Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets

Pablo Corbalán
University of Trento, Italy
p.corbalanpelegrin@unitn.it

Gian Pietro Picco
University of Trento, Italy
gianpietro.picco@unitn.it

Sameera Palipana
Cork Institute of Technology, Ireland
sameera.palipana@cit.ie

ABSTRACT

We propose Chorus, a new ultra-wideband (UWB) localization scheme in which the target device computes the time difference of arrival (TDOA) of signals sent *concurrently* by localization anchors in known positions. This scheme, similar to GPS, is the opposite of existing TDOA schemes for UWB, where the target transmits the signal and anchors compute the time difference. This reversed perspective enables several advantages in Chorus, including support for countless targets.

The cornerstone and novelty of Chorus is the use of *concurrent transmissions* from anchors; distance information is acquired at the receiver based on the channel impulse response (CIR) resulting from the fused signals. We contribute *i)* an analytical model enabling a priori estimation of the CIR resulting from the superposition of concurrent signals *ii)* techniques to accurately extract the time-of-flight information necessary for localization from the measured CIR, and *iii)* real-world experiments that validate the model as well as assess the practical feasibility and performance of Chorus.

Experiments with the DW1000 UWB chip show that Chorus achieves sub-meter positioning accuracy. However, our model shows that performance is limited by idiosyncrasies of the DW1000 that, if removed in next-generation UWB hardware, could unlock an order of magnitude improvement in accuracy.

CCS CONCEPTS

• **Networks → Location based services**;

KEYWORDS

Ultra-wideband, Concurrent Transmissions, TDOA Localization

ACM Reference Format:
Pablo Corbalán, Gian Pietro Picco, and Sameera Palipana. 2019. Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets. In *The 18th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2019) (IPSN '19), April 16–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3302506.3310395

IPSN '19, April 16–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6284-9/19/04…$15.00
https://doi.org/10.1145/3302506.3310395

1 INTRODUCTION

Global navigation satellite systems like GPS are arguably the most successful localization systems to date. One of their key features is that the localization infrastructure, provided by satellite-based anchors, supports an *arbitrarily large density* of users. The reason is that GPS and the like *do not require transmissions from localization targets*; the latter are passive listeners processing the signals from a satellite constellation. Therefore, the number and density of targets does not impact the capacity of the communication channel, which depends only on the configuration of the localization anchors.

However, GPS cannot operate in many so-called GPS-denied environments notably including indoors, where people work and dwell most of the time. This limitation has fueled decades of research on alternatives, among which those based on wireless radio communication are arguably the most popular [22], boosted by the pervasiveness of this technology on our personal devices and, increasingly, on the Internet of Things ones deployed around us.

Radio-based localization: Conflicting tradeoffs. Nevertheless, to date no radio-based localization system has satisfactorily addressed accuracy *and* scalability as effectively as in GPS. Systems based on RSSI estimation [1, 36] operate on a similar principle; the signal from several anchors (e.g., WiFi access points or BLE beacons) is processed on the (passive) localization targets. However, accuracy is generally poor (meters); it can be significantly improved by fingerprinting [11] but at the cost of a significant overhead and rigidity in the system deployment and management. Recent WiFi-based approaches based on channel state information (CSI) improve up to decimeter-level accuracy, but require dedicated hardware.

On the other hand, ultra-wideband (UWB) radios recently returned to the forefront of academic and commercial interest, thanks to their pulse-based operation enabling higher ranging accuracy (<10 cm), and the availability of small, cheap, and low-power UWB radios. However, the localization schemes commonly used with UWB have their own limitations. Two-way ranging (TWR), arguably the most common scheme and part of the IEEE 802.15.4 standard [12], is based on request-reply exchanges estimating the time of flight between the localization target and *all* anchors (Figure 1a). This inherently limits the scalability of the system, especially when a high rate of localization is required. Alternate schemes, like time difference of arrival (TDOA) provide higher scalability, in that they require a *single* packet transmission from the target, whose receipt is timestamped at each localization anchor (Figure 1b). Location can be determined based on the difference between these timestamps, caused by the different distance of anchors w.r.t. the target. However, *i)* this requires anchors precisely time-synchronized, typically via an out-of-band wired infrastructure, and *ii)* does not entirely remove scalability issues, since the transmissions from multiple tags must be properly scheduled to avoid collisions, problematic in

dense environments and/or with high-rate localization. We discuss further this necessary background and related work in §2.

Reconciling tradeoffs with concurrent UWB transmissions. We propose Chorus, a system that *i)* exploits the superior accuracy of UWB, but *ii)* in a scheme where, similar to GPS, the localization target is a passive listener and never transmits.

In a nutshell, Chorus can be regarded as a sort of "flipped" TDOA, still relying on the difference between timestamps but completely reversing the perspective: the TDOA computation is performed *at the localization target*, based on signals that are sent *simultaneously by all anchors*—as in a chorus, hence the name.

Simultaneous transmissions from multiple senders to the same receiver is problematic for radio communication, due to collisions preventing the signal decoding. However, the authors of [5] recently observed that this is not necessarily the case for UWB, and demonstrated empirically that the channel impulse response (CIR) information made available by UWB chips enables successful reconstruction of the time of arrival of signals belonging to concurrent transmissions, and therefore distance estimation. In [5], this was demonstrated in the base TWR scheme (Figure 1c). Instead, in Chorus the use of concurrent transmissions is precisely what enables TDOA computation at the target (Figure 1d).

Chorus unlocks several advantages currently not found together in any other indoor localization system:

- *accurate localization*, inherited from UWB;
- *infinite scalability* in the number and/or density of localization targets, as in GPS;
- *location privacy*, as positioning data remain local to the target as in GPS and TWR, and unlike mainstream TDOA;
- *efficient use of the communication/ranging channel*, occupied only for the duration of a single packet exchange, as in TDOA. This enables very high localization rates, precluded to TWR;
- *no need for scheduling target transmissions*, required instead by TWR and TDOA, as Chorus is entirely passive like GPS;
- *on-demand localization* as the target can decide *autonomously* if and when to locate itself without fear to disrupt schedules. This enables adaptive schemes where the localization system changes its positioning rate dynamically or is even switched off for prolonged periods, e.g., to reduce energy consumption.

Challenges and contribution. A fundamental contribution we put forth is the definition of the Chorus TDOA scheme enabling the unprecedented advantages above. However, while this scheme is conceptually simple, its realization poses significant challenges:

Ch1. the accuracy of distance estimation degrades with concurrent transmissions, due to their mutual interference [5];

Ch2. to compute a position, we must identify the individual pulses corresponding to each concurrent transmission in the merged signal and map them to the corresponding anchor;

Ch3. UWB chips are designed to provide an accurate time of arrival (TOA) estimation for a *single* pulse in the CIR, while Chorus requires it for *all* concurrent pulses found in the same CIR;

Ch4. UWB chips are not designed with concurrent transmissions in mind, which are therefore hampered by hardware characteristics that do not bear an impact on isolated transmissions.

We tackle challenge Ch2 and Ch3 directly with our design of Chorus (§3). We exploit an existing [6] TOA estimation technique and adapt it for concurrent transmissions, showing experimentally its effectiveness (Ch3). This contribution goes beyond Chorus, and is generally applicable to concurrent ranging [5]. As for Ch2, inspired by recent work [7], we empirically demonstrate that pulse shaping is not a viable mechanism for pulse identification, due to multi-path components (MPC), and show instead the effectiveness of response position modulation, i.e., the introduction of known, ns-level time shifts among the concurrent transmissions.

The complexity of Ch1 and Ch4, instead, requires a multi-pronged approach (§4) that goes beyond the evaluation of our Chorus prototype based on the DecaWave DW1000 UWB chip. A cornerstone of this approach is a *model* (§5) that, based on isolated CIR signals, estimates the CIR resulting from their concurrent transmission. The input CIRs can be gathered empirically, from in-field experiments, or be generated analytically, in our case via the well-known UWB model in the IEEE 802.15.4 standard [24]. This contribution allows us to fully ascertain the potential of our technique *i)* without the effort of experiments and the bias of the environment (Ch1), and *ii)* unconstrained by hardware limitations (Ch4). Further, it is applicable to concurrent ranging approaches [5].

Chorus is evaluated in §6 by leveraging both our prototype and the aforementioned model(s). Experiments with the former in two different setups at our premises show sub-meter localization accuracy. This is a good result compared to the state of the art, considering the several advantages Chorus brings. Further, it is obtained despite the 8 ns uncertainty in TX scheduling of the DW1000 chip, known [5] to negatively affect the accuracy of ranging based on concurrent transmissions—an instance of Ch4.

This is precisely where models come into play. We show that the model in §5 can be used to estimate accurately the prototype performance, regardless of the method used for gathering the input

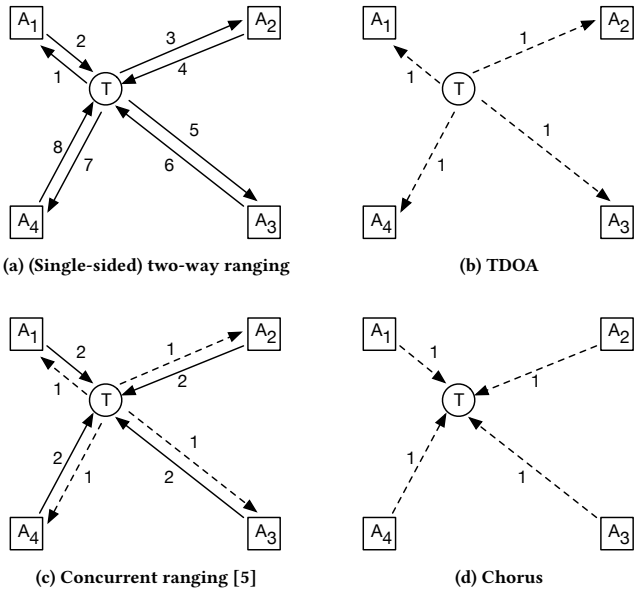


Figure 1: Basic localization schemes and concurrent transmission variants. Solid (dashed) lines denote unicast (broadcast) transmission; numbers denote temporal ordering.

CIRs. This enables us to push our investigation further by exploring the effect of alternate values of the TX scheduling uncertainty, effectively exploring what alternate hardware designs could achieve, and show that *i)* this uncertainty is *the* main limiting factor for Chorus, and *ii)* its removal unlocks significantly higher accuracy. In a sense, although we do not solve directly challenge Ch4, we empower hardware designers with knowledge of the requirements and tradeoffs involved. Our model shows that if this uncertainty is completely eliminated in hardware, localization accuracy is improved by an order of magnitude w.r.t. with 8 ns uncertainty.

The paper ends (§7) with concluding remarks reflecting on the potential of Chorus and the implications of these findings on next-generation UWB hardware.

2 BACKGROUND AND RELATED WORK

We provide the necessary background on UWB radios and review related work on state-of-the-art localization systems.

Impulse Radio UWB. UWB radios based on impulse radio transmit a time-hopping sequence of very short pulses [32], typically ≤2 ns, spreading the signal energy across a bandwidth ≥500 MHz. This very large bandwidth provides excellent time resolution and allows UWB radios to accurately measure the time of arrival (TOA) of a signal and distinguish the direct path from multipath components (MPC), especially by measuring and analyzing the channel impulse response (CIR). As a result, UWB radios are commonly used for precise distance estimation based on ranging schemes.

IEEE 802.15.4 UWB PHY Layer. The IEEE 802.15.4-2011 standard [12] provides an UWB PHY layer based on impulse radio. A UWB frame is composed of two parts: *i)* a synchronization header (SHR) encoded in single pulses, and *ii)* a data portion, including the PHY header and data payload, sent using a combination of burst position modulation (BPM) and binary phase-shift keying (BPSK). The SHR includes a preamble for synchronization and the start frame delimiter (SFD) to delimit the end of the SHR and the beginning of the data portion. The preamble consists of the repetition of a pre-defined *symbol* drawn from a ternary alphabet $\{-1, 0, +1\}$ (positive, absent, and negative pulse) and associated to a preamble code. The symbol duration depends on the pulse repetition frequency (*PRF*). The standard provides two preamble code lengths: 31 elements (*PRF* = 16 MHz) and 127 (*PRF* = 64 MHz). The preamble symbol elements are then spread by interleaving each element with zeros. The highest frequency at which a compliant device shall emit pulses is 499.2 MHz (fundamental frequency), which defines the standard chip duration of ≈2 ns.

UWB Ranging. The IEEE 802.15.4 standard also specifies a two-way ranging (TWR) scheme between two devices, an initiator and a responder. In its simplest form, single-sided TWR (SS-TWR), the initiator sends a POLL message and stores its TX timestamp t_1. The responder replies with a RESPONSE, including in the data payload the POLL RX timestamp (t_2) and the RESPONSE TX timestamp (t_3). When the initiator receives the latter, with RX timestamp t_4, it can compute the time-of-flight $\tau = \frac{1}{2}((t_4 - t_1) - (t_3 - t_2))$ and estimate distance as $d = \tau \times c$, where c is the speed of light in air.

To ameliorate the impact of clock drift in SS-TWR [13, 20], other schemes have been proposed, e.g., double-sided TWR (DS-TWR), which requires 3 or 4 message exchanges to compute the distance at the responder or the initiator, respectively. To estimate its 2D location, the initiator must perform TWR with at least 3 anchors, although more are used in practice to increase localization accuracy.

Concurrent Ranging. Distance estimation against N anchors therefore requires between $2 \times N$ (SS-TWR) and $4 \times N$ (DS-TWR) messages; these long and energy-expensive exchanges limit scalability w.r.t. number of users and update rate, further reduced by the signalling required, e.g., to discover the neighbors to range with.

Concurrent ranging [5] tackles these issues with the N anchors transmitting concurrently their RESPONSE to a POLL broadcast by the initiator, which can then estimate distances based on the received CIR. This scheme reduces the initiator messages to only 2, saving energy and time. The accuracy obtained is lower than with standard schemes, but still sub-meter. This is due to the TX scheduling precision of the radio chip (ϵ = 8 ns in the DW1000); while this is normally compensated by attaching the timestamps in the data payload, this is possible only for one of the N concurrent signals.

Other challenges highlighted in [5] include the reliable detection of the direct path of each response and the identification of the anchor corresponding to each path. These are tackled in [7], respectively via *i)* pulse shaping, to map the shape of CIR peaks to anchor identities, and *ii)* response position modulation, to facilitate path detection in multipath environments by artificially time-shifting responses. Further, the actual detection of responses and therefore of their time of arrival (TOA), is based in [7] on a simplified version of the well-known Search and Subtract algorithm [6].

Inspired by concurrent ranging [5], Chorus exploits a similar notion in a completely different scheme in which mobile tags do not transmit at all and compute their own position based on the reception of a single (concurrent) message, significantly reducing energy and time costs w.r.t. standard schemes but also w.r.t. concurrent ranging. Our design takes inspiration also from the techniques in [7], although we show that real-world environments with multipath hampers their applicability, and propose effective adaptations or alternate techniques.

Localization Systems. UWB localization systems can be categorized in ranging-based [15, 16] and TDOA-based [14, 26, 28]. PolyPoint [16] and SurePoint [15] exploit antenna and channel diversity to increase ranging accuracy; however, they require even longer message exchanges, trading off accuracy for energy. Systems based on TDOA require tight time synchronization of anchors, achieved via wired [14, 26] or wireless [23, 29] mechanisms. The custom receiver design in Harmonium [14] exploits bandstitching to increase the bandwidth of the received signal and therefore the time resolution for TDOA estimation. Slocalization [26] extends bandstitching to transmitters and exploits backscatter communications to achieve sub-μW localization at the cost of significantly reducing (milli-Hz) update rates. Atlas [28, 29] extends conventional TDOA with a one-way wireless time synchronization technique similar to the one we use in Chorus. In all these TDOA schemes anchors listen and mobile tags transmit, which either incurs scheduling costs or unreliability due to collisions [28]. These drawbacks are removed by Chorus, as the tag is a passive listener; further, the tag hardware could consist of the receiver only.

SnapLoc [8] also exploits concurrent transmissions for TDOA-based localization in a way very similar to Chorus; the two systems, developed independently, have been published simultaneously in

the same venue. Differences include implementation details, evaluation methodology, and the (complementary) slant of the contribution. SnapLoc tackles the DW1000 TX scheduling uncertainty by compensating it with dedicated wired and wireless mechanisms relying on the deployed infrastructure. The model (§5) we contribute in this paper, instead, provides a *foundation* for analyzing and predicting the impact of this uncertainty and, in general, the accuracy attainable in a given environment. Interestingly, the results in [8] are in agreement with our model, mutually reinforcing the soundness of both approaches.

Finally, in recent years WiFi-based systems have also achieved decimeter-level error by exploiting channel state information (CSI) in TDOA [34], angle-of-arrival [17], or time-of-flight [30] schemes. However, CSI techniques either require expensive SDR solutions or exploit complex MIMO systems, increasing monetary and energy costs [10]. Similarly, time-based WiFi techniques suffer from the lower bandwidth of WiFi w.r.t. UWB, and require bandstitching and frequency hopping to increase time resolution [30, 34].

3 CHORUS DESIGN

We describe the core principles (§3.1) of Chorus, followed by the techniques (§3.2–§3.4) tackling the challenges outlined in §1.

3.1 Basic Principles of Operation

We assume an infrastructure of $N \geq 4$ anchors with known location and able to communicate with each other. We assign conventionally the initiator role to anchor 1.

Localization Primitive. Chorus is based on a communication primitive involving the transmission of two messages from the anchor infrastructure: one for time synchronization and the other for localization. The synchronization message, however, could be omitted using a wired time synchronization infrastructure as in [14, 26]. Mobile tags are entirely passive and estimate their own position by only listening to the second message.

The initiator transmits a POLL message at time t_p to synchronize all anchors in range. Anchors estimate the TX time as $t_p = t_a - \tau_{ia}$, where t_a is the RX timestamp and $\tau_{ia} = \|\mathbf{p_a} - \mathbf{p_i}\| / c$ is the *known* time of flight between the initiator i and anchor a, determined based on the known positions $\mathbf{p_a}$ and $\mathbf{p_i}$, and the speed of light in air, c.

After the POLL message, all anchors (including the initiator) schedule the TX of a RESPONSE message at time $t_{TX} = t_p + \delta_{TX}$, where δ_{TX} is a fixed delay required to process the POLL message and prepare the transceiver to transmit.

Mobile tags do not transmit any message in Chorus; to compute their own position they only need to listen for the RESPONSE message. Tags may also periodically listen for POLL messages to learn the precise time when RESPONSE messages are going to be transmitted. As multiple anchors transmit their RESPONSE concurrently, there may be RX errors due to collisions. However, this does not affect Chorus; tags compute their position based on the CIR of response messages, which is available even when RX errors occur.

The localization update rate therefore depends solely on the frequency at which this two-message primitive is repeated, and on the processing time required to compute the position at the tags.

Position Estimation. As mobile tags are passive listeners not tightly synchronized with the infrastructure, they cannot directly measure their distance from anchors. Instead, once they have estimated the TOA of the N anchor transmissions (§3.3), they can compute $N-1$ non-redundant TDOA measurements

$$\Delta t_{ai} = t_a - t_i = \frac{\|\hat{\mathbf{p}} - \mathbf{p_a}\| - \|\hat{\mathbf{p}} - \mathbf{p_i}\|}{c} \quad (1)$$

by selecting one anchor (e.g., the initiator) as time reference. The geometric representation of Eq. 1 is a hyperbola whose foci are the position of the anchors ($\mathbf{p_a}$ and $\mathbf{p_i}$) and where the mobile tag is located ($\hat{\mathbf{p}}$). In the absence of noise, the intersection of at least 3 hyperbolas yields the unique tag position [3]; however, in the presence of noise, the set of TDOA equations yield inconsistent tag positions and cannot provide a solution. To determine the tag position $\hat{\mathbf{p}}$, we solve the non-linear least squares problem by minimizing the squared difference between the measured TDOA estimates $\hat{\Delta}t_{ai}$ and the theoretical TDOA measurements Δt_{ai} as

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{a=2}^{N} \left(\hat{\Delta} t_{ai} - \Delta t_{ai}\right)^2 \quad (2)$$

Large-scale deployments. We assume that mobile tags roam in a *confined* area with at least $N \geq 4$ anchors in range—a common assumption in the literature. However, Chorus can scale to larger, non-confined settings if its two-message exchanges are properly scheduled across multiple "cells" of anchors. This can be achieved, e.g., by *i)* using different per-cell RF configurations (e.g., complex channels [21]), requiring extra coordination to enable tags to dynamically switch to the right cell setting, or *ii)* exploiting a time-slotted schedule coordinating transmissions from nearby cells, at the cost of reducing the maximum achievable update rate. We are currently evaluating these alternatives as part of our ongoing work.

3.2 Anchor Identification & MPC Avoidance

As Chorus relies on multiple anchors transmitting concurrently, the CIR at the receiver contains information from the direct path of each anchor transmission and the associated multi-path components. If all anchors transmit at the same time t_{TX} and are at similar distances from the tag, their peaks in the CIR overlap [5], making it difficult to *i)* accurately identify the anchor associated to each peak, and *ii)* correctly discern the direct path of each anchor transmission from other responses and/or strong MPC. Next, we discuss techniques to address these issues.

Pulse shapes. The DW1000 offers the option to change the TX pulse width via the 8-bit register `TC_PGDELAY` [21]. This is exploited for concurrent ranging in [7], by encoding the anchor IDs as different pulse shapes; a matched filter is then used to map the pulse shapes observed in the CIR with the anchor ID. Inspired by this work, we measured 3 pulse shapes (0x93, 0xc8, and 0xe6 using channel 7) in an open field, to avoid MPC. For each pulse shape, we attached a 30 dBm attenuator to the transmitter, and performed 1000 wireless TX, collecting 1000 CIR signals. We then removed the direct CIR pulse from the CIR and averaged the 1000 aligned pulses, obtaining a template for every pulse shape as in [7]. We tested the effectiveness of the pulse shapes to identify the anchor ID using the same dataset. Across 3000 CIR pulses, we only obtained a single mis-identification, yielding a 99.97% identification accuracy. We then evaluated the reliability of this procedure in the same real multipath environment used in §6 (Figure 8b). The 5 anchors all

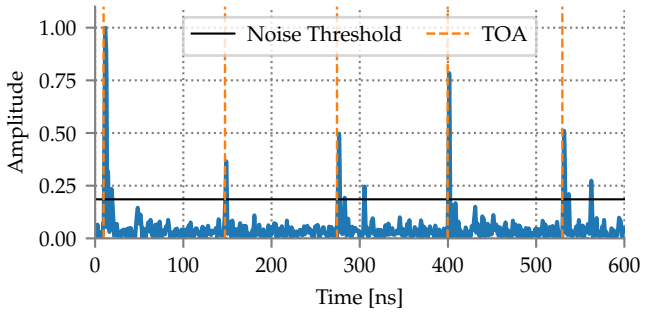


Figure 2: TOA estimation from concurrent responses.

used pulse shape 0x93 for TX, while the tag used 0x93, 0xc8, and 0xe6 to detect the anchor ID with the matched filter. Across 513×5 CIR samples, 0x93 was correctly identified only in 36.1% of the cases; otherwise, it was mistaken for 0xc8 (59.7%) and 0xe3 (3.3%).

The reason is that, in real multipath environments, we cannot assume the channel to be always separable [6, 33], that is $|\tau_i - \tau_j| < T_p$, for some $i \neq j$, where T_p is the TX pulse duration, and τ_i, τ_j the time delays along different paths. Therefore, we cannot always separate the direct path from closely-spaced MPC, which deforms the received pulse shapes and causes an ambiguity in the matched filter output, where a low-bandwidth pulse can be a superposition of time-shifted high-bandwidth pulses. We conclude that, unlike what reported in [7], this technique is unreliable in real MPC environments, where it likely mis-identifies anchors.

Response Position Modulation. We therefore resort to another technique from [7]: adding an artificial time shift T_a (≈100s of ns) to the RESPONSE TX time t_{TX}. We define $T_a = (a-1)T$, where a is the anchor ID; the real TX time of anchor a becomes $t_{TX\,a} = t_{TX} + T_a$. By applying this response position modulation, the CIR at the tags contains an ordered sequence of responses shifted based on *i)* the time of flight of each anchor to the tag, and *ii)* the time shift T_a.

The constant interval T is crucial. In Chorus, it relates *directly* to the target area and the time span of the measured CIR, and should be sufficiently large to avoid that two responses overlap in the CIR. We suggest using $T > 2d_{\text{MAX}}/c$, where d_{MAX} is the maximum distance between any two anchors. Information about the typical multipath delay spread in the target area also helps in determining a good value for T.

Figure 2 shows a CIR with 5 concurrent transmitters using $T = 128$ ns where a sequence of 5 main peaks ordered by the anchor IDs, and some significant MPC in between, can be clearly seen. Note how this technique also helps separating the response peaks from MPC from previous responses. By estimating the direct path of the first anchor, we can retrieve the anchor ID associated to each CIR peak based on the time shift w.r.t. the first main CIR peak.

3.3 Time of Arrival Estimation

Estimating the TOA of a UWB signal in real multipath environments is not trivial; several techniques exist, e.g., based on thresholds [9] or maximum likelihood estimation [6, 18]. In Chorus, accurate TOA estimation is critical; further, we need to extract from the CIR the TOA of N signals transmitted concurrently, instead of a single one. We describe now the information provided by the radio that is useful for our CIR analysis, and how we use it to develop our TOA estimation algorithm.

CIR estimation. The DW1000 measures the CIR upon packet RX with a sampling period $T_s = 1.016$ ns and stores it in a large (4096 B) internal buffer. The time span of the CIR is the duration of a preamble symbol: 1016 samples for a 64 MHz pulse repetition frequency (PRF) or 992 for a 16 MHz PRF. Each sample is a complex number whose real and imaginary parts are 16-bit signed integers.

First path index (`FP_INDEX`). The DW1000 performs a leading edge detection (LDE) algorithm on the CIR to detect the direct path (i.e., the TOA of the transmitted signal), and stores its CIR sample index (`FP_INDEX`) in the `RX_TIME` register [21]. LDE detects the direct path when the sampled amplitude first goes above a dynamically adjusted threshold, whose value is also made available by the DW1000. This threshold is based on two factors: *i)* a noise threshold η based on the estimated noise standard deviation σ_n of the CIR and *ii)* a peak multiplier that further increases the first path threshold based on the noise peak value. DecaWave, however, offers little detail about their LDE algorithm, and may employ additional mechanisms for direct path estimation.

Ordering the CIR buffer. Given the time shifts applied to anchor transmissions, in Chorus the `FP_INDEX` should contain the direct path of the first responder, i.e., the initiator. However, this is not necessarily true. For a single transmitter, we observed that the DW1000 arranges the CIR buffer so that `FP_INDEX` $\approx$ 750. However, in rare occasions, the DW1000 reports `FP_INDEX` to be at index 1015, i.e., the end of the buffer. For concurrent responders, we observed that the response located around sample $\approx$ 750 is the one associated to the successful anchor transmission, i.e., the one whose message is received, and not necessarily the first path from the initiator. For example, when using 5 concurrent responders with a time shift of 128 ns, this would cause the peaks of the last 2 responders to be (circularly) shifted at the beginning of the CIR buffer, which can confuse TOA estimation algorithms.

For these reasons, we exploit knowledge about the ID of the succesful sender reported by the DW1000 (i.e., the ID of the anchor whose response is received) and the known time shifts we apply (§3.2) to re-arrange the CIR buffer, placing the response of the initiator at the beginning of the buffer. In case of RX errors, for which there is no successful response, we use `FP_INDEX` and search back for possible previous responses using the noise threshold η and the time shift T. At the end of this procedure, the CIR buffer is ordered starting with the initiator response, followed by the assigned sequence of anchor responses, e.g., as depicted in Figure 2.

Dynamic threshold. To distinguish the anchor CIR pulses from noise and/or minor MPC from previous responses, we exploit the noise threshold $\eta = 12 \times \sigma_n$ used by the DW1000 in its LDE algorithm, where σ_n is the estimated noise standard deviation in the CIR measured during a packet reception. In certain occasions, however, we noticed that the DW1000 overestimates σ_n, providing abnormally high noise thresholds that prevent us from reliably detecting the anchor responses. Therefore, we constructed a distribution of σ_n from multiple CIR

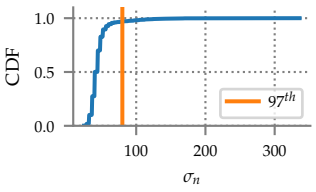


Figure 3: CDF of σ_n.

measurements with concurrent transmissions (Figure 3), and set a maximum value for σ_n as the 97^{th} percentile of the distribution. Hence, whenever the DW1000 outputs an abnormal measurement for σ_n, we correct it by setting σ_n to the 97^{th} percentile.

TOA estimation algorithm. To estimate the TOA of the direct paths from all responders, we adapt the Search and Subtract (SS) algorithm from [6]. This algorithm can be used to accurately estimate the direct path in a non-separable channel. We follow the original design, enforcing that considered pulses go over the noise threshold η. The key steps of our adapted algorithm are:

S1. Upsample the rearranged CIR using FFT with upsampling factor $L = 30$.

S2. Segment the CIR in chunks of length equal to the time shift T, to isolate the sub-CIRs from each responder.

S3. For each chunk, apply the SS algorithm with K iterations. Once the K strongest paths from the chunk have been estimated, select the direct path as the one with minimum time index, i.e., the earliest. Note that we consider the paths from SS only if they go over the noise threshold η.

S4. When all chunks corresponding to the number of responders N have been analyzed and the time indexes of the first paths obtained, subtract the assigned time shifts T_a from each anchor TOA and end the process.

3.4 Sources of Error

There are two main sources of error in Chorus localization estimates: *i)* the clock drift among concurrent transmitters, and *ii)* the transmission scheduling precision.

Clock drift. In Chorus, all anchors must schedule their TX to start with a given delay δ_{TX} after the initiator's POLL TX time t_p. However, the crystal oscillator of each anchor incurs a different clock drift e, causing an anchor a to transmit at $t_p + \delta_{TX}(1 + e_a)$, and inducing an error w.r.t. initiator i in the measured TDOA:

$$\hat{\Delta}t_{ai} - \Delta t_{ai} = \delta_{TX}(1 + e_a) - \delta_{TX}(1 + e_i) = \delta_{TX}(e_a - e_i) \quad (3)$$

Compared to SS-TWR [13, 20], the clock drift has twice more impact in Chorus. In the DecaWave EVB1000 platform we use, the crystal oscillator has a frequency offset of 10 ppm. Using $\delta_{TX} = 300\,\mu s$, the resulting error is 3 ns. We analyze the impact of clock drift in an indoor experiment with 4 nodes (3 anchors plus the initiator) placed in the corners of the INDOOR environment, used in our evaluation (§6). We time synchronize the anchors w.r.t. the initiator using the POLL message of the Chorus primitive, setting the initiator to transmit a POLL periodically every 100 ms. Anchors receive this message and send the RX timestamp to a central server computing the clock offset and drift of each node w.r.t. the initiator (node 1). Figure 4 shows the measured clock drift of the 3 anchors. Based on our results, and assuming a maximum clock drift of 3 µs/s and $\delta_{TX} = 300\,\mu s$, the TDOA error would be 0.9 ns. To reduce the effect of the clock drift and its impact on Chorus, the response delay must be then minimized by increasing the data rate and reducing the preamble length, the data payload, and the required processing time after receiving the POLL message from the initiator.

TX scheduling precision. The DW1000 UWB transceiver provides the capability to schedule delayed transmissions, however, it ignores the lowest 9 bits of the delayed TX time [21]. As a result, the chip is only able to schedule a packet TX with a precision of

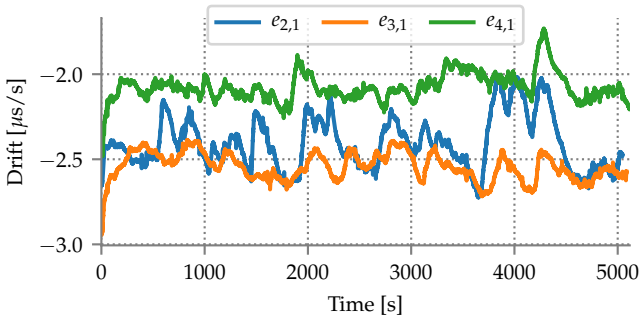


Figure 4: Clock drift.

$4/(499.2 \times 10^6) \approx 8$ ns. This directly affects concurrent transmissions from anchors, with an error $\epsilon \in [0, 8\,ns]$, which translates to a distance error up to ≈ 2.4 m. This is the most significant source of error in Chorus, whose impact we assess in our evaluation (§6).

4 FROM PROTOTYPE TO MODEL—AND BACK

The novelty of Chorus and the challenges outlined in §1 require a multi-pronged approach to its evaluation, exploiting both the realism and practical value of our prototype implementation as well as the abstraction power of models. Figure 5 outlines our approach.

The core of Chorus is constituted by the two colored components, executed on some processing hardware (e.g., a smartphone) connected with the UWB chip. The first component performs TOA estimation (§3.3) and anchor identification (§3.2) on CIRs generated by concurrent transmissions, and outputs the TOA value for each anchor. These data are fed to the actual localization computation, performed via our TDOA solver (§6.1).

These core components are the *same* for both our prototype and model-based evaluations; what is different is the nature of the CIRs input to them. In our DW1000-based prototype, concisely described in §6, the CIRs containing concurrent transmissions are acquired directly by the UWB transceiver. In our model-based evaluation, instead, they are generated by fusing the CIRs from isolated transmissions based on the model in §5 and on the same time shifts used in our prototype configuration. In this paper, we feed the model with two types of isolated CIRs. *Empirical* ones are acquired by the DW1000 via dedicated in-field experiments in the same environments where we evaluate our prototype. *Synthetic* ones are instead generated via the well-known IEEE 802.15.4 UWB model [24], and therefore free from the bias induced by specific hardware platforms *and* deployment environments.

Before delving in the evaluation, we describe our model for concurrent transmissions, which plays a central role in our findings.

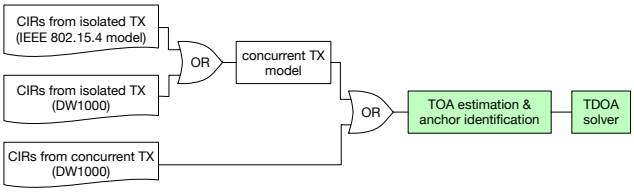


Figure 5: Prototype, model, and evaluation toolchain.

5 CONCURRENT TRANSMISSIONS MODEL

We model the impact of concurrent transmissions on channel impulse response (CIR) estimation.

Single Transmission. In the case of a single transmitter, the received signal $r(t)$ at the receiver can be modeled as

$$r(t) = s(t) * h_i(t) + n(t) \tag{4}$$

where $h_i(t)$ is the CIR, $s(t)$ is the transmitted signal, $n(t)$ added noise, and $*$ the convolution operation. The CIR $h_i(t)$ is

$$h_i(t) = \sum_{l=1}^{L} \gamma_l e^{-j\theta_l} \delta(t - t_l) \tag{5}$$

where L here is the total number of paths and γ_l, θ_l, and k_l are the attenuation, phase shift, and time delay associated to path l, respectively. In our case, as we are purely interested in the CIR estimation, we consider that the transmitted signal $s(t)$ corresponds to the preamble, which is the only part of the packet required to estimate the CIR at the receiver. The IEEE 802.15.4a standard [12] defines the preamble as a number of repetitions of a pre-defined symbol which is composed of single pulses (§2). By measuring the received signal $r(t)$ and knowing the preamble signal $s(t)$, the receiver can estimate the CIR $h(t)$ using, e.g., the *deconvolution* operation [27] or maximum likelihood-based methods [35].

Concurrent Transmissions. In the case of N signals transmitted concurrently by N transmitters, we consider an additive collision channel as in [31], according to which the received signal becomes

$$r(t) = \sum_{i=1}^{N} s_i(t) * h_i(t) + n(t) \tag{6}$$

where $s_i(t)$ and $h_i(t)$ correspond to the transmitted signal of sender i and the associated CIR. Hence, the received signal becomes a superposition of the convolution of the transmitted signals with the CIRs that model their propagation, plus the added noise.

In concurrent ranging [5], the concurrent transmitters employ the same preamble signal but incur different delays due to the distinct propagation time of the signal from each transmitter to the receiver. Similarly, if transmitters employ response position modulation [7] the signal is time shifted based on an assigned delay. In these cases, the transmitted signal of sender i becomes $s_i(t) = s(t - t_i)$. Due to commutativity [25] of the convolution operator of a linear time-invariant (LTI) system, we can simplify Eq. 6 to

$$r(t) = \sum_{i=1}^{N} s(t) * h_i(t - t_i) + n(t) \tag{7}$$

Applying the distributive property [25] of convolution on Eq. 7, we can obtain the overall CIR as

$$h(t) = \sum_{i=1}^{N} h_i(t - t_i) \tag{8}$$

This models the impact of concurrent transmissions on the CIR estimation and analytically explains the empirical observations found in previous concurrent ranging work [5, 7]. Using this model, we can estimate the resulting $h(t)$ when multiple $s_i(t)$ signals are transmitted concurrently.

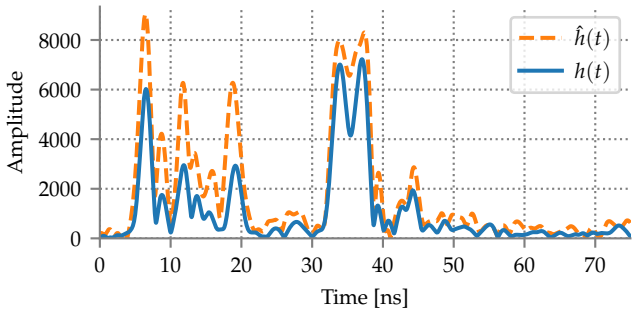


Figure 6: Comparison of the estimated CIR $\hat{h}(t)$ vs. the real $h(t)$ measured in an indoor environment with two concurrent responders at $d_1 = 4$ m and $d_2 = 8$ m from the receiver.

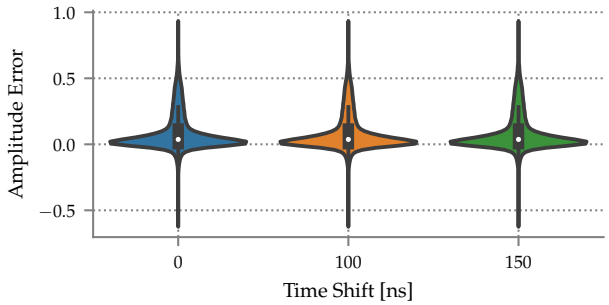
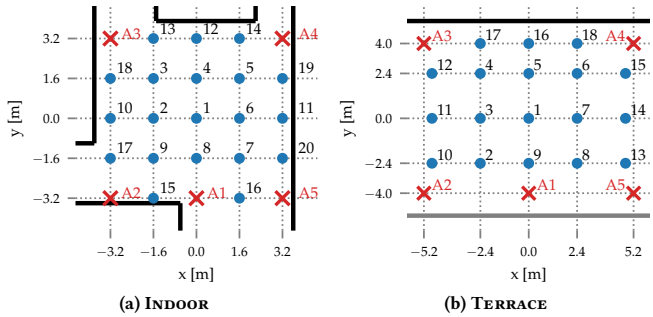


Figure 7: Model amplitude error.

Small-scale Validation. We validate our model in a small-scale experiment; the results we present in §6, where prototype experiments closely align with the model, can be considered a further, large-scale validation.

We deploy 3 nodes in a line along a corridor; the initiator and 2 anchors A_1 and A_2 respectively at $d_1 = 4$ m and $d_2 = 8$ m from the initiator. Nodes perform concurrent ranging as explained in §2. We consider 3 values (0 ns, 100 ns, and 150 ns) for the time shift T modulating the response of A_2 w.r.t. A_1. For each value of T, we collect 100 $h_1(t)$ and $h_2(t)$ signals with each anchor transmitting in isolation and 100 $h(t)$ signals with both anchors transmitting concurrently. Then, we merge the individual signals using Eq. 8, obtaining 100 modeled CIRs $\hat{h}(t)$. To compare $\hat{h}(t)$ against the measured $h(t)$, we upsample both signals by a factor $L = 30$, align them with a correction for the TX scheduling precision of DW1000, and measure the amplitude error at each time delay sample as $\frac{\hat{h}(t) - h(t)}{\max(h(t))}$ considering only the signal section of interest.

Figure 6 compares a CIR $\hat{h}(t)$ obtained using our model with the measured $h(t)$ with no artificial time shift. We observe that our model overestimates the amplitude of the peaks from the closest transmitter but captures well the CIR from the farthest. Figure 7 shows violin plots for each T value, with the measured amplitude error distribution across all merged signals. Our model is slightly conservative and tends to overestimate the amplitude of the direct paths and multipath components (Figure 6). Nonetheless, we observe that the model $\hat{h}(t)$ follows closely the measured $h(t)$, and serves our purpose of analyzing Chorus in §6 without the intrinsic limitations of current UWB transceivers.



(a) Indoor (b) Terrace

Figure 8: Evaluation deployments with fixed tag positions. Black lines represent walls and the grey line the metallic balustrade in Terrace.

6 EVALUATION

We present our experimental setup and main evaluation metrics (§6.1–§6.2) and validate our TOA estimation algorithm (§6.3) against the LDE algorithm executed on the DW1000. We analyze the performance of our Chorus prototype in two different environments (§6.4). Finally, we assess the full potential of Chorus using our models fed with empirical traces (§6.5) and CIR signals generated via the IEEE 802.15.4 UWB channel model (§6.6).

6.1 Experimental Setup

Environments. We perform our experiments in two different environments: *i)* Indoor, a $6.4 \times 6.4\ m^2$ area inside an office building, and *ii)* Terrace, a larger $10.4 \times 8\ m^2$ open area next to a building and surrounded by a wall on one side and a metallic balustrade on the other. Note that Terrace is by no means free of strong MPC and in fact presents larger power decay profiles than Indoor. In both environments, we deployed 5 anchors. Although 4 anchors are the minimum for TDOA, the extra one enables localization in cases where we are unable to detect one TOA estimate; recent research on UWB TDOA considers even higher number of anchors in similar areas [28]. We consider 20 and 18 different tag positions in Indoor and Terrace, respectively. About half of these positions actually represent a sort of worst-case, as they are placed on the boundary of the areas, on the line between anchors. Figure 8 shows both deployments along with the fixed tag positions. We set all UWB nodes at a 50 cm height from the ground.

UWB Settings. In all experiments, we set the DW1000 to use channel 7 with center frequency $f_c = 6489.6$ GHz and bandwidth[1] 1081.6 MHz. We use the shortest preamble length of 64 symbols with preamble code 17, the highest $PRF = 64$ MHz, and the highest 6.8 Mbps data rate. Finally, we set the response delay $\delta_{TX} = 280\,\mu s$.

Chorus Configuration. Unless otherwise stated, we use a time shift $T = 128$ ns for response position modulation (§3.2). This corresponds to a distance of 38.36 m, sufficiently higher than the maximum distance difference (13.12 m) among anchors. This shift also allows us to include and distinguish up to 7 anchor responses distributed over the time span of the CIR. For TOA estimation, we set the number of iterations of the Search and Subtract algorithm per CIR chunk to $K = 3$ and the noise threshold $\eta = 12 \times \sigma_n$ as per the default DW1000 configuration suggested by DecaWave [21].

Implementation. We implemented Chorus atop Contiki OS [4] using the DecaWave EVB1000 platform [19], equipped with the DW1000 transceiver, an STM32F105 ARM Cortex-M3 MCU, and its own PCB antenna. To collect the log files, we connect a laptop to the EVB1000 tag, which logs via the USB interface RX packets and errors along with the estimated CIRs. We collect 8 CIR signals per second as this requires reading over SPI and logging over USB a 4096B buffer. We run all analysis offline using Python and Matlab.

Positioning Solver. To determine the tag location, we implemented a non-linear least squares solver that uses the Trust Region Reflective algorithm [2]. Given our evaluation environments (Figure 8), our solver disregards outlier TDOA measurements $\Delta\hat{t}_{ai} > 50$ ns as they are not possible (even with the TX scheduling error of the DW1000) in our target areas. These rare outliers appear due to late and strong MPC from previous anchor responses in the CIR. Our solver is also able to choose other anchors besides the initiator as the time reference to compute the TDOAs and then the tag position. This is useful in case the initiator response cannot be detected but we have ≥ 3 TDOA estimates for localization.

6.2 Metrics

To evaluate Chorus, we look into two main metrics: *i)* the measured *TDOA error* and *ii)* the *localization error*. For each metric, we report statistics such as the average error μ and standard deviation σ as well as different percentiles for the absolute errors.

TDOA Error. We compute the TDOA error of each anchor a w.r.t. the reference (i.e., the initiator i) as $\Delta\hat{t}_{ai} - \Delta t_{ai}$, where $\Delta\hat{t}_{ai}$ is the TDOA value estimated by Chorus from the CIR and the Δt_{ai} the theoretical TDOA computed using Eq. 1. The TDOA error informs us about the precision of our TOA estimation algorithm and gives us indications about the localization accuracy attainable by Chorus.

Localization Error. For each CIR signal, we compute the absolute positioning error as $\|\hat{\mathbf{p}} - \mathbf{p_r}\|$, where $\hat{\mathbf{p}}$ is the position estimate obtained by Chorus and $\mathbf{p_r}$ is the known position of the tag.

Besides these two metrics, in our real-world experiments (§6.4) we also consider the *localization success rate* as the fraction of CIR signals where we obtain enough information (≥ 3 TDOA estimates) to compute the tag location. This last metric helps us understand the reliability and robustness of Chorus in real environments.

6.3 TOA Estimation Baseline

Before looking into the accuracy of Chorus, we compare the performance of our TOA estimation algorithm against the `FP_INDEX` provided by the LDE algorithm from DecaWave. This serves to validate our implementation and also as a TOA estimation baseline. To this end, we collected a total of 9262 non-concurrent CIR signals in Terrace, ≈100 CIR signals per anchor and tag position pair. Then, we estimated the TOA of the individual

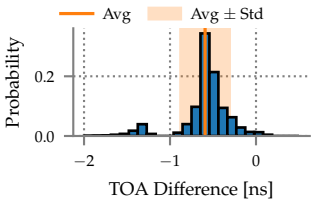


Figure 9: TOA algorithm estimation comparison.

[1]Note that the DW1000 receiver bandwidth is limited to 900 MHz.

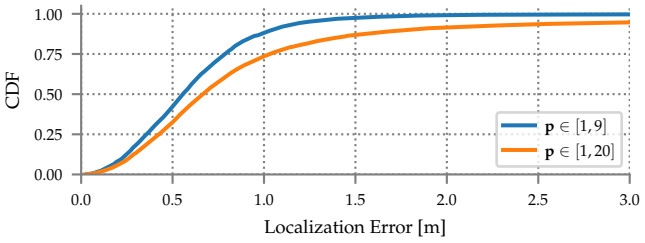
signals using our algorithm (§3.3), and measured the difference between our TOA estimate and the FP_INDEX from the DW1000.

Figure 9 shows a normalized histogram of the estimated differences. Compared to LDE, our algorithm underestimates the TOA by 0.59 ns on average. This is the result of conceptual differences between the algorithms. The LDE algorithm estimates the direct path as the first path whose amplitude goes over a dynamically adjusted threshold that depends on the measured noise in the CIR. This leads to measuring the direct path at different relative amplitudes of the first CIR pulse. Instead, our algorithm is based on peak detection, and selects the TOA at the beginning of the first CIR pulse. Overall, the standard deviation of the estimation difference is $\sigma = 0.29$ ns with a maximum absolute difference of 1.9 ns, demonstrating the ability of our algorithm to correctly estimate the TOA using $K = 3$ iterations. Changing K to 1 or 2 iterations yields a maximum difference of 42.4 ns and 26.1 ns, respectively. These significant errors are the result of strong and late MPC that are selected by the Search and Subtract algorithm as the direct path. These errors underline the importance of selecting $K \geq 3$.

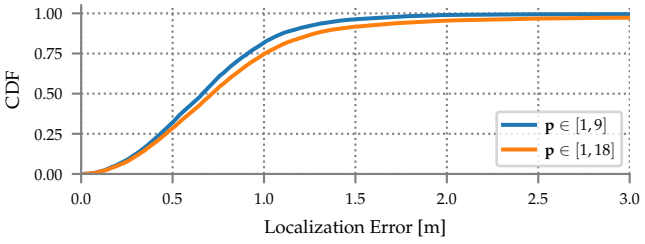
6.4 Prototype-based Evaluation

We report the results obtained with Chorus in Indoor and Terrace, establishing the performance achievable with today's commercially available UWB radios. For each tag position, we collected ≈515 CIR signals with 5 concurrent transmissions using $T = 128$ ns. This amounts to a total of 10294 signals in Indoor and 9263 in Terrace.

Localization Accuracy. Figure 10 shows the cumulative density functions (CDFs) of the localization errors obtained in both tested environments. Overall, Chorus achieves sub-meter localization error in 73.5% and 74.6% of the cases in Indoor and Terrace, respectively, despite the TX scheduling precision $\epsilon = 8$ ns of the DW1000. Recall that $\epsilon = 8$ ns can lead to a distance error of ≈ 2.4 m. However, we observe that performance increases considerably if we only consider the center tag positions $\mathbf{p} \in [1, 9]$, which are arguably the positions of interest for most applications. In these positions, Chorus yields sub-meter error in 88.28% and 81.7% of the cases in Indoor and Terrace, respectively. The CDFs converge with $99.4^{th} \approx 2.4$ m, i.e., the distance error produced by the TX scheduling precision ϵ. The remaining positions $\mathbf{p} \in [10, 20]$ can be considered as worst-case as they are on the border of the target area, where the distance difference among anchors increases and, in many cases, next to a wall, suffering from stronger MPC and increasing the localization error.



(a) Indoor



(b) Terrace

Figure 10: Chorus localization accuracy for different set of tag positions in Indoor (top) and Terrace (bottom).

TDOA Errors. In 96% of our TDOA estimates in both environments, the error is <8 ns, in line with the TX scheduling precision $\epsilon = 8$ ns. However, for 1% and 3% of our estimates in Indoor and Terrace, respectively, the error is >20 ns. These large errors are due to strong and late MPC, mostly in the stress positions on the borders of the target area. Recall that our algorithm chops the CIR into chunks of length T (i.e., the time shifts of the response position modulation) then obtains the time delay of the K strongest paths, selecting the earliest as the direct path. When there are large distance differences among anchors in environments with long power decay profiles (i.e., with late and strong MPC), as in Terrace, our algorithm could potentially select a multipath from a previous anchor response as the direct path. In Terrace, this occurs especially in the corner positions (10, 12, 13, 15) where the distance difference among anchors is large. Nonetheless, these large TDOA errors are rare and can generally be discarded by the solver as impossible; therefore, they bear reduced impact on localization accuracy.

Aggregating samples. The major source of error in Chorus is the TX scheduling precision $\epsilon = 8$ ns of the DW1000 (§3.4), which adds a random shift to the TDOA estimates. To compensate this error, we can aggregate the TDOA estimates from M signals, compute the median of each TDOA $\Delta\hat{t}_{ai}$, and pass these aggregated TDOA estimates to the solver. Figure 11 shows the localization accuracy when aggregating M signals in Indoor. By collecting only $M = 3$ signals before computing the location, we improve the 75^{th} percentile from 1.03 m to 0.78 m, obtaining sub-meter accuracy in 84% of the cases—including the worst-case border positions. In Terrace, we obtain similar results, e.g., with $M = 3$ and $M = 5$ samples, Chorus achieves sub-meter error in 85.83% and 89.2% of the cases, respectively. Overall, increasing M improves the accuracy, but requires more CIR signals, and therefore higher consumption and lower update rate. With $M = 5$, Chorus still requires less messages than SS-TWR to obtain a location estimate.

Localization Success Rate. To compute a tag location, Chorus first needs to accurately determine the TOA of the N signals transmitted concurrently by the anchors. However, we observed that

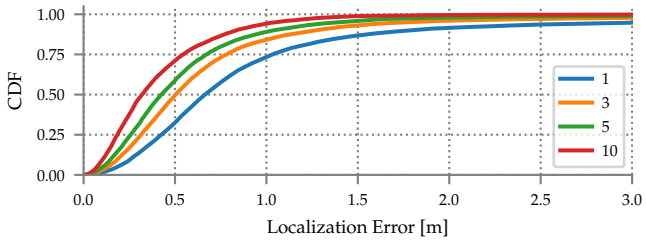


Figure 11: Localization error in Indoor after aggregating the TDOA samples of M CIR signals.

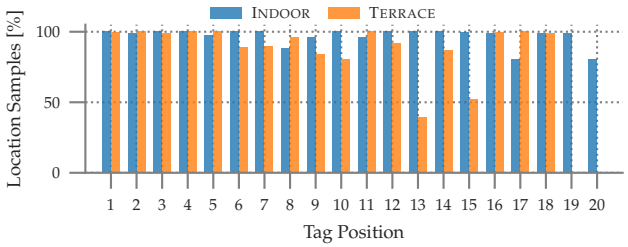


Figure 12: Chorus localization success rate per tag position. Note that in TERRACE we only have 18 tag positions.

sometimes the CIR peaks associated to a given anchor response *do not appear* in the CIR and therefore are not correctly detected. We see 3 possible causes: *i)* the CIR pulses corresponding to the anchor have low amplitude w.r.t. the noise floor and are disregarded based on our noise threshold η *ii)* the interference among concurrent transmissions cancels a given response from the CIR *iii)* the given anchor does not successfully receive the POLL message or fails to transmit the RESPONSE on time (very rare, if any). We observe this effect especially in the boundary positions of TERRACE, e.g., in position 13, we only detected 28% of the TOA estimates from anchor 2. Similarly, the detection rates are $\leq$ 78% for anchors 4 and 5 in position 10. In these positions, again, there is a large distance difference among anchors, resulting in different power loss and fading and making it difficult for the corresponding CIR peaks to emerge from noise and be detected. The failure to detect TOA estimates then reduces the available information to compute the tag position in the solver. Nonetheless, Chorus is able to estimate 94.79% and 88.37% of the expected TOAs in INDOOR and TERRACE, respectively. Figure 12 shows the resulting localization success rate (i.e., the percentage of signals where Chorus obtains $\geq$ 3 TDOA estimates) per tag position after discarding TDOA values > 50 ns. On average, Chorus achieves a localization success rate of 96.7% in INDOOR and 89.53% in TERRACE. Statistically, this means that in *difficult positions*, a tag may need to listen for more than a single message to compute its position reliably. Yet, the energy consumption of Chorus to estimate a location remains lower than, e.g., SS-TWR, where tags need to perform a two-way exchange with the N anchors.

Summary. Chorus achieves sub-meter localization error in $\geq$73.5% of the signals where it detects at least 4 TOA estimates despite the TX scheduling precision $\epsilon = 8$ ns of DW1000 and even in our challenging setup where half of the points are on the border. Aggregating the information from several signals reduces the localization error. Notwithstanding the adverse effects of self-interference, Chorus is able to provide a location estimate in 89% of the cases in TERRACE and 96.7% in INDOOR. Strong and late MPC combined with large distance differences between the anchors to the tag can produce significant, although rare, TDOA errors.

6.5 Model-based Evaluation: Empirical Traces

We now look into the performance of Chorus using our model (§5) fed with empirical CIR traces obtained in INDOOR and TERRACE. To this end, we first collected non-concurrent ≈100 CIR signals for each anchor and tag position combination in the two target environments. Then, we merged the individual CIR signals using Eq. 8 considering the time shift T and the different time of flight

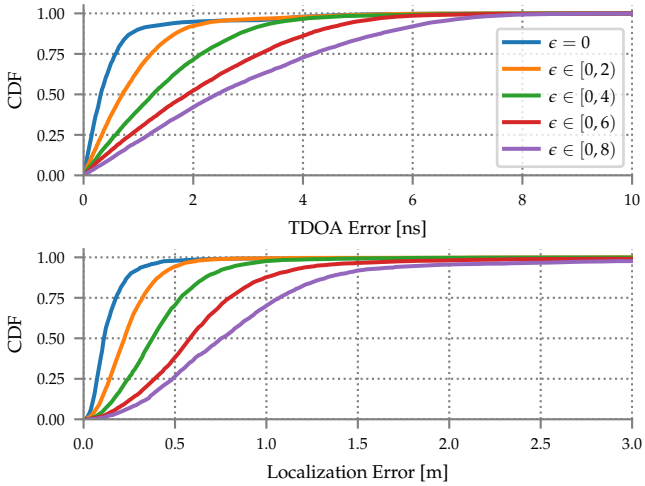


Figure 13: TDOA and localization errors using $T = 128$ ns and $K = 3$ iterations in INDOOR for different simulated transmission scheduling precisions ϵ.

from each anchor to the tag positions, obtaining 2049 CIR signals for INDOOR and 1810 for TERRACE with 5 concurrent transmissions. By using our model, we can easily assess the impact of *i)* the transmission scheduling precision ϵ and *ii)* the time shift T.

TX Scheduling Precision. We analyze first the impact of the TX scheduling precision ϵ of the DW1000 on TDOA and localization accuracy. To this end, when we merge the individual CIRs, we add a random time shift obtained from a uniform distribution $U[0, \epsilon)$ to each anchor response. This time shift simulates the TX scheduling precision. We change ϵ from 8 ns to 0 ns (i.e., no error) in steps of 2 ns, measuring the resulting TDOA and localization errors across the two datasets in INDOOR and TERRACE. We first focus on INDOOR.

Figure 13 shows the CDFs of the measured TDOA and localization errors in INDOOR for different scheduling precisions ϵ. The first thing to observe is that, for $\epsilon = 8$ ns (i.e., the precision of the DW1000), the CDF is very similar to the one obtained with our prototype (Figure 10a). This confirms that our *model faithfully captures the real-world behavior of concurrent transmissions*. With $\epsilon = 8$ ns, Chorus achieves $75^{th} = 4.2$ ns TDOA error, obtaining sub-meter error in 70.2% of the cases instead of 73.5% as in §6.4. With our generated signals, we noticed that the overestimated amplitude of MPC (§5) decreases the performance slightly w.r.t. our prototype.

Overall, increasing the TX scheduling precision lowers TDOA and localization errors. For instance, with $\epsilon = 2$ ns the TDOA $99^{th} = 4.99$ ns, obtaining sub-meter accuracy in 99.4% of the cases with $\mu = 0.26$ m and $\sigma = 0.25$ m. With no error, $\epsilon = 0$, Chorus yields $\mu = 0.14$ m with $\sigma = 0.15$ m. These results prove the potential of Chorus to provide decimeter-level localization by simply increasing the TX scheduling precision of the DW1000 chip. Fundamentally, this implies changing the frequency of the TX scheduling clock from 125 MHz to 250 MHz ($\epsilon = 4$ ns) or 500 MHz ($\epsilon = 2$ ns).

Now, we switch the attention to the results from TERRACE, which proved to be more challenging for our system. Figure 14 shows the localization error CDFs in this environment. Again, with $\epsilon = 8$ ns, the error curve is similar to that in our prototype (Figure 10b), providing sub-meter localization error in 71% of the cases. The

CDFs are also similar to the results obtained in INDOOR, but with slightly lower performance. With $\epsilon = 2$ ns and $\epsilon = 0$, Chorus achieves sub-meter error in ≥98.5% of the cases, ≈1% less than in INDOOR. In TERRACE, we observe that, for all considered ϵ, ≈2% of the TDOA errors are ≥ 20 ns with $99^{th} \geq 63$ ns. These large errors are outliers that appear due to strong and very late MPCs from previous responses. These MPC are the same ones that provided high maximum errors in our TOA baseline (§6.3) using $K \in \{1, 2\}$. In these situations, when we run Search and Subtract in a CIR chunk, we obtain the K strongest paths within the chunk, among which delayed MPCs from previous anchor responses are selected. Selecting the earliest path as the direct path (in this case the MPC from previous anchor) produces larger TDOA errors, which lead to higher localization errors $99^{th} = 1.4$ m with $\epsilon = 0$.

To tackle this issue, we analyze the impact of the response position modulation time shift T on the TDOA and localization errors.

Impact of the Time Shift. We generate new CIRs from the measured individual signals, changing the time shift T from the default 128 ns to 224 ns, with steps of 32 ns. We cannot use higher values with 5 concurrent transmitters as the 5^{th} response would then overlap in the same CIR chunk with the 1^{st} response due to the $\approx 1\,\mu s$ time span of the CIR. As previously mentioned, with $T = 128$ ns, 2% of our samples suffer from strong MPC from previous responses, producing higher TDOA and localization errors. As we increase the time shift, the percentage of samples with high errors decreases.

Table 1 depicts the localization error for the different time shifts T and $\epsilon = 0$. For all time shifts $T \geq 160$ ns, the error is <40 cm in 99% of the samples, demonstrating the potential of Chorus to reliably achieve decimeter-level localization by increasing the hardware scheduling precision. With the highest time shift $T = 224$ ns, however, the MPC from anchor 5 may affect the response of anchor 1 (initiator), resulting in slightly lower performance w.r.t. $T = 192$ ns. Despite increasing the time shift, we still have a few outliers that trigger the maximum TDOA error above 90 ns.

The best performance is with $T = 192$ ns, where the TDOA errors yield $95^{th} = 0.99$ ns and $99^{th} = 1.4$ ns. In this case, across 7240 TDOA estimates we have a total of 5 outliers (0.069%) for which the error is ≥52 ns. The remaining 7235 TDOA samples have an error ≤2 ns. This yields a *maximum* localization error of only 0.4 m, with $99^{th} = 0.32$ m and an average of 0.12 m. Based on our results, even using $\epsilon = 4$ ns still yields sub-meter localization error in 99% of the samples, which again would mostly require doubling the frequency of the TX scheduling precision of the radio chip.

Summary. Increasing the TX scheduling precision of the radio chip allows Chorus to achieve decimeter-level accuracy while providing in principle location information to countless users in the target area. The main challenge in real environments are late and very strong MPC from previous responses, which can be easily addressed by increasing the time shift T between consecutive responses. In this regard, if the CIR span offered by the hardware were increased to >1 µs, this would allow us to further extend T and provide opportunities for even more anchor concurrent transmissions.

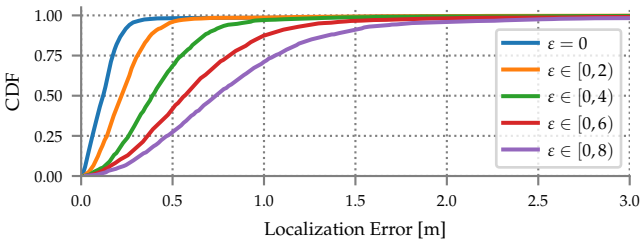


Figure 14: Localization error using $T = 128$ ns and $K = 3$ in TERRACE for different TX scheduling precisions ϵ.

Table 1: Localization errors in TERRACE for different time shifts T with $K = 3$ iterations and $\epsilon = 0$.

	Localization Error [m]							
T[ns]	μ	σ	50^{th}	75^{th}	90^{th}	95^{th}	99^{th}	Max
128	0.16	0.4	0.12	0.17	0.23	0.28	**1.41**	**10.13**
160	0.16	0.41	0.12	0.17	0.23	0.26	**0.4**	**8.92**
192	0.12	0.07	0.11	0.17	0.22	0.25	**0.32**	**0.4**
224	0.12	0.09	0.12	0.17	0.22	0.26	**0.32**	**1.49**

6.6 Model-based evaluation: Synthetic Traces

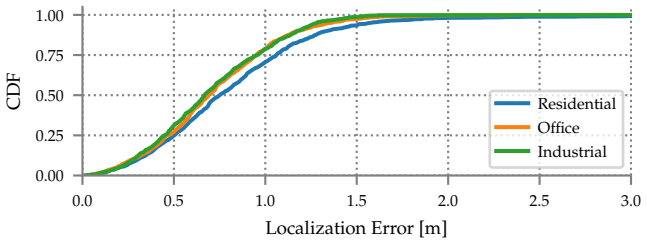
In §6.5, we fed the model in §5 with empirical traces gathered in exactly the same positions as the experiments in §6.4. This allowed us to successfully validate the model, and is a reasonable methodology for practical deployments. However, when many estimates across the target area are required (potentially spanning its entirety), the effort to gather the CIRs in all of them may become prohibitive.

Therefore, we turn our attention to the UWB channel model in the IEEE 802.15.4 standard [24], which is well-known, very detailed, offers configuration profiles for various target environments, and is available as Matlab code. Replacing the empirical traces with this model enables us to explore the performance of Chorus in many more points, without deployment effort. Further, and equally important, it allows us to validate our techniques on signals different from those generated *i)* by the DW1000 chip *ii)* only in our two deployments, therefore increasing confidence about the feasibility and robustness of Chorus.

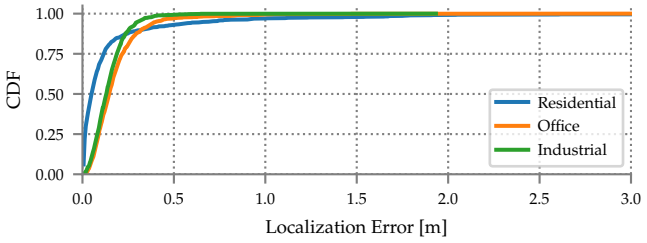
Methodology. We model the TERRACE placement (Figure 8b) and generate, for each anchor, 1000 CIRs from the model in [24]. We use the configuration profiles for parameters and power decay defined for residential, office, and industrial environments. We then select 1000 random positions in the target area and compute (§5) the fused CIRs resulting from concurrent transmissions.

Results. Figure 15 shows the localization error for different values of the TX uncertainty ϵ. Despite the remarkable differences in how the input CIR signals have been acquired, these results are comparable to those in §6.4–§6.5. For $\epsilon = 8$ ns, we achieve sub-meter accuracy in ≥ 78% of the cases, except for the residential profile (70.6%), due its very strong and delayed MPC. Nevertheless, even in this case the localization error is only $\mu = 0.83$ m. In general, the model in [24] yields[2] stronger MPC components w.r.t. the empirical traces we collected in our deployments. This complicates TOA estimation, requiring the use of $K = 10$ iterations of our algorithm (§3.3) to obtain the results shown. Finally, Figure 15b confirms that reducing or eliminating the TX scheduling error enables significantly higher accuracy.

[2]The model in [24] pre-dates the DW1000 by a decade; a validation of the former against the latter is outside the scope of this paper.



(a) $\epsilon = 8$ **ns**



(b) $\epsilon = 0$ **ns**

Figure 15: Localization error with the IEEE 802.15.4 UWB channel model [24].

7 CONCLUSION AND OUTLOOK

In this paper we *i)* propose Chorus, a novel and potentially revolutionary localization scheme for UWB *ii)* contribute techniques towards its practical realization and accurate modeling *iii)* evaluate its good performance with readily available UWB technology and model the one potentially attainable, and therefore ultimately *iv)* inspire the design of a next generation of UWB hardware designed to exploit the peculiar and beneficial tradeoffs unlocked by concurrent transmissions in general and Chorus in particular.

Chorus relies on a GPS-like scheme, hitherto unexplored in the context of UWB, where the localization target is a passive listener instead of an active transmitter. This change of perspective enables several advantages, notably including the ability for countless target devices to self-position in the area of interest.

The evaluation of our DW1000-based Chorus prototype shows that it achieves sub-meter accuracy in real-world environments. However, our analytical model for estimating the signal resulting from concurrent transmissions enables us to investigate the accuracy of Chorus beyond what enabled by UWB chips available today. By replacing the concurrent CIRs gathered in-field with synthetic ones, generated with two different techniques, we are able to accurately predict the current performance of our prototype and, therefore, simulate the one attainable were hardware limitations removed. Results show that an order of magnitude improvement in accuracy can be obtained via simple hardware modifications that cater for the needs of concurrent transmissions.

Prototype experiments and model analysis therefore reinforce each other, and are the basis from which to glimpse at what could be the future of Chorus and UWB concurrent transmissions.

Acknowledgments. This work is partially supported by EIT Digital via the D-TWIN project (18388).

REFERENCES

[1] P. Bahl and V. N. Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proc. of INFOCOM.*
[2] M. A. Branch, T. F. Coleman, and Y. Li. 1999. A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM Journal on Scientific Computing* 21, 1 (1999), 1–23.
[3] R. M. Buehrer and W. Venkatesh. 2011. Fundamentals of Time-of-Arrival-Based Position Locations. In *Handbook of Position Location: Theory, Practice and Advances, Chapter 6.* Wiley-Blackwell, 175–212.
[4] P. Corbalán, T. Istomin, and G. P. Picco. 2018. Poster: Enabling Contiki on Ultra-wideband Radios. In *Proc. of EWSN.*
[5] P. Corbalán and G. P. Picco. 2018. Concurrent Ranging in Ultra-wideband Radios: Experimental Evidence, Challenges, and Opportunities. In *Proc. of EWSN.*
[6] C. Falsi et al. 2006. Time of Arrival Estimation for UWB Localizers in Realistic Environments. *EURASIP J. on Advances in Signal Processing* 2006, 1 (2006), 032082.
[7] B. Großwindhager et al. 2018. Concurrent Ranging with Ultra-Wideband Radios: From Experimental Evidence to a Practical Solution. In *Proc. of ICDCS.*
[8] B. Großwindhager et al. 2019. SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags. In *Proc. of IPSN.*
[9] I. Guvenc and Z. Sahinoglu. 2005. Threshold-based TOA estimation for impulse radio UWB systems. In *Proc. of IEEE Int. Conf. on Ultra-Wideband.*
[10] D. Halperin et al. 2010. Demystifying 802.11N Power Consumption. In *Proc. of HotPower.*
[11] S. He and S. H. G. Chan. 2016. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surveys Tuts* 18, 1 (2016), 466–490.
[12] IEEE 2011. *IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs).* IEEE.
[13] Y. Jiang and V. C. M. Leung. 2007. An Asymmetric Double Sided Two-Way Ranging for Crystal Offset. In *Proc. of ISSSE.*
[14] B. Kempke et al. 2016. Harmonium: Asymmetric, Bandstitched UWB for Fast, Accurate, and Robust Indoor Localization. In *Proc. of IPSN.*
[15] B. Kempke et al. 2016. SurePoint: Exploiting Ultra Wideband Flooding and Diversity to Provide Robust, Scalable, High-Fidelity Indoor Localization. In *Proc. of SenSys.*
[16] B. Kempke, P. Pannuto, and P. Dutta. 2015. PolyPoint: Guiding Indoor Quadrotors with Ultra-Wideband Localization. In *Proc. of HotWireless.*
[17] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. In *Proc. of SIGCOMM.*
[18] J. Lee and R. A. Scholtz. 2002. Ranging in a Dense Multipath Environment Using an UWB radio link. *IEEE J. Sel. Areas Commun.* 20, 9 (Dec 2002), 1677–1683.
[19] DecaWave Ltd. 2013. DecaWave ScenSor EVB1000 Evaluation Board.
[20] DecaWave Ltd. 2014. APS011 Application Note: Sources of Error in DW1000 based Two-way Ranging (TWR) Schemes.
[21] DecaWave Ltd. 2016. DW1000 User Manual.
[22] D. Lymberopoulos and J. Liu. 2017. The Microsoft Indoor Localization Competition: Experiences and Lessons Learned. *IEEE Signal Processing* 34, 5 (2017).
[23] C. McElroy, D. Neirynck, and M. McLaughlin. 2014. Comparison of wireless clock synchronization algorithms for indoor location systems. In *Proc. of ICC.*
[24] A. F. Molisch et al. 2004. IEEE 802.15. 4a channel model—final report. *IEEE P802* 15, 04 (2004), 0662.
[25] A. Oppenheim, A. Willsky, and S. Nawab. 1996. *Signals & Systems (2nd Ed.).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
[26] P. Pannuto, B. Kempke, and P. Dutta. 2018. Slocalization: Sub-µW Ultra Wideband Backscatter Localization. In *Proc. of IPSN.*
[27] M. Pourkhaatoun and S. A. Zekavat. 2011. A Review on TOA Estimation Techniques and Comparison. In *Handbook of Position Location: Theory, Practice, and Advances, Chapter 7.* Wiley-Blackwell, 213–243.
[28] J. Tiemann et al. 2016. ATLAS - An Open-Source TDOA-based Ultra-Wideband Localization System. In *Proc. of IPIN.*
[29] J. Tiemann et al. 2016. Multi-User Interference and Wireless Clock Synchronization in TDOA-based UWB Localization. In *Proc. of IPIN.*
[30] D. Vasisht, S. Kumar, and D. Katabi. 2016. Decimeter-level Localization with a Single WiFi Access Point. In *Proc. of NSDI.*
[31] M. Wilhelm et al. 2014. On the Reception of Concurrent Transmissions in Wireless Sensor Networks. *IEEE Trans. Wireless Commun.* 13, 12 (Dec 2014).
[32] M. Z. Win and R. A. Scholtz. 1998. Impulse radio: How it works. *IEEE Comm. Lett.* 2, 2 (1998), 36–38.
[33] M. Z. Win and R. A. Scholtz. 1998. On the Energy Capture of Ultrawide Bandwidth Signals in Dense Multipath Environments. *IEEE Comm. Lett.* 2, 9 (1998), 245–247.
[34] J. Xiong, K. Sundaresan, and K. Jamieson. 2015. ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization. In *Proc. of MobiCom.*
[35] L. Yang and G. B. Giannakis. 2004. Ultra-wideband Communications: An Idea Whose Time Has Come. *IEEE Signal Processing* 21, 6 (2004), 26–54.
[36] M. Youssef and A. Agrawala. 2005. The Horus WLAN location determination system. In *Proc. of MobiSys.*