ultrasound packet structure



(a) (b)

Fig. 7. Performance of modulation scheme

SNR at the receiver with a quarter of the bandwidth, the packet length T would have to be quadrupled, reducing the data rate and making it more likely that interference is caused by multiple packets being received concurrently.

In order to maintain system performance, we propose the scheme shown in Figure 6. Since chirp signals sweep across a given frequency range, it is possible to partially overlap symbols without causing significant interference. This not only allows us to increase the data rate, but also increase the symbol duration, which increases the SNR of the signal at the receiver while keeping packet duration to a minimum. The preamble is stretched over the entire packet, making it easily discernible after matched filtering and improving the range resolution of the transmission. Data symbols have also been increased in length, with the start of each data symbol being separated by several milliseconds. Instead of using different chirp rates to denote four different data symbols, only an up-chirp (frequency increasing over time) and a highly orthogonal down-chirp (frequency decreasing over time) are employed to keep the scheme robust to errors. Each symbol is faded in and out in amplitude over $5ms$ to prevent the loudspeaker from producing audible clicks due to rapid changes in signal power.

Figure 7 shows the performance of our modulation scheme for two different packet lengths of $150ms$ and $250ms$. The $150ms$ packets use a data symbol length of $25ms$, a preamble length of $150ms$ and a symbol separation of $9ms$. The $250ms$ packets use a data symbol length of $30ms$, a preamble length of $250ms$ and a symbol separation of $15ms$. To test the performance of this modulation scheme, we transmitted 500 packets with random data while varying the transmit power to an iPhone 5S, which performed the demodulation of the received data. The $150ms$ long packets perform nearly as well as the $250ms$ long packets in terms of Bit Error Rate (BER) and Packet Reception Rate (PRR), with only a negligible impact on PRR down to an SNR of $2.5dB$. The $250ms$ long packets do not exhibit a significant impact on PRR until an SNR of $1.8dB$.

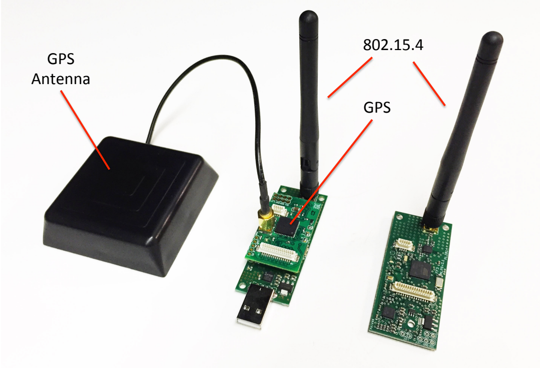


Fig. 8. Network Master (left) Utrasonic Board (right)

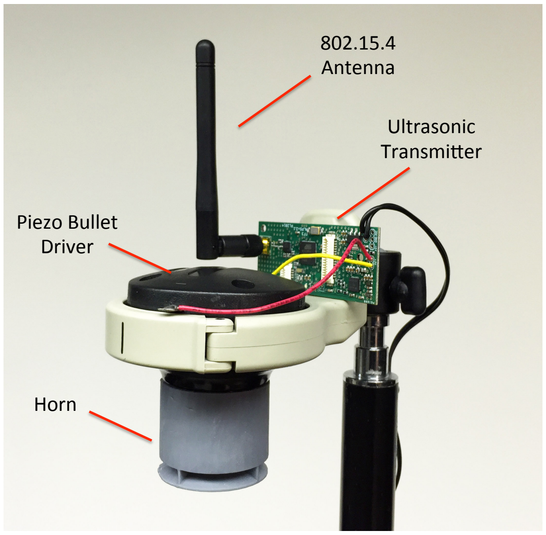


Fig. 9. Ultrasonic beacon

V. System Design

As decribed earlier, the two main hardware components in our system are the *Ultrasonic Beacons* and the *Network Master* that broadcasts GPS time. It is critical that all components in the system exhibit low levels of jitter and that the beacons are able to uniformly transmit our audio signal with minimal distortion.

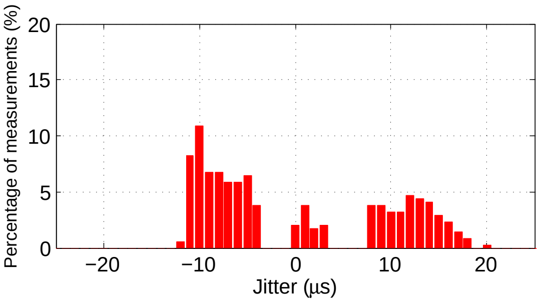


Fig. 10. Ultrasonic beacon transmission jitter

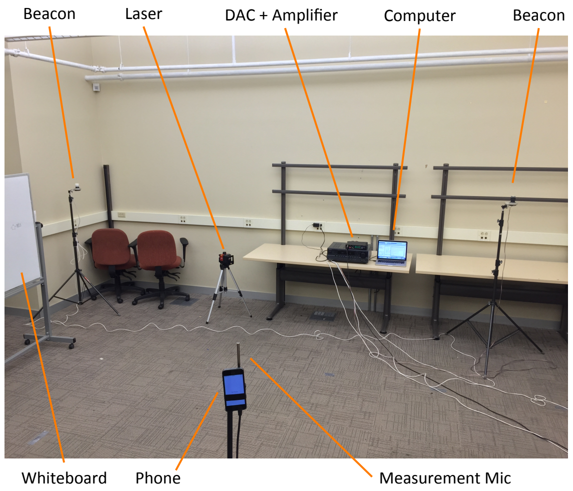


Fig. 11. Experimental setup

A. Hardware Design

Figure 8 shows a picture of our *Network Master* with a Venus 7 GPS chipset receiver connected to an external antenna. The GPS antenna is placed with a long cable to a nearby window such that it can receive at least 3 satellites required to generate an accurate PPS signal. Each node consists of a $16MHz$ ATmega256RFR2 processor with integrated 802.15.4 radio. In order to increase transmission range, we include a CC2591 amplifier that allows the node to transmit $100-200m$ indoors.

Figure 9 shows one of the *Ultrasonic Beacon* nodes mounted to a tripod. This configuration is easy to install into drop-down ceiling tiles in commercial spaces by drilling a hole for the horn. The beacon hardware consists of the main PCB, a piezo electric tweeter bullet driver and a custom designed ultrasonic horn. The *Ultrasonic Beacon* board consists of the same hardware as the *Network Master* with the addition of a TLV320 mono 24-bit, $192kHz$ DAC and an Akustica AKU340 ultrasonic MEMS microphone. The TLV320 communicates directly with the main processor using I^{2}S, which we generate using an SPI port and a timer output, and has both an integrated amplifier as well as an audio codec for sampling microphone data. It is worth noting that the TLV320 has a programmable PLL allowing us to transfer data at non-standard audio rates that are easier to generate from our micro-controller. In contrast to other comparable audio solutions (like embedded Linux platforms), this tightly coupled design allows us to have an end-to-end jitter from reception of an RF packet to excitation of the speaker of less than $20\mu s$ as shown in Figure 10.

In a typical loud speaker, as the audio frequency increases, the spread of the signal decreases, eventually forming a beam. In our system, we ideally want an omnidirectional speaker that has a flat frequency response across the $18-24kHz$ frequency band that can uniformly deliver data without distortion. Since no such speaker was commercially available, we designed a custom transducer based on a multi-sector omnidirectional horn design shown at the bottom of Figure 9.

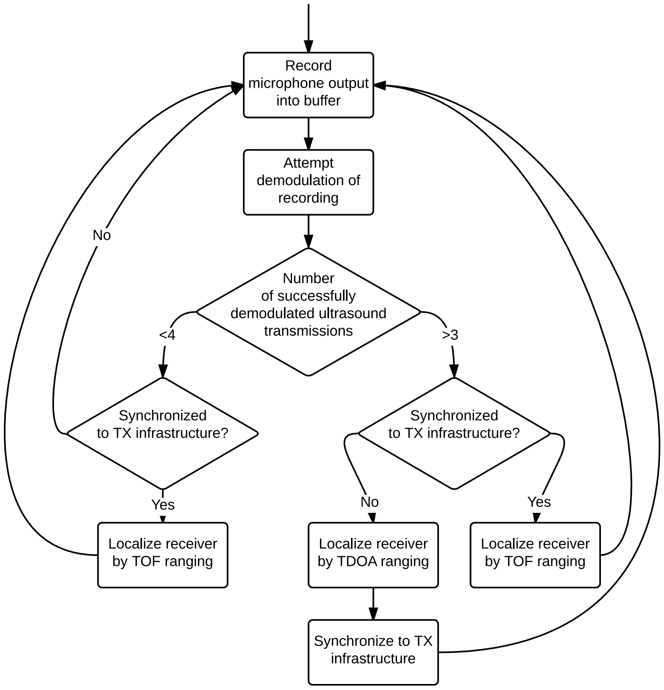

Record microphone output into buffer
Attempt demodulation of recording
Number of successfully demodulated ultrasound transmissions
<4
>3
No
Synchronized to TX infrastructure?
Synchronized to TX infrastructure?
Yes
No
Yes
Localize receiver by TOF ranging
Localize receiver by TDOA ranging
Localize receiver by TOF ranging
Synchronize to TX infrastructure

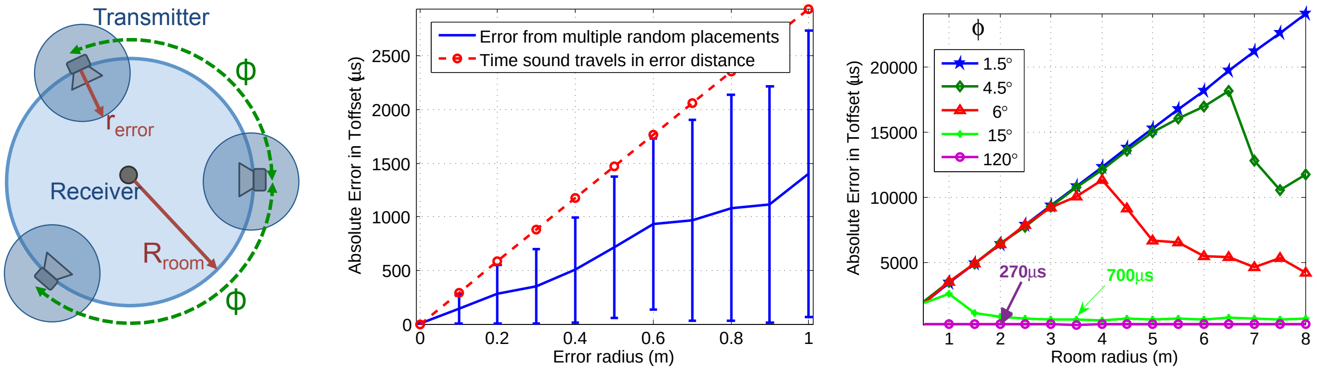
Fig. 12. Phone app flowchart

B. Software Design

In this section we discuss the transmitter firmware as well as the receiver demodulation software.

1) Ultrasonic Beacon Firmware: Each *Ultrasonic Beacon* synchronizes with the *Network Master* over a single hop using its 802.15.4 radio. Once the beacon is synchronized, it plays back the ultrasonic signal stored in its flash memory at the beginning of its designated TDMA slot. The *Ultrasonic Beacon* hardware platform is based upon low-power, highly integrated components, which make the playback of ultrasound signals at a high sampling rate a non-trivial task. The platform is capable of playing approximately 1 second of 16-bit mono audio at a $125kHz$ sampling rate. Data is transferred between the MCU and the Audio Codec via a UART port running in SPI mode. The SPI's clock-line is routed into a counter on the MCU, which divides its frequency by 16. The SPI data, clock and the counter output mimic the data, clock and word clock lines of a left justified I^{2}S interface respectively. I^{2}S requires continuous dataflow during playback, therefore there is a tight time constraint on reading the waveform stored in flash memory and copying it to the SPI output register. Any additional computations that are required need to be carefully timed to ensure a continuous output. The limited flash memory of $256kB$ on the ATmega256RFR2 MCU is shared among the firmware and the ultrasonic signal. The signal is uncompressed and is loaded as multiple arrays of 16-bit integers into flash from a C header file that is populated by a MATLAB script.

2) iOS App: Figure 12 shows an overview of the receiver demodulation software that runs on iOS. It was prototyped in MATLAB before being ported to Objective-C. The algorithm for demodulating the ultrasonic transmissions and calculating the receiver's position was translated into C code using MATLAB's C coder. The iOS app continuously listens on the



(a) (b) (c)

Fig. 13. Deployment and geometry error (a) Model of transmitter and receiver placement in a room (b) Effect of measurement error in transmitter position (c) Effect of transmitter placement geometry

microphone and periodically passes filled audio buffers to the demodulator. Once the phone's position is determined using TDOA ranging and multilateraion, the app computes the TOF of the signals it has just received to determine the start of the previous TDMA cycle relative to the captured audio buffer. The audio sample index s_0 at which the TDMA cycle started is stored in memory and the phone is now synchronized to the transmission infrastructure. A counter keeps track of how many audio samples have been captured since s_0 and calculates successive TOF values of successfully demodulated ultrasonic packets based on their TOA in relation to s_0. Whenever the phone successfully demodulates enough ultrasonic packets to localize itself using TDOA it will resynchronize to the transmitters.

VI. Evaluation

In this section, we evaluate the effect of errors in deployment and measurement of beacon node locations on our ability to synchronize time which is equivalent to estimating the T_{offset} value defined in Section III-B. We also evaluate the sensitivity of transmitter placement. These two evaluations are performed through simulation given the model described in the section below. We then experimentally evaluate the timing accuracy and ranging capabilities of our platform.

A. Deployment and Geometry Error

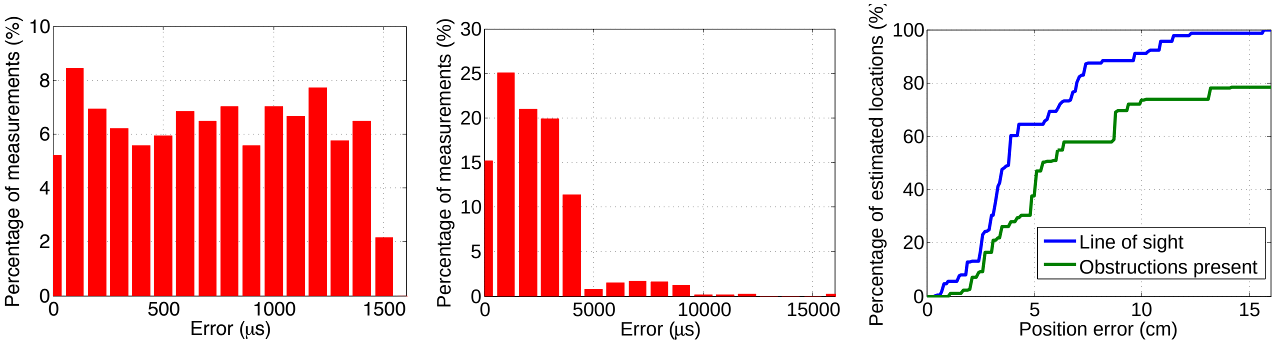
We assume the spatial configuration of transmitters and receivers in 2-D space as shown in Figure 13(a). The room is circular with a radius of R_{room}. For simplicity, the three transmitters are assumed to be placed in the periphery of the room. We assume some symmetry in placement as indicated by the separation angle ϕ between two pairs of transmitters. The measurement error r_{error} in transmitter position is assumed to be uniform in all directions around a transmitter and to be equal at all three transmitters. The true position of each transmitter is at a random location on the circle that defines possible deployment error. The receiver is placed in the center of the room and we assume that the speed of sound is constant at 340m/s.

1) Effect of incorrect measurements in transmitter positions: First, we evaluate the impact of measurement error in the transmitters' positions with R_{room} = 5m and ϕ = 120°. We then analyze the effect of r_{error} on T_{offset} by sweeping r_{error} from 0 to $1m$ and generating 400 random configurations of the transmitters on their error circles and estimating the T_{offset} for each configuration. The solid blue lines in Figure 13(b) show all possible values of errors. We see that the worst-case error for each r_{error} is bounded at the top by the time taken by sound to travel the same distance. Typically, while deploying this system with nominal care, the measurement error can be restricted to below $10cm$, which is equivalent to an error of $270\mu s$ in T_{offset} .

2) Effect of transmitter placement geometry and room size: Next, we evaluate the effect of room size and geometry of the placement by increasing R_{room} up to $10m$ and varying ϕ from the worst possible placement ($\phi = 0°$) to the best possible placement ($\phi = 120°$) geometry. We assume a value of $10cm$ for r_{error} from the previous section. Figure 13(c) shows the worst case error among 400 simulations at each data point. We see that for $\phi = 1.5°$ the error is quite high ($15ms$ for $R_{room} = 5m$) and grows proportionally to the room size. This is because the transmitters are almost at the same location, therefore the three unique TOA equations that we expect from the transmitters are identical, leading to insufficient information. The position of the receiver could be incorrectly estimated anywhere within the room, which is why we see the worst case error in T_{offset} growing linearly with the room size. For the best case geometry of $\phi = 120°$, we see that the error ($270\mu s$) is not dependent on the room size and is determined by r_{error}, except when R_{room} is small and comparable to r_{error}. This is because at $\phi = 120°$ the transmitters are sufficiently separated to provide the timing information required to estimate the receiver position and T_{offset}. For intermediate values of ϕ, we see that as the room size increases, the error increases with room size until the point where the room size is large enough to provide sufficient spatial separation to the transmitters. This can be seen when each line reaches a maximum and then decreases.

B. Synchronization and Ranging Performance

In order to evaluate the synchronization error of our method and the resulting positioning error, we set up four ultrasonic transmitters in a $4.5x5.5m$ area. Our ranging signals were generated using a MOTU Ultralite mk3 10 channel audio



(a) Time-sync accuracy LOS (b) Time-sync accuracy multi-path (c) Position estimation accuracy

Fig. 14. Acoustic time synchronization accuracy and position error

interface and received by an iPhone 5S and an Audix TM-1 measurement microphone, which were co-located on a microphone stand. The experimental setup can be seen in Figure 11. The second recording device allowed us to tightly time synchronize all transmitters as well as the measurement microphone to obtain ground truth TOF and position measurements. The iPhone synchronizes itself to the transmitters using our audio synchronization method and recorded its calculated position and TOF values. We perform measurements at 20 random locations in the room, for which 60 samples were recorded each. A mean absolute time synchronization error of $720\mu s$ with a maximum of $1484\mu s$ was achieved. This resulted in a maximum absolute distance error of $15.6cm$. Figure 14(c) shows the Cumulative Distribution Function of positioning error resulting from time synchronization using our method. 98% of the samples exhibit a positioning error of less than $12cm$ and 100% show an error of $16cm$ or less in LOS conditions. This small-scale experiment validates the concept that highly accurate clock synchronization is possible through audio sampling.

Further, we conducted an experiment where one or two transmitters (depending on the position of the receiver) were obstructed by a large white board. In this case, approximately 78% of the calculated position samples were within an absolute distance error of less than $14cm$ (see Figure 14(c)) and a maximum error of $4.16m$ was encountered. The large maximum error is due to the phone being unable to measure its position accurately due to multipath or severely attenuated ranging signals, causing it to synchronize erroneously to the transmitting infrastructure when placed in certain locations. Figure 14(b) shows that this is reflected in a maximum absolute time synchronization error of $39.1ms$ with a mean of $2.33ms$. These points could be eliminated if we received one good set of samples and then performed TOF ranging. We purposefully disabled this capability to highlight the impact of obstructions on TDOA localization. One could also apply filtering to the recorded signals by thresholding and sampling over multiple TDMA cycles to obtain an accurate lock on the receiver's position is therefore advisable in a real-world scenario. A software-based Phase-Loop Lock (PLL) controller that slowly adjusts the clock based TDOA inputs when available rather than immediately resetting the timing offset on each sample can prevent isolated erroneous position measurements from causing significant time synchronization errors.

.

VII. Limitations and Future Work

Although promising, there are a few limitations to this approach. In 3-D space, our system relies on the receiver being in LOS of at least four transmitters in order to synchronize. However, it is possible that obstructions inside the building or the person holding the mobile device block one or more transmitters. We experimentally studied the effect of this as shown in Figure 14(b) and Figure 14(c). It is also possible that in certain areas such as long corridors, fewer than three transmitters are present. In these cases, we can utilize the inertial sensors on the phone to track the mobile device using pedestrian dead reckoning [29]. Another challenge that we are currently working on is distinguishing between multipath and LOS signals. The presence of a multipath signal in the absence of a direct LOS signal could result in a TOF measurement which is equivalent to the receiver being located at a much larger distance away from the transmitter. We are exploring the possibility of utilizing the building layout and geometry, as well as incorporating additional sensors for localization such as Bluetooth Low Energy or Visible Light Communication [30] to overcome this problem.

VIII. Conclusions

In this paper, we present a platform for indoor ranging and localization of mobile devices. The platform consists of a hardware module that can be wirelessly triggered to simultaneously transmit ultrasonic chirps in the frequency ranges just outside of human hearing that are still detectable by smartphones. The beacons are synchronized from an external GPS time source and designed to transmit modulated ultrasonic ranging and data signals with low levels of jitter. When three or more transmitters are available, the system can synchronize the smartphone with global time allowing direct TOF measurements while the clock remains stable. This improves the system in terms of robustness when transmitters are blocked, as well as installation flexibility by reducing node density requirements. In practice, we are able to synchronize the free running audio sampling clock to an average of $720\mu s$ and remain synchronized with drift rates on the order of $2m$ per hour.

The platform also provides an ultrasonic microphone input that can be used for beacon-to-beacon ranging. This additional capability can be used in the future to help automatically configure beacon locations. The combination of inter-beacon and TOF ranging will enable approaches like simultaneous localization and mapping of location infrastructures as devices move through indoor environments.

IX. Acknowledgements

This research was funded in part by the Bosch Research and Technology Center in Pittsburgh and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

[1] Apple Inc. iBeacon for Developers: https://developer.apple.com/ibeacon (viewed 2/13/2015).

[2] Gimbal Inc. Gimbal: http://www.gimbal.com (viewed 2/13/2015).

[3] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (Mobicom '00)*, pages 32–43, New York, NY, USA, 2000. ACM.

[4] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 99–112, New York, NY, USA, 2012. ACM.

[5] B.W. Parkinson and S.W. Gilbert. Navstar: Global positioning system - ten years later. *Proceedings of the IEEE*, 71(10):1177 – 1186, oct. 1983.

[6] Gaetano Borriello, Alan Liu, Tony Offer, Christopher Palistrant, and Richard Sharp. Walrus: wireless acoustic location with room-level resolution using ultrasound. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pages 191–203, New York, NY, USA, 2005. ACM.

[7] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, volume 2, pages 775 –784, 2000.

[8] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42 –47, oct 1997.

[9] Konrad Lorincz and Matt Welsh. Motetrack: a robust, decentralized approach to rf-based location tracking. In *Proceedings of the 1st International Conference on Location- and Context-Awareness (LoCA'05)*, pages 63–82, Berlin, Heidelberg, 2005. Springer-Verlag.

[10] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pages 155–168, New York, NY, USA, 2011. ACM.

[11] Isaac Amundson and Xenofon D. Koutsoukos. A survey on localization for mobile wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments (MELT '09)*, pages 235–254, Berlin, Heidelberg, 2009. Springer-Verlag.

[12] Kaveh Pahlavan, Xinrong Li, Mika Ylianttila, Ranvir Chana, and Matti Latva-aho. An overview of wireless indoor geolocation techniques and systems. In *Proceedings of the IFIP-TC6/European Commission International Workshop on Mobile and Wireless Communication Networks (NETWORKING '00)*, pages 1–13, London, UK, UK, 2000. Springer-Verlag.

[13] Zheng Sun, R. Farley, T. Kaleas, J. Ellis, and K. Chikkappa. Cortina: Collaborative context-aware indoor positioning employing rss and rtof techniques. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM '11 Workshops)*, pages 340 –343, march 2011.

[14] Chunyi Peng, Guobin Shen, Zheng Han, Yongguang Zhang, Yanlin Li, and Kun Tan. A beepbeep ranging system on mobile phones. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pages 397–398, New York, NY, USA, 2007. ACM.

[15] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp '02)*, pages 264–280, London, UK, UK, 2002. Springer-Verlag.

[16] Mike Hazas and Andy Ward. A high performance privacy-oriented location system. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PERCOM '03)*, pages 216–223, Washington, DC, USA, 2003. IEEE Computer Society.

[17] Michael McCarthy, Paul Duff, Henk L. Muller, and Cliff Randell. Accessible ultrasonic positioning. *IEEE Pervasive Computing*, 5(4):86–93, October 2006.

[18] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P. Martin. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*, pages 97–108, New York, NY, USA, 2011. ACM.

[19] D. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 1991.

[20] Zatti S. Gusell, R. The accuracy of clock synchonization achieved by tempo. *IEEE transactions on Software Engineering*, 1989.

[21] F. Cristian. Probabilistic clock synchronization. *Distributed Computing*, 1989.

[22] Ochsenreiter W. Kopetz, H. Clock synchonization in distributed real-time systems. *IEEE Computer*, August 1987.

[23] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pages 147–163, New York, NY, USA, 2002. ACM.

[24] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 138–149, New York, NY, USA, 2003. ACM.

[25] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 39–49, New York, NY, USA, 2004. ACM.

[26] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the 10th IEEE International Conference on Information Processing in Sensor Networks*, IPSN '11, pages 73–84, April 2011.

[27] Clark, Nick. AUD1 Assistive Listening Software: http://www.aud1.com/ (viewed 2/13/2015).

[28] Elliott Kaplan and Christopher Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005.

[29] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304. ACM, 2012.

[30] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 249–260. IEEE Press, 2014.