

3D Localization for Sub-Centimeter Sized Devices

Rajalakshmi Nandakumar, Vikram Iyer and Shyamnath Gollakota
University of Washington
(rajaln,vsiyer,gshyam)@uw.edu

ABSTRACT

The vision of tracking small IoT devices runs into the reality of localization technologies — today it is difficult to continuously track objects through walls in homes and warehouses on a coin cell battery. While Wi-Fi and ultra-wideband radios can provide tracking through walls, they do not last more than a month on small coin and button cell batteries since they consume tens of milliwatts of power. We present the first localization system that consumes microwatts of power at a mobile device and can be localized across multiple rooms in settings like homes and hospitals. To this end, we introduce a multi-band backscatter prototype that operates across 900 MHz, 2.4 and 5 GHz and can extract the backscatter phase information from signals that are below the noise floor. We build sub-centimeter sized prototypes which consume 93 μ W and could last five to ten years on button cell batteries. We achieved ranges of up to 60 m away from the AP and accuracies of 2, 12, 50 and 145 cm at 1, 5, 30 and 60 m respectively. To demonstrate the potential of our design, we deploy it in two real-world scenarios: five homes in a metropolitan area and the surgery wing of a hospital in patient pre-op and post-op rooms as well as storage facilities.

CCS CONCEPTS

- Information systems → Sensor networks; Global positioning systems;
- Hardware → PCB design and layout;

KEYWORDS

Wireless Localization; LoRa Backscatter; smart dust; IoT

ACM Reference Format:

Rajalakshmi Nandakumar, Vikram Iyer and Shyamnath Gollakota . 2018. 3D Localization for Sub-Centimeter Sized Devices. In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18), November 4–7, 2018, Shenzhen, China*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3274783.3274851>

1 INTRODUCTION

Recent years have seen significant advances in wireless localization [37, 40]. However existing solutions do not meet the requirements for size-constrained IoT applications. Fig. 1 shows battery life of common radio technologies like BLE, LoRa, ultra-wideband (UWB) and Wi-Fi, each running at a 1% duty cycle with small coin

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SenSys '18, November 4–7, 2018, Shenzhen, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5952-8/18/11...\$15.00
<https://doi.org/10.1145/3274783.3274851>

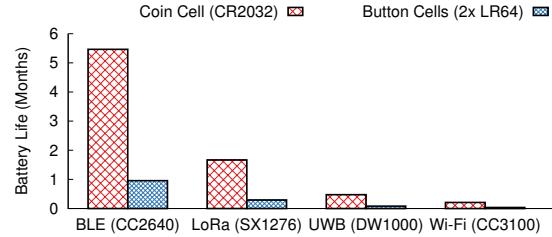


Figure 1: Radio localization battery life. Battery life estimates for different technologies operating at 3 V from coin and button cell batteries running at 1% duty cycle. §5 provides a comprehensive power consumption comparison.

and button cell batteries for equal comparison. The shorter battery life limits the adoption of tracking solutions based on these radio technologies by making them inconvenient for consumer applications and infeasible for large scale commercial deployments. Requiring large batteries on the other hand prevents scaling down the size of IoT devices. While RFID tags are attractive from a power and size perspective, they have a limited range and do not work consistently through walls and other barriers. Consumers often deploy devices in rooms throughout homes, and similarly commercial deployments in settings like hospitals require covering multiple patient rooms with a variety of obstructions and walls. Achieving localization in these scenarios would therefore require readers in every room which significantly increases deployment cost.

This paper presents *μLocate*, the first wireless localization system that consumes microwatts of power at the mobile IoT devices and can be localized through walls in settings like homes and hospitals. Our design can achieve 3D localization capabilities while supporting IoT devices that can be scaled to sub-centimeter form factor. To achieve this, we design a backscatter based solution that satisfies all of the above requirements. Specifically, we make the following hardware and systems contributions:

- We design and build a *sub-centimeter sized IoT platform that supports low-power localization capabilities*. Our platform integrates a low-power microcontroller and RF switch for backscatter rather than an active radio, as well as all required off chip passive components and antennas. We custom fabricate flexible circuits using laser micro-machining techniques and use an off the shelf microcontroller available in an ultra-miniature 2 mm × 1.5 mm package to achieve the small form factor. By utilizing highly integrated commercially available microcontrollers we minimize the size of the assembly.
- We achieve low-power long-range backscatter through walls by building on recent work on LoRa Backscatter [35]; however, this prior work requires implementing complex computation to perform chirp spread spectrum (CSS) coding on an FPGA platform, which

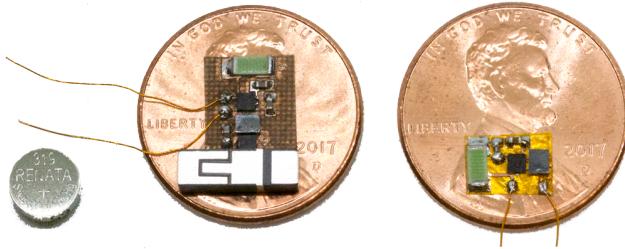


Figure 2: μ Locate prototypes. Our miniaturized prototypes require two button cell batteries (left), which are as small as 5.8 mm in diameter. Our multi-band prototype based on the KL03 microcontroller is $11.8 \times 7.5 \times 2.1$ mm and includes chip antennas for 900 MHz, 2.4 GHz, and 5 GHz. Our 5 GHz prototype (right) measuring $7.2 \times 5.1 \times 0.5$ mm, is designed to operate at shorter ranges and in an even smaller form factor. The prototypes are placed on a US penny for scale.

consumes around 5–10 mW using off the shelf components. We present *a novel backscatter architecture that enables CSS backscatter using off-the-shelf microcontrollers at significantly lower power*. Specifically, because these microcontrollers lack the capability to easily implement the complex CSS coding, we instead delegate this coding to the access point, which transmits the CSS signal. By doing this, our low power microcontroller simply needs to run an oscillator to frequency shift the CSS signal and encode data using ON-OFF keying in reflections.

- Since the localization accuracy is directly proportional to the signal bandwidth, we design and build *a novel multi-band backscatter hardware system that can concurrently operate across the ISM bands at 900 MHz, 2.4 GHz and 5 GHz*. Specifically, the access point (AP) transmits signals across these frequencies which are backscattered by our device. We combine the received signals across all of these frequencies to disambiguate between the multipath reflections and extract the direct line-of-sight path.

To summarize, our system works as follows: the AP, which is placed at a known location, transmits a 500 kHz chirp spread spectrum signal. The IoT device uses a low-power microcontroller to shift this signal by 1–2 MHz and backscatter it back to the AP. The AP then extracts the phase information from the weak backscattered signals that are below the noise floor. It repeats this process concurrently across the 900 MHz, 2.4 GHz and 5 GHz bands and combines the phase information to disambiguate the multipath in the environment.

Implementing this system introduces the following three algorithmic challenges: First, in contrast to direct radio signals that attenuate as d^2 , backscatter signals attenuate as d^4 . As a result, we need a way to extract the phase from backscattered signals which are below the noise floor at long distances. Second, our IoT devices use small, low-power microcontrollers to shift the chirp spread spectrum signal from the AP. This introduces frequency and sampling offsets in the weak backscattered signals that have to be corrected to accurately estimate phase. Third, querying all the 500 kHz bands sequentially across all the ISM bands requires a total of 572 frequencies which takes more than four seconds, introducing

a significant delay overhead. Concurrently querying all these 572 frequencies requires the AP to proportionally reduce the power at each of the frequencies to be compliant with FCC regulations; this in turn would significantly reduce the range of our system.

In the rest of the paper, we address the above challenges and build multiple prototypes of our design shown in Fig. 2. We build our prototypes using commercial off-the-shelf components including switches, microcontrollers and chip antennas. Our first prototype uses the 2.0×1.6 mm Kinetis KL03 microcontroller with a 2.4 GHz and 900 MHz dual band chip antenna along with a 5 GHz chip antenna. We miniaturize our second prototype using a laser micro-machining method that produces flexible circuits. We use the Kinetis microcontroller with only the 5 GHz chip antenna that limits the range but enables a further miniaturized device. We also present an ASIC design for our multi-band backscatter approach to further reduce size and power.

Our evaluation shows the following:

- **Power.** Our off-the-shelf hardware and ASIC consume $93 \mu\text{W}$ and $5 \mu\text{W}$ respectively. This translates to an expected operational lifetime of 5–10 years of duty cycled operation on small, 5.8 mm diameter button cell batteries for our off-the-shelf microcontroller hardware and ASIC prototypes.
- **Accuracy.** We demonstrate 3D localization accuracy which scales with the distance. Our system gives localization errors of 2, 12, 50, and 145 cm at 1, 5, 30 and 60 m respectively between the AP and our backscatter devices.
- **Latency.** Finally, across distances up to 60 m from an AP, our algorithm can compute the location values using between 9–28 frequencies, which translates to a latency of 25–70 ms.

In addition to the above benchmarks which characterize our system performance, we also deploy the system in the following two real world scenarios:

- Five homes in a metropolitan area including three single-story apartments and two multi-story townhouses. We select a variety of locations and orientations across different rooms, behind closed doors, in closets, on shelves, and even hidden in couches to determine whether our system can localize objects across an entire home to enable item tracking applications.
- Surgery wing of a hospital including patient pre-op and post-op rooms as well as storage facilities. We run experiments in various locations for tracking mobile equipment such as on IV poles and vital signs monitors that travel with patients between different rooms.

Our key contributions are in the system design and sub-centimeter sized hardware prototypes we create – which together enable the first microwatt power localization system for mobile devices that operates through walls. Finally, we also note that instead of retrofitting localization into existing Wi-Fi and RFID protocols, we take a clean-slate approach to the problem of localization that does not require a radio chipset. We believe that since IoT standards are still in flux, a clean-slate design that enables localization would make an excellent candidate for adoption.

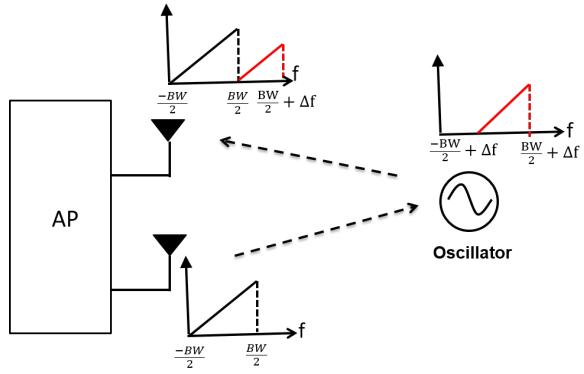


Figure 3: Low-power delegating architecture. The AP transmits a chirp spread spectrum signal with a bandwidth BW to the IoT device with an oscillator and RF switch. The switch backscatters the coded signal back to the AP with a frequency shift of Δf .

2 SYSTEM DESIGN

Our design has three key components: 1) our low-power architecture which delegates the coding operation to the access point, allowing us to decode backscatter signals at large distances using sub-centimeter sized devices, 2) our phase-extraction algorithm that can extract the phase from signals below the noise floor and 3) our online search algorithm that dynamically queries a different set of frequencies given the signal quality to reduce latency. In the rest of this section, we describe each of these components.

2.1 Low-Power Delegating Architecture

The goal of our architecture is to enable localization at long ranges with very small, low power backscattering IoT devices. To achieve this we have to address two competing requirements: 1) since backscattered signals are orders of magnitude weaker than active radio transmissions, to achieve long range we need to perform significant amounts of coding, and 2) in order to be compatible with off-the-shelf low-power microcontrollers, the IoT device design must be limited to simple operations.

To achieve this we delegate the complex coding operations to the AP. Our design works as shown in Fig. 3. The μ Locate AP transmits a coded narrowband signal while the μ Locate IoT device simply backscatters this signal transmitted by the AP with a frequency shift of 1–2 MHz. This shifting can be achieved using the built in oscillators on commercially available microcontrollers (see §3). Our architecture therefore eliminates the need for an active radio on the μ Locate device.

Shifting the signal on the IoT device has three key advantages: 1) It helps distinguish the backscattered signal from the direct signal transmitted by the AP in the frequency domain, allowing the receiver to easily decode it. More importantly, at the shifted frequencies, the receiver effectively receives a coded backscattered signal that it can use to extract the phase required for localization. Furthermore, we achieve this without requiring the IoT device to perform the complex coding operations itself, allowing for centimeter scale low-power implementations without a custom ASIC. 2) by shifting the signal to different frequencies, multiple tags can

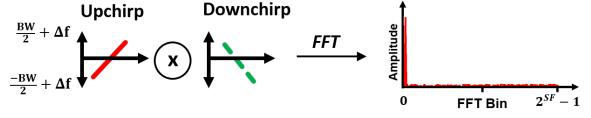


Figure 4: Decoding the weak backscatter signal. The receiver correlates the received upchirp with a downchirp. In the absence of CFOs and timing offsets, taking an FFT of this correlation produces a peak in the first FFT bin.

co-exist and be localized simultaneously using a single AP, and 3) by using ON and OFF keying at the backscatter device, one can also enable data communication in addition to localization.

What kind of coding do we use at the AP? The objective here is to pick the coding scheme that can be used to decode the phase of backscatter signals that are far below the noise floor. To this end, we use chirp spread spectrum as our coding mechanism. In chirp spread spectrum, we transmit a signal with a linearly changing frequency over bandwidth (BW) varying between $-\frac{BW}{2}$ to $\frac{BW}{2}$. Chirp signals have the following advantages that make them the best fit for our application: First, in comparison to phase, amplitude and discrete frequency-shift modulation, chirp spread spectrum (CSS) achieves an efficient trade-off between bandwidth and decoding capability, when the signal is drowned by noise [3]. Further, it is resilient to both in-band and out-of-band interference [1], and 2) unlike direct-sequence spread-spectrum that requires complex synchronization and has a long acquisition time when the signal is below the noise floor [5, 28], CSS receivers have comparatively lower-complexity and significantly shorter acquisition times [1].

Specifically, we choose a narrow BW of 500 kHz where the chirp duration, T is 7 ms which we find balances accuracy and latency well. The receiver at the AP samples these signals at 1 MHz. In the next few sections, we first describe, how to estimate the phase from CSS signals. We then show how to selectively query across the 900 MHz, 2.4 GHz and 5 GHz ISM bands to disambiguate the multipath, estimate the range, and achieve 3D localization. Finally, we describe how we can achieve real-time tracking.

2.2 Below-Noise Backscatter Phase

Assume that the AP, which is placed at a known location, is separated from the IoT device by a distance of d . When the AP transmits the chirp signal it propagates a total distance of $2d$ including the time it takes for the backscattered signal from the IoT device to arrive back at the AP. The wireless channel of such a signal is,

$$h = ae^{-2\pi f \frac{2d}{c}} \quad (1)$$

Here a is the attenuation, f is the frequency at which the signal is being transmitted and c is the speed of RF signals in the medium. At a high level, if we can extract the phase of the backscattered signal at a specific frequency, we can estimate the range, d .

Thus, if the AP transmits a tone at a single frequency f , in the absence of multipath, the phase of the backscatter signal can be used to estimate the range. However such single tone signals (e.g., RFID) have a limited range in the context of backscatter communication and hence cannot achieve the long ranges that are required for IoT localization. As described earlier the AP instead transmits a

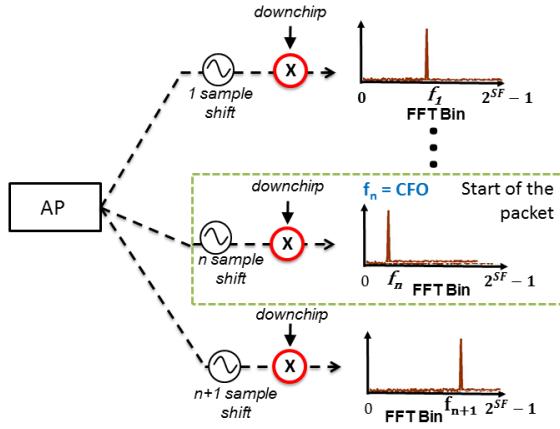


Figure 5: Estimating the CFO and the chirp start. To correct for the CFO and timing offsets, the AP iterates through different timing offsets until it can separate it from the CFO.

linear frequency modulated chirp pulse which allows our system to operate at longer ranges without further amplifying the signal. As shown in Fig. 3, the tag then shifts this chirp signal by a frequency Δf , and the shifted signal is received back at the AP. Hence the receiver receives a chirp signal whose frequency varies from $-BW/2 + \Delta f$ to $+BW/2 + \Delta f$.

We use correlation to decode this signal. Specifically, the receiver first correlates the received signal with a downchirp, a signal where the frequencies linearly decrease from $+BW/2 + \Delta f$ to $-BW/2 + \Delta f$. This downchirp is synthesized on the receiver. During the multiplication step of the correlation, the linear change in the frequency between the receiver upchirps and the synthesized downchirps cancel each other out. During the addition step of the correlation, we effectively sum the energy across all the chirp frequencies providing coding gain and allowing us to decode the backscatter signals below the noise floor. This decoding process is illustrated in Fig. 4.

Extracting the channel phase information from this signal requires us to address three challenges: 1) Since the chirp signal is spread across frequencies, we do not get the phase at a single frequency but rather the chirp phase which is a combination of phases across all the frequencies in the chirp, 2) to decode and estimate the phase of the signal we need to accurately estimate the beginning of the backscatter chirp, and 3) our small low-power microcontrollers which shift the incoming chirp by Δf introduce an unknown carrier frequency offset (CFO) between the AP and the IoT device that changes the phase of the received signal.

To address the above challenges, we first describe how we jointly estimate the carrier frequency offset (CFO) and correct for the start of the backscatter chirp. We then show how to compute the channel phase information given the phase of the backscattered chirp.

Step 1. Estimating beginning of chirp and CFO. To decode and estimate the phase of this signal, we first need to estimate the beginning of the chirp. For this, we leverage a key property of the chirp signal – a time delay in the chirp signal translates to frequency shift. Specifically, as shown in Fig. 4, we first multiply the received signal with a downchirp whose frequency varies from

$+BW/2 + \Delta f$ to $-BW/2 + \Delta f$ and then take an FFT of this signal. When we perform an FFT, in the ideal case with no offset from the beginning of the chirp, our receiver sees a peak in the FFT in the first bin. However when the receiver has a timing offset, the peak appears in the shifted bin.

The problem however is that this alone is not sufficient to estimate the beginning of the backscatter chirp. Specifically, since our microcontroller is not synchronized with the AP, it introduces an additional carrier frequency offset (CFO). This results in the backscatter device shifting the chirp signal by $\Delta f + CFO$ instead of Δf which causes an additional frequency shift of CFO when the receiver performs the FFT.

Thus, we need to disambiguate between the frequency shift in the FFT caused due to CFO versus a timing offset in the estimation of the start of the backscatter chirp. To estimate the beginning of the chirp signal in the presence of CFO, we first multiply the received signal with a downchirp and then take an FFT of this signal. Ideally, if the signal is aligned with its beginning and there is no CFO, the result of this multiplication will be a constant signal with a peak at index zero in its FFT. In practice however, the peak will be shifted to a frequency f_0 which is the sum of the frequency offsets, f_{to} and f_{CFO} , caused by the timing misalignment of the received chirp and the CFO respectively. Now if we move the beginning of our chirp one sample closer to its real beginning and repeat the above operation, there will be a new peak at frequency $f_1 = f_0 - 1$. This change occurs only due to the change in the frequency offset caused by the timing misalignment of the chirp, since the CFO remains constant in the above operation.

As we see in Fig. 5, if we iterate over these operations, at some iteration n , the new peak f_{n+1} will become greater than f_n . This is because at f_n the frequency offset due to misalignment of the transmission becomes zero and the only residual frequency offset left is due to the CFO. In the next operation, the signal will again be misaligned and f_{n+1} will be greater than f_n . At this point, we stop this operation and the symbol at instant n is the beginning of the transmission. The residue f_n is then the CFO. We then apply this CFO to the rest of the transmission to estimate the right chirp phase.

Step 2. Extracting channel phase from the chirp phase. When the AP correlates the received backscattered upchirp with a downchirp, it provides coding gain but also effectively adds the phase across all the frequencies in the chirp. Thus, the phase of the chirp extracted after correcting for the timing offset and CFO is the sum of the phases at each of the frequencies in the chirp,

$$\phi_{chirp} = \phi_{-\frac{BW}{2} + \Delta f} + \dots + \phi_{\frac{BW}{2} + \Delta f}$$

If we assume that the channel path of the signal remains constant throughout the duration of the chirp¹, then the phase of the received signal varies linearly with respect to frequency. Said differently, from Eq. 1, we see that the phase of the received signal changes linearly with the frequency. Thus, we can rewrite the above equation as,

$$\phi_{chirp} = \phi_{-\frac{BW}{2} + \Delta f} + \phi_{-\frac{BW}{2} + \Delta f} \frac{\lambda_{-\frac{BW}{2} + \Delta f}}{\lambda_{-\frac{BW}{2} + \Delta f + 1}}$$

¹This is a reasonable assumption since a chirp duration is 7 ms.

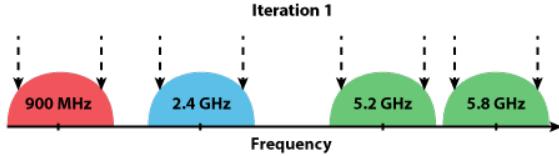


Figure 6: Dynamic frequency selection. In the first iteration we start with the frequencies that are most separated. This translates to the frequencies at the edges of the three ISM bands.

$$+ \dots + \phi_{-\frac{BW}{2} + \Delta f} \frac{\lambda_{-\frac{BW}{2} + \Delta f}}{\lambda_{\frac{BW}{2} + \Delta f}}$$

Here λ_s are the wavelengths corresponding to the different frequencies in the chirp. Solving the above equation we can compute $\phi_{-\frac{BW}{2} + \Delta f}$ which is the channel phase at the beginning frequency corresponding to the chirp which we can then use to estimate the range using Eq. 1.

2.3 Multipath disambiguation

In practice, due to multipath, the obtained phase is actually the sum of phase of the direct line of sight signal and the phase of the different multipath reflections. Hence, to accurately localize the IoT device, we should disambiguate the various multipath reflections from the direct line of sight signal. To achieve this, we dynamically send chirps across the ISM bands in 900 MHz, 2.4 GHz and 5 GHz and estimate the phase of each frequency using the above method. At a high level, by combining the phase information across all these frequencies we can simulate an ultra-wide band transmission that can be used to disambiguate the multipath in the environment. Since our chirps are 500 kHz wide, we can transmit multiple chirps in adjacent bands across the three ISM bands. This however would significantly increase the latency of localization. Specifically, there are bandwidths of 26 MHz, 80 MHz and 180 MHz in the 900 MHz, 2.4 and 5 GHz ISM bands. Dividing them into 500 kHz intervals results in 572 chirps across all these bands which translates to 4 seconds using a 7 ms chirp. We instead design a dynamic frequency selection algorithm that significantly reduces the number of queried frequencies by 10-20x.

Dynamic Frequency selection. Our algorithm is based on three key principles.

- We determine the ISM bands that can be used depending on the distance of the IoT device. Specifically, signals at 5 GHz have very high attenuation and can be decoded using the above method only if the IoT device is in the same room as the AP. Similarly signals in the ISM band of 2.4 GHz experience a lower attenuation compared to 5 GHz but have a higher attenuation than 900 MHz. Thus we can prune a number of frequencies depending on the attenuation observed from the IoT device. Specifically, we first send a chirp in the 2.4 GHz band and determine the SNR of the chirp. If the SNR is very high, the device is at a short distance from the AP and hence all three ISM bands can be used. Otherwise, only the 900 MHz band and 2.4 GHz band can be used to estimate the distance of the IoT device. When the receiver cannot decode the initial chirp, we use only the 900 MHz band.

Algorithm 1 Dynamic Frequency Selection

```

1: min_bands,max_bands           ▷ Min and max of available bands
2: range = 0
3: function QUERY(min_bands,max_bands,range)
4:   newrange = Range_estimate(min_bands ∪ max_bands)
5:   if newrange – range < threshold then
6:     return newrange
7:   if newrange – range > threshold then
8:     for i in 1..length(min_bands) do
9:       if max_bandsi ≤ min_bandsi then
10:        midi = ∅
11:       else
12:         midi =  $\frac{\min_bands_i + max_bands_i}{2}$ 
13:       if mid = ∅ then           ▷ No more frequencies available
14:         return newrange
15:       min_bands = mid_bands ∪ mid
16:       max_bands = mid ∪ max_bands
17:     return QUERY(min_bands,max_bands,newrange)
18: function Range_estimate(frequency1..n)
19:   phase1..n = ESTIMATEPHASE(frequency1..n)           ▷ §2.2
20:   Channel = DFT(frequency1..n,phase1..n)
21:   peaks = FINDPEAKS(Channel,prominencethreshold) return peaks

```

- Each new frequency provides a new linear equation for the multipath combination at that frequency. However picking two frequencies that are next to each other does not provide independent equations since the amplitude and phases of different multipaths are similar for adjacent frequencies. Thus, querying frequencies that are separated by the highest bandwidth provides more useful information than querying adjacent frequencies. Thus, we can reduce the number of frequencies that are queried by picking them such that the gap between them is maximized.
- The backscatter device can reflect signals simultaneously across multiple frequencies. Thus, the AP can concurrently transmit four coded signals centered at frequencies f_1, f_2, f_3 and f_4 , and the backscattered phase can be decoded at all these frequencies. This is used to parallelize the number of queries. We note however that requiring the AP to transmit multiple concurrent frequencies in the ISM band requires us to reduce the power on each of these frequencies proportionally to comply with FCC regulations. This would reduce the range of our system. We instead use the following rule to opportunistically parallelize our transmissions: if the signal can be decoded at 2.4 GHz then the 900 MHz is much stronger and hence we can query four frequencies concurrently at 900 MHz. Similarly if the signal can be decoded at 5 GHz, we can query four frequencies concurrently at 2.4 GHz. Finally, if the signal strength is strong at any of these ISM bands, we increase the number of concurrent frequencies in that ISM band in the next round.

Using the above principles, we can design a binary search algorithm shown in Alg. 1. Specifically, once we identify the ISM bands that can be used, the AP first sends a chirp at the minimum and the maximum frequencies of the chosen bands as shown in Fig. 6. To improve the distance resolution, the next frequency to query is picked using a recursive binary search function that chooses

frequencies at the extremes of the spectrum. After each query, the receiver computes the new distance estimate by using an inverse FFT on the phases at all the queried frequencies to get the time-domain multi-path profile. By using a fixed energy threshold over this profile, we identify the closest (and therefore most direct) path from the device.

The length of each chirp is 7 ms. Hence the total latency would linearly increase by 7 ms for every new set of concurrent frequencies queried. This will lead to very high latency of 4 s if all the frequencies in the three bands are queried sequentially to localize the IoT device. Further the localization error increases if there is a change in the environment (like a person walking) during this period. Our solution is to modify the algorithm to query a new set of frequencies only when the difference between the current location and the previous location estimates is less than a threshold, ϵ . Said differently, we can stop querying after the algorithm converges to a specific 3D location. The challenge is to choose this threshold after which we end the algorithm. Since the range of our system is 60 m, a lower value for the fixed threshold for ϵ like 1 cm means that the system will not converge when the device is at a larger range thereby querying all frequencies. On the other hand, a higher threshold value for ϵ of say 10 cm will increase the localization error when the IoT device is at a closer range. To avoid this, we use a threshold that dynamically varies depending upon the approximate range of the IoT device. This leverages the intuition that the localization error increases with the distance between the AP and IoT device. Thus, we set the threshold to be a function of the distance, $\epsilon(d)$. To this end, we first measure the received power of the backscattered signal to estimate an approximate distance value. We then fix a threshold based on this range. We pick the threshold values empirically as described in §4.1.

Least squares estimation. Finally, to determine the 3D location of the device, we use three antennas at the AP and measure the range of the device with respect to each antenna. We note however that using simple triangulation of all distances is not optimal for 3D localization. Specifically, because of errors in distance estimation from each of the antennas, it is likely that the spheres centered at the three antennas will either not intersect or intersect at multiple points. To address this, we use a non-linear least squares method [23, 37] that is typically employed to determine the 3D location using the estimation ranges from the above method.

Specifically, say A_1, A_2, A_3 are the locations of the antenna and d_1, d_2 and d_3 are the range estimations obtained from the above method. Now the 3D location L of the device is obtained by minimizing the sum of least square distance errors:

$$\min\{|A_1 - L| - d_1; |A_2 - L| - d_2; |A_3 - L| - d_3\}$$

These three equations are sufficient to solve for the three co-ordinates. However, the accuracy of this method increases by increasing the number of antennas and subsequently the number of range estimations. Additionally, to improve the accuracy of our system, we add a tight min and max bound for each of the location co-ordinates of the system. These bounds used correspond to the maximum dimensions of the experimental space. Because our access point remain in a fixed location, in realistic deployments we can use the

floorplan of a room to determine these bounds. This helps the least squares method converge within a few iterations, despite using a limited number of antennas with small separation.

3 SYSTEM IMPLEMENTATION

3.1 AP Prototype

The AP has both a transmitter and receiver that are co-located, i.e., we use a mono-static deployment. The transmitter is implemented using an N210 USRP with a UBX160 daughterboard. The output of the USRP is connected to three different power amplifiers, one each for 900 MHz, 2.4 and 5 GHz using a power divider. We configure the USRP to transmit at power levels such that each amplifier outputs 30 dBm when transmitting in order to stay within FCC limits. Each amplifier is then connected to a 3 dBi antenna for 900 MHz and 6 dBi antennas for both 2.4 GHz and 5 GHz. We mount each of the three antennas to a metal pole separated by an average distance of 30 cm. The antennas are placed in different planes to have resolution across all the three axes.

The receiver on the AP is implemented using three N210 USRPs each with a UBX160 daughterboard. The output of each USRP is connected to a multi-band antenna capable of operation in all three bands. The transmitter and the receiver USRPs are synchronized using a common external clock and frequency reference. We note that the USRPs still have a fixed phase offset despite the common clock due to differences in cable lengths and sampling; however the external clock prevents drift over time therefore requiring a one time calibration to determine the fixed phase offset on start up.

Since the bandwidth of the UBX160 daughterboard is only 160 MHz, we shift the carrier frequency of the USRPs to transmit across the three ISM bands. We note that the USRP N210 can change its carrier frequency without turning off and maintains the same carrier frequency offset and phase offset throughout the measurement without additional calibration between frequency shifts. We connect all three receivers to a single host computer using a Gigabit Ethernet switch in order to synchronize the changes in carrier frequency. We use software radios in order to maximize flexibility during testing, however note that a future implementation could easily improve on this design. Specifically, all three receivers could use the same local oscillator (LO) for down conversion thereby guaranteeing synchronization. Additionally, switching frequencies on an SDR is limited by the time it takes for the PLL to lock and generate a stable output at each frequency. By using separate, continuously running LOs for each frequency the delay can be significantly reduced. We use the USRP receiver to digitize the raw radio signals and process these in software using the algorithms described above.

3.2 Device prototypes

Off-the-shelf prototypes. We build three different prototypes. The first uses the DE0-Nano FPGA development board to control an RF switch. We use an HMC190BMS8 RF switch for 900 MHz and 2.4 GHz [8], and a UPG2163T5N switch for 5 GHz [27]. Both switches are mounted on a 2-layer Rodgers 4350 substrate and toggle between open and short impedance states. The switches are connected to the same multiband antenna used at the AP. By

Component	Microcontroller (μW)	ASIC Design (μW)
Oscillator	60	3
RF Switch	33	2
Total Power	93	5

Table 1: Power consumption. Breakdown of IoT device average power consumption by component for both a microcontroller implementation operating with a 3 V supply and an ASIC implementation operating with a 1.2 V supply.

using the onboard 50 MHz oscillator and PLL, we use this setup to experiment with different offsets prior to settling on 2 MHz.

The second prototype focuses on achieving our desired centimeter scale form factor and low power consumption. Specifically, all our low power IoT device needs is an oscillator and RF switch, as the coding is offloaded to the transmitter. A potential solution is to use commercially available ultra-low power MEMS oscillators such as [33]. The oscillators themselves consume as little as $33 \mu\text{A}$ for continuous operation with no load, and while this would achieve a battery life of over 10 yrs operating on two 1000 mAh AAA batteries, if we constrain ourselves to the form factor of two 5.8 mm diameter button cell battery each with a 21 mAh capacity its lifetime would only be a few months as it can only run continuously. In order to optimize for both size and form factor without custom silicon, we leverage low power microcontrollers designed for IoT applications. A microcontroller such as the Kinetis KL03 requires roughly $30 \mu\text{A}$ to run its onboard oscillator at 8 MHz, and only 77nA in its lowest power sleep mode [29]. Because the platform is programmable, we can adjust the duty cycle to achieve significantly longer lifetimes on platforms with tiny batteries.

We fabricate our off-the-shelf prototype on a standard 1-sided FR4 flex PCB material [9] using the Kinetis KL03 microcontroller which is available in an $2.0 \times 1.6 \text{ mm}$ WLCSP package, two UPG2163T 5N RF switches [27], and a 900 MHz and 2.4 GHz dual band chip antenna as well as a 5 GHz chip antenna. We select these ceramic chip antennas which are specifically designed for small form factor applications and specify antenna gains of up to 3 dBi with efficiencies of 60-70% at 900 MHz and 2.4 GHz [2] and up to 79% at 5 GHz [34]. The final assembly is shown in Fig. 2 which consumes an average of $93 \mu\text{W}$.

The final prototype further miniaturizes the device by focusing on just 5 GHz as shown in Fig. 2. We use a fast-turnaround laser micro-machining method to produce flexible circuits. We begin by placing a sheet of copper foil on a low-tack adhesive and cut the outline of the desired copper traces using a UV DPSS laser micromachining system. Next we peel the excess copper off of the adhesive leaving only the desired pattern. We then place a piece of $25 \mu\text{m}$ thick Kapton tape, which can withstand high temperatures required for soldering, onto the copper and lift the traces off of the adhesive. This method could be repeated and stacked to produce a multi-layer design connected through vias as with a normal PCB. We use only the 5 GHz antenna in this prototype and hence are limited to a smaller range.

IC Design. Further miniaturization and power optimization can be achieved by implementing a custom IC which allows for combining the RF switch and impedances into a single chip. Further this significantly decreases the required area to only a few mm^2 . The full IC design consists of a frequency synthesizer, RF switch and at

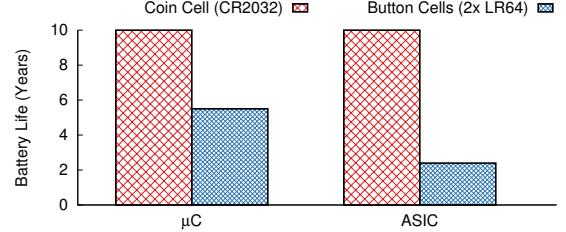


Figure 7: Prototype battery life. Battery life estimates for duty cycled operation of our prototypes operating at 3 V from a coin or two button cell batteries. The plot is limited to 10 years, which is the shelf life of a button cell.

least two impedances states. We design and simulate a complete solution in a TSMC 65 nm LP process. The frequency synthesizer consists of a PLL that produces the control signal for the RF switch. The switch then toggles between different passive elements that can be implemented on chip. In addition to reducing the oscillator power consumption, an ASIC design which integrates the RF switch significantly reduces capacitance compared to off-the shelf designs which require interconnects between chips. The lower capacitance therefore reduces the dynamic power consumption of the switch. Our design also operates at 1.2 V unlike the minimum 1.8 V required by the microcontroller. This has the added benefit of being able to use a single 1.5 V button cell battery without the need for a boost converter which would consume significant area and require additional power. We note that compared to prior backscatter designs [14, 17] our IC is very simple and only shifts the frequency of the incoming signal. In total this consumes $5 \mu\text{W}$ of power. The additional power savings compared to the microcontroller based design allow us to run the device at 100% duty cycle while achieving similar battery life.

Fig. 7 shows the lifetime of both our off-the-shelf and IC designs with different battery-sizes. We limit the maximum of the plot to 10 years as this is the typical maximum shelf life of button cell batteries. These battery life values demonstrate that our design is so low power that the system performance is no longer limited by the electronics but rather the battery technology. Table 1 includes a detailed breakdown and demonstrates we can achieve the goal of low-power IoT localization in a centimeter scale form factor.

4 EVALUATION

We evaluate our system in line-of-sight and through-wall settings. We then deploy μ Locate in five different homes and a hospital to measure performance in real world environments.

4.1 Benchmarking Accuracy

LOS scenario. We conduct experiments on a 100 by 100 m open field. We place the AP at one end of the field and move our FPGA IoT prototype away from the AP in increments of 10 m along different angles. Fig. 8 plots the 3D localization error and shows that:

- We have a 60 m range in LOS scenarios at which the worst case 3D accuracy is 1.5 m. Beyond that distance, the received power of the backscattered signal was too low to decode even with the chirp spread spectrum coding.

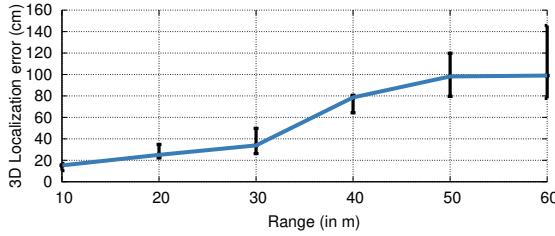


Figure 8: Long range LOS benchmark. The plot shows the 3D localization error for our line of sight benchmark. The figure shows that our operational range is around 60 m.

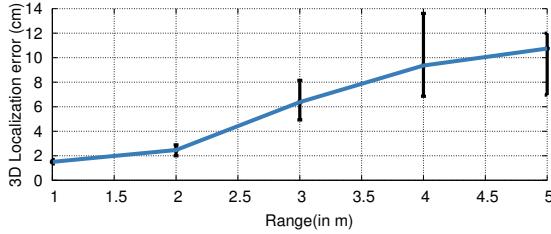


Figure 9: Short range LOS benchmark. The plot shows the 3D localization accuracy for line of sight benchmarks at distances below 5 m. At these distances, our system ends up using all frequencies across 900 MHz, 2.4 GHz and 5 GHz.

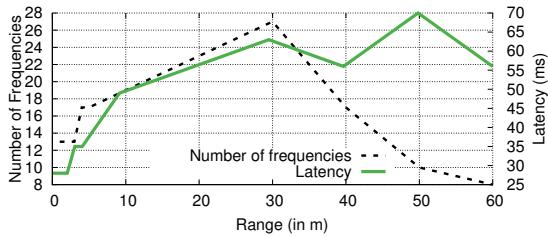


Figure 10: Number of queried frequencies and its corresponding latency. The plot shows the total number of frequencies and the latency required for 3D localization across all locations in Fig. 9 and Fig. 8.

- The accuracy scales with the distance from the IoT prototype. Specifically, we can achieve a localization error of 15 cm at a distance of 10 m, which increases to around 25 cm at a distance of 20 m. This further increases to 78 cm at a distance of 40 m. This change is due to the fact that above 30 m, the received power at 2.4 GHz is noisy due to attenuation in comparison to 900 MHz. This introduces error into the phase measurements reducing the accuracy.

We note that at the above measured distances our algorithm did not pick any 5 GHz frequencies since the corresponding backscattered signal was very weak at these distances. So we re-run the experiments in a 5 m room with finer increments of 1 m. Fig. 9 shows the 3D localization accuracy at these distances. The figure shows that when the IoT prototype is within a meter from the AP, the worse case localization error was less than two cm. At a distance of 2 m we could still achieve a 3 cm worse case accuracy. However

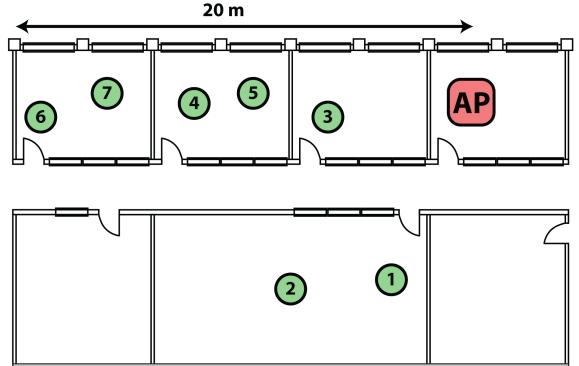


Figure 11: Through-walls setup. Floorplan showing the AP and IoT device in an office environment spanning five rooms. All the doors were closed during the experiments.

the worst case error was less than 14 cm up to distances of 5 m. The reason for these low errors was that the algorithm was able to use frequencies in the 5 GHz range which significantly improve the location accuracies. In particular, 5 GHz helps with the accuracies for two main reasons, 1) higher frequencies translate to smaller wavelengths which allows for better resolution and 2) unlike 900 MHz and 2.4 GHz, each of which have a limited amount of bandwidth, our algorithm could query frequencies across a 180 MHz bandwidth in the 5 GHz range. These results demonstrate that for close range room scale applications we can leverage extra information from 5 GHz whereas applications that require longer ranges cannot leverage these signals.

Finally, Fig. 10 shows the number of frequencies that were queried by the AP before it converged to the location values for all the distances in the above two experiments. The plot shows the following.

- At distances less than 5 m the AP had to query less than 20 frequencies. In fact, when the IoT devices was 1–2 m away the number of queries was even less at 15. This is because at short distances, the direct path is stronger than the non-line-of-sight paths for all the three frequency bands and hence the AP converges on the locations quickly. The latency for these locations is less than 35 ms as multiple frequencies can be queried in parallel.
- Between 10 and 20 m, the AP queries both 2.4 GHz and 900 MHz to disambiguate the direct path and this increases the number of frequencies to 25. However the 900 MHz band frequencies can still be queried simultaneously, leading to a latency of 65 ms.
- An interesting trend happens at longer distances. Here only 900 MHz frequencies are queried. Further, since the accuracies at these distances are much lower, the threshold values are also lower. As a result the number of iterations are reduced to eight for 60 m. However since the SNR is weak, we have to query these frequencies sequentially. These two factors counteract each other and hence the latency stays between 55–70 ms.

Through-walls scenario. Next, we conducted experiments in an office building across multiple office rooms. The offices were separated with dry wall, metal studs, and wooden doors and had typical

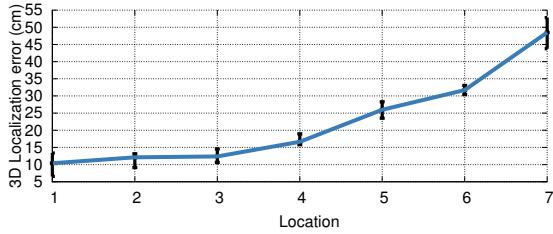


Figure 12: Through-walls accuracy. The plot shows the 3D localization accuracy across the points indicated in Fig 11 rather than range due to their arbitrary placement.

office furniture including tables, chairs and leather couches. Additionally, the tested locations had multiple Wi-Fi access points and 915 MHz RFID readers representing significant interference from other devices. Note that our chirp coding is resilient to both in-band and out-of-band interference [5]. We place the AP in the first office room as shown in the layout in Fig. 11. We then move our IoT prototype to different rooms with their doors closed shown as different points in the layout. For each location, we repeat the localization experiment multiple times and then compute the 3D localization errors.

Fig. 12 plots the 3D localization error as a function of different positions shown in Fig. 11. The figure shows that for the most part the localization accuracy decreases as the distance and the number of walls increases. It is however not always the case due to multipath and fading effects. We note however that the worst case accuracies at location 6 which has 3 walls separating the AP and the IoT prototype was still around 33 cm. This is expected because the 900 MHz and 2.4 GHz backscatter signals were strong enough to be able to reliably decode the phase information, which allows us to disambiguate the multipath.

Mobility evaluation with our 5 GHz miniaturized hardware. Our miniaturized hardware can use both 5 GHz ISM bands. However, since it operates at higher frequencies its range is limited to a single room. We evaluate our ability to track mobile devices. A key challenge with tracking moving objects is that the channel changes quickly with time; however, our above localization solution that queries different frequencies sequentially would take hundreds of milliseconds, during which the channel could change. We instead query multiple 5 GHz frequencies simultaneously using our 7 ms chirps and compute the location. We focus on 5 GHz since antenna sizes at these frequencies can be significantly smaller than 900 MHz and 2.4 GHz resulting in a sub-centimeter form factor device that can be tracked in a room.

To evaluate this design, we performed a controlled experiment by mounting the IoT device on a linear actuator. We use this linear actuator to move the device over a distance of 30 cm with a constant velocity. We then estimate the new location of the IoT device for every 7 ms and compute the error with respect to the actual location of the device on the linear actuator. We repeat this experiment by placing the IoT device setup at increasing distances from the AP in increments of 1 m. Fig. 13 plots the 3D tracking accuracy when the IoT device was placed at different distances from the AP. We see that the tracking error is less than 1 cm when the IoT device is

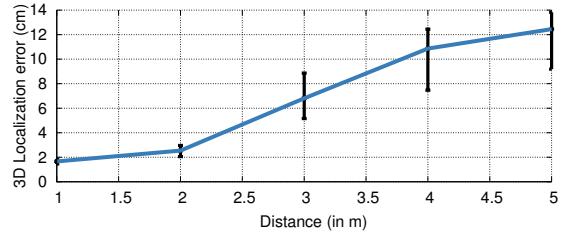


Figure 13: Evaluating mobile tracking accuracy. Evaluation of tracking accuracy using only 5 GHz at short ranges compatible with our smallest 5 GHz prototype.

placed within a distance of 1 m and remains less than 15 cm when it is placed at a distance of 5 m.

4.2 Real World Deployments

4.2.1 Home Deployments. We deploy our system in five homes in a major metropolitan area to understand its performance in realistic use cases. We select a variety of homes including three single story apartments and two multi-story townhouses. The single story apartments had two to three rooms each, while the multi-story townhouses had two or more floors with two and three rooms respectively. In the apartment deployments, we select a central location for the AP to maximize coverage. For the townhouse deployments we place the AP on the bottom floor for convenience. In each of the homes, we select a variety of locations including different rooms, behind closed doors, in closets, on shelves and even hidden in couches to understand if our system can actually localize objects across a whole home and enable item tracking applications.

Fig. 14 shows CDFs for 3D localization accuracy across each of the five homes. The figure shows that for the first three homes the worst-case localization accuracy was less than 30 cm. These three homes correspond to single story apartments where all the devices are on the same floor. The worst-case accuracy was around 60 cm and 1.2 m for home 4 and 5 respectively. These two homes were multi-storied townhouses where the devices were on different floors. The higher error is due to two main factors. First, it was difficult to get the baseline distance measurements across floors. This contributed to errors in estimating the actual location of the IoT device. Second, in home 5, different floors were connected through a narrow staircase whereas the direct path was through thick ceilings that significantly attenuated the signal. This highlights a basic challenge with localization techniques that require some direct path to appear at the receiver. Fig. 15 depicts the above results by classifying them into categories across all the homes. We categorize the locations as LOS, NLOS on the same floor, hidden within a couch, in a closed closet and finally on a different floor. For the reasons described above, the accuracy was lower when the device was on a different floor. When the IoT device was hidden in a couch on the same floor, the error was less than 30 cm.

4.2.2 Hospital deployment. In order to evaluate realistic use cases in healthcare scenarios we deploy our system in a local hospital. Specifically, we perform experiments in the surgery wing of the hospital and perform localization in patient pre-op and post-op rooms as well as storage facilities. Fig. 16 shows the floor plan of

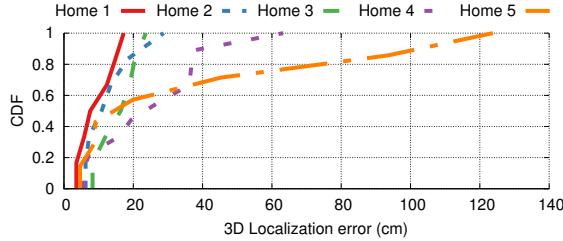


Figure 14: Accuracy per home. The plot shows a CDF of localization error for all points measured in a home.

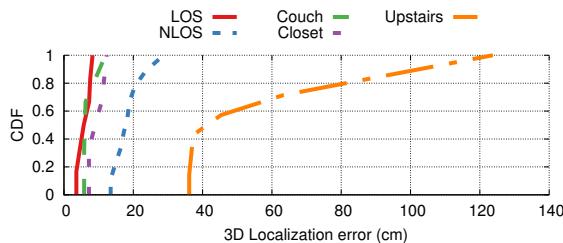


Figure 15: Accuracy by scenario. This plot shows a CDF of localization errors across different categories such as LOS placement and locations in closets.

the approximately 5000 ft^2 surgery wing. The area includes a waiting room and check-in desk, followed by a hallway with a row of patient rooms for pre-op and post-op care as well as a storage room. We perform measurements at locations that represent realistic use cases as indicated in Fig. 16. We select a location for the AP in the side hallway in order to minimize disruption to hospital staff. We select locations in patient rooms and a storage closet as these are typical scenarios where hospital staff maintain a standard inventory of items. Additionally, we select other arbitrary locations in the hallways for tracking mobile equipment such as IV poles and vital signs monitors that travel with patients to different rooms. The majority of these locations do not have direct line of sight to the AP, and include barriers such as curtains, sliding glass doors, and standard wooden doors.

Fig. 17 shows the tracking accuracy across each of these locations that are ordered by distance. We note that the duplicated points represent separate measurements at the same location with barriers such as curtains or doors open and closed. Our system achieves a mean accuracy of 35.12 cm across all of the different locations in this hospital setting. Further, as is expected from our design, the accuracy scales with the distance to the tag: close by locations can achieve an error lower than 20 cm while farther locations have localization error of 70 cm. These errors are small enough that we can track the equipment in the hospital across different rooms as well as the closet area. We note that in hospital post-op and pre-op settings the layout is typically on a single floor. Further the barriers between the rooms are either curtains or thin doors. Thus, we can achieve high localization accuracy in this application setting.

5 RELATED WORK

Wireless localization and tracking. Prior work on ranging with Bluetooth and ZigBee use signal strength as a proxy for distance [11, 13, 15]. RSSI based solutions however are known to perform poorly due to multi-path and further cannot achieve 3D localization. More importantly, radio technologies including Blueooth, ZigBee, SIGFOX and LoRa radios consume tens of mW of power and hence have a limited battery-life for continuous tracking applications.

Another research thread has been focused on RFID localization [32]. RFID has a range of a few meters and is used today for inventory applications. Tagoram [42] leverages the mobility of an RFID tag to create a virtual antenna array and achieve accurate RFID tracking. RF-IDraw [39] uses a total of eight antennas to track an RFID using Angle of Arrival techniques. MobiTagbot [31] determines the precise spatial ordering of closely-spaced RFID tags using synthetic aperture radar techniques. [23] uses the non-linear properties of backscatter hardware to achieve centimeter resolution in LOS scenarios up to a few meters. Rsolve [24] shows that existing RFID tags can backscatter signals over a bandwidth of 220 MHz centered around 915 MHz. While these systems have significantly advanced RFID localization, RFID has a limited range of 2-5 m and does not work through walls. In contrast, we leverage CSS modulation to achieve significantly longer ranges. Instead of using standard UHF RFID tags, [7] proposes backscattering wide band FMCW signal which has hundreds of MHz to GHz of bandwidth for localization. In contrast the chirps in our system have a bandwidth of 500 kHz. This narrowband nature enables us to achieve a high sensitivity and long ranges.

There has also been significant interest in UWB radios which can provide very high localization accuracies. [10, 26] achieve a localization accuracy of 4 mm using ultra-wideband radios up to distances of 3.6 m but consume 195 mW. Polypoint [18] achieves an accuracy of 1.4 m at a range of 20 m but consumes 150 mW. Harmonium [19] consumes 75 mW and provides a localization accuracy of 31 cm at ranges of 2.7 m. RF-Echo [6] consumes 62 mW but can achieve an accuracy of 26 cm at distances of 90 m. While these methods achieve good accuracy, they consume more power than Bluetooth which makes them a poor choice for continuous tracking applications. The closest related work is Slocalization [25] (short for slow/static localization) which backscatters UWB signals to achieve low-power localization. This design however works only with static scenarios and incurs delays on the order of minutes to hours to output the location value. This is because FCC regulations significantly limit the transmitted power of UWB signals compared to typical transmissions in ISM bands. Further, since the backscatter system in [25] does not use coding like the CSS modulation used in our design, it requires integrating the received signal over 10 minutes to more than an hour, depending on the deployment, to get the location value. In contrast, our approach can provide the location value within 70 ms while achieving a range of 60 m and thus can support practical applications.

Finally, significant advances have been made in Wi-Fi based localization in recent years. PinLoc [30] uses the Wi-Fi channel state information (CSI) signatures, while Arraytrack [40] and SpotFi [20] use AoA techniques to achieve sub-meter localization using Wi-Fi



Figure 16: Hospital deployment. Deployment in a hospital surgery wing. We select nine points shown on the floorplan (Left) including duplicates to test doors and curtains. We include patient pre/post-op rooms (Center) and storage facilities (Right).

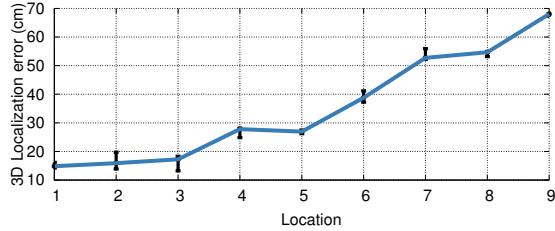


Figure 17: Hospital accuracy. Localization accuracy results for each of the points marked in Fig. 16.

devices. ToneTrack [41] shows how to combine the CSI information across multiple 2.4 GHz Wi-Fi channels to achieve sub-meter accuracies with the WARP hardware. Chronos [37] uses the Chinese remainder theorem to demonstrate the above concept on Wi-Fi hardware. While these Wi-Fi based localization systems make algorithmic contributions we leverage, Wi-Fi is power consuming and is not used in many IoT applications.

Backscatter communication. There has been recent interest in backscatter as a low-power communication mechanism. These techniques either backscatter existing TV [22], Wi-Fi [4, 16] and FM signals [38] or generate Wi-Fi compliant transmissions using techniques like Passive Wi-Fi [14, 17, 43] and FS-backscatter [44]. These Wi-Fi based approaches have a receiver sensitivity of only -90 dBm and hence have a limited range and cannot work across rooms unless the signal source is placed close to the backscatter tag [21].

There has also been recent interest in long range backscatter solutions [35, 36] of which [35] achieves a longer range and is compatible with off-the-shelf LoRa radios. However this prior work does not support localization. Further, existing implementation of LoRa backscatter requires FPGAs and consumes 5–10 milliwatts of power. In contrast, we introduce a novel architecture that delegates the complex CSS coding operations to the AP and introduce a CSS backscatter design that has orders of magnitude lower power.

6 DISCUSSION AND CONCLUSION

We present the first wireless localization system that consumes microwatts of power in a sub-centimeter form factor and can be localized across a whole-home or hospital through walls. In this section, we outline limitations and avenues for future research.

Supporting multiple devices. In addition to using time-division multiplexing, we can also set different backscatter devices to shift the signals by different frequencies to support multiple devices

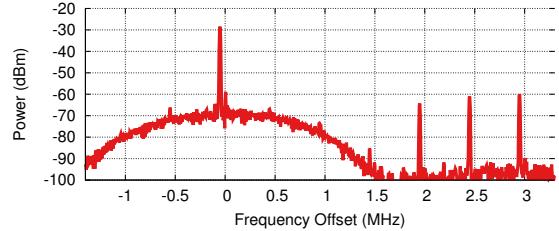


Figure 18: Feasibility of multiple devices. Snapshot of the chirp signal captured on a spectrum analyzer. The plot shows the baseband spectrum of the original coded transmissions as well as three backscattered signals at frequency offsets of 2, 2.5, and 3 MHz.

as shown in Fig. 18. Specifically, each backscatter device could use a different shift and reflect the incoming signal from the AP at the same time. The receiver can process the received signal across different shifts to concurrently localize multiple devices. More specifically, recent work has demonstrated that upto 256 devices can concurrently transmit using chirp spread spectrum [12]. Since our design leverages chirp spread spectrum, one can design large scale networks where devices can concurrently transmit and be localized at the same time.

Using multiple APs. In order to achieve high accuracies with a single AP, our algorithm relies on knowledge of the floor plan of a room to eliminate infeasible location estimates. We realize however that some applications may require more generalizable solutions for example where the localized objects might be moved outside of a known set of rooms. The key reason for requiring these constraints however was that we use a single AP for localization. Adding additional access points, or additional antennas with greater separation would provide better angular resolution and hence address this limitation. For example, an AP placed on the orthogonal wall of a building or on another floor in multi-floor scenarios would help disambiguate multipath and provide more information to estimate the location. Future work could build upon the basic techniques we show here to explore the trade off between the infrastructure overhead of adding additional APs and antennas versus the robustness and accuracy improvements they would contribute.

Communication. While our focus is to design low-power hardware that achieves localization, we note that we can use the same architecture for long range communication. The data is encoded using ON-OFF keying modulation by turning the oscillator ON and

OFF. Thus we can enable an IoT device that can both send information and be localized. Evaluating the BER of this communication system is not in the scope of this paper.

LOS path. Existing 3D localization algorithms assume that while there is multipath, there is at least some energy from the direct path at the receiver. Our design also makes a similar assumption. We note however that our design also increases the probability that the direct path signal has some energy by leveraging frequency diversity across the three ISM bands. Specifically, while the direct path signal could be weak at a specific frequency, it is likely to be noticeable at at least one of the three ISM bands.

ACKNOWLEDGEMENTS

We thank the SenSys reviewers for their helpful feedback. This work was funded in part by the National Science Foundation and Google Faculty Research Awards.

REFERENCES

- [1] 2016. Lora Modulation Basics. (2016). <http://www.semtech.com/images/datasheet/an1200.22.pdf>.
- [2] Pulse Larsen Antennas. 2017. W3320 ISM868/915, ISM2.4G. (2017). <http://www.mouser.com/ds/2/336/W3320-1082542.pdf>
- [3] A. Berni and W. Gregg. 1973. On the Utility of Chirp Modulation for Digital Signaling. *IEEE Transactions on Communications* 21, 6 (Jun 1973), 748–751. <https://doi.org/10.1109/TCOM.1973.1091721>
- [4] Dinesh Bharadia, Kiran Raj Joshi, Manikanta Kotaru, and Sachin Katti. 2015. BackFi: High Throughput WiFi Backscatter (*SIGCOMM '15*).
- [5] Ludovic Champion and Nicolas Sornin. 2014. Chirp Signal Processor. (7 2014). European Patent Application EP2975814A1.
- [6] L Chu, Z Luo, D Sylvester, D Blaauw, and H Kim. 2017. RF-Echo: A Non-Line-of-Sight Indoor Localization System Using a Low-Power Active RF Reflector ASIC Tag. In *Mobicom'17*.
- [7] Itay Cnaan-On, Stewart J Thomas, Matthew S Reynolds, and Jeffrey L Krolik. 2014. Multichannel radar backscatter communication and localization. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 76–80.
- [8] Analog Devices. 2015. HMC190BMS8. (2015). <http://www.analog.com/media/en/technical-documentation/data-sheets/hmc190b.pdf>
- [9] Pulsar Professional FX. 2017. Fab-in-a-box Single Sided Flex PCB. (2017). https://pcbfx.com/main_site/pages/products/flex_pc_board.html
- [10] H. G. Han, B. G. Yu, and T. W. Kim. 2015. 19.6 A 1.9mm-precision 20GS/S real-time sampling receiver using time-extension method for indoor localization. In *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*.
- [11] Rasmus HENRIKSSON. 2016. Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware (*Master's thesis in Communication Engineering*, 2016).
- [12] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. 2019. NetScatter: Enabling Large-Scale Backscatter Networks (*NSDI'19*).
- [13] A. K. M. M. Hossain and W. S. Soh. 2007. A Comprehensive Study of Bluetooth Signal Parameters for Localization. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*.
- [14] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. 2016. Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices. In *Proceedings of the 2016 ACM SIGCOMM Conference*.
- [15] Troy A. Johnson and Patrick Seeling. 2012. Localization Using Bluetooth Device Names. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '12)*.
- [16] Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R. Smith, and David Wetherall. 2014. Wi-Fi Backscatter: Internet Connectivity for RF-powered Devices. In *Proceedings of the 2014 ACM Conference on SIGCOMM*.
- [17] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. 2016. Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*.
- [18] Benjamin Kempke, Pat Pannuto, and Prabal Dutta. 2015. PolyPoint: Guiding Indoor Quadrotors with Ultra-Wideband Localization. In *Proceedings of the 2Nd International Workshop on Hot Topics in Wireless (HotWireless '15)*.
- [19] Benjamin Kempke, Pat Pannuto, and Prabal Dutta. 2016. Harmonium: Asymmetric, Bandstitched UWB for Fast, Accurate, and Robust Indoor Localization (*IPSN '16*).
- [20] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. *SIGCOMM'15* (2015).
- [21] Manikanta Kotaru, Pengyu Zhang, and Sachin Katti. 2017. Localizing Low-power Backscatter Tags Using Commodity WiFi. In *CoNext'17*.
- [22] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. 2013. Ambient Backscatter: Wireless Communication out of Thin Air (*SIGCOMM '13*).
- [23] Yunfei Ma, Xiaonan Hui, and Edwin C. Kan. 2016. 3D Real-time Indoor Localization via Broadband Nonlinear Backscatter in Passive Devices with Centimeter Precision (*MobiCom '16*).
- [24] Yunfei Ma, Nicholas Selby, and Fadel Adib. [n. d.]. Minding the Billions: Ultra-wideband Localization for Deployed RFIDs. In *Mobicom'17*.
- [25] Pat Pannuto, Benjamin P. Kempke, and Prabal Dutta. 2018. SLocalization: Sub-muW, Static, Decimeter-Accurate Localization with Ultra Wideband Backscatter. In *IPSN (IPSN)*.
- [26] T. Redant, T. Ayhan, N. De Clercq, M. Verhelst, P. Reynaert, and W. Dehaene. 2014. 20.1 A 40nm CMOS receiver for 60GHz discrete-carrier indoor localization achieving mm-precision at 4m range. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*.
- [27] Renesas. 2011. UPG2163T5N. (2011). <http://www.cel.com/pdf/datasheets/upg2163t5n.pdf>
- [28] Olivier BA SELLER and Nicolas Sornin. 2016. Low power long range transmitter. (Feb. 2 2016). US Patent 9,252,834.
- [29] NXP Semiconductors. 2017. Kinetis KL03 32 KB Flash. (2017). <https://www.nxp.com/docs/en/data-sheet/KL03P24M48SF0.pdf>
- [30] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information (*MobiSys '12*).
- [31] Longfei Shangguan and Kyle Jamieson. 2016. The Design and Implementation of a Mobile RFID Tag Sorting Robot (*MobiSys '16*).
- [32] Longfei Shangguan and Kyle Jamieson. 2016. Leveraging Electromagnetic Polarization in a Two-Antenna Whiteboard in the Air (*CoNEXT '16*).
- [33] SiTime. 2017. SiT1576 Low-Jitter, 1Hz-2.0 MHz Super-TCXO. (2017). <https://www.sitime.com/products/datasheets/sit1576/SiT1576-datasheet.pdf>
- [34] Taoglas Antenna Solutions. 2017. CA.50 5150-5900 MHz Ceramic Chip Monopole. (2017). <http://www.taoglas.com/wp-content/uploads/2015/08/CA.50.pdf>
- [35] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R. Smith, and Shyamnath Gollakota. 2017. LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* (2017).
- [36] Ambuj Varshney, Oliver Harms, Carlos Pérez-Penichet, Christian Rohner, Frederik Hermans, and Thieno Voigt. 2017. LoReA: A Backscatter Architecture That Achieves a Long Communication Range. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys '17)*.
- [37] Deepak Vasisht, Swaran Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point. In *NSDI*.
- [38] Anran Wang, Vikram Iyer, Vamsi Talla, Joshua R. Smith, and Shyamnath Gollakota. 2017. FM Backscatter: Enabling Connected Cities and Smart Fabrics. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*.
- [39] Jie Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals (*SIGCOMM '14*).
- [40] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *NSDI*.
- [41] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. [n. d.]. ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization (*MobiCom '15*).
- [42] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-time Tracking of Mobile RFID Tags to High Precision Using COTS Devices (*MobiCom '14*).
- [43] Pengyu Zhang, Dinesh Bharadia, Kiran Joshi, and Sachin Katti. 2016. HitchHike: Practical Backscatter Using Commodity WiFi. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM (SenSys '16)*. ACM, New York, NY, USA, 259–271. <https://doi.org/10.1145/2994551.2994565>
- [44] PENGYU ZHANG, Mohammad Rostami, Pan Hu, and Deepak Ganesan. 2016. Enabling Practical Backscatter Communication for On-body Sensors. In *Proceedings of the 2016 ACM SIGCOMM Conference*.