SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags

Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer
Department of Electrical and Information Engineering, Graz University of Technology, Austria
{grosswindhager, michael.stocker, mrath, cboano, roemer}@tugraz.at

ABSTRACT

A large body of work has shown that ultra-wideband (UWB) technology enables accurate indoor localization and tracking thanks to its high time-domain resolution. Existing systems, however, are typically designed to localize only a *limited number* of tags, and involve the exchange of *several messages* following a given schedule. As a result, the scalability of current solutions in terms of *tag density* is limited, as well as their efficiency and responsiveness. In this paper, we present SnapLoc, a UWB-based indoor localization system that allows an unlimited number of tags to self-localize at a theoretical upper bound of 2.3 kHz. In SnapLoc, a tag obtains the responses from multiple anchors *simultaneously.* Based on these signals, the tag derives the time difference of arrival between anchors and estimates its position. Therefore, SnapLoc does *not* require tags to actively transmit packets, but to receive only a single message. This allows tags to passively localize themselves and ensures that the performance of SnapLoc does not degrade with high node densities. Moreover, due to the (quasi-)simultaneous responses, a tight clock synchronization between anchors is not needed. We have implemented SnapLoc on a low-cost platform based on the Decawave DW1000 radio and solved limitations in the transceiver's timestamp resolution to sustain a high localization accuracy. An experimental evaluation shows that SnapLoc exhibits a 90% error and median error of 33 cm and 18 cm, respectively, hence enabling decimeter-level accuracy at fast update rates for countless tags.

CCS CONCEPTS

• **Computer systems organization** → *Embedded and cyber-physical systems*; • **Networks** → Location based services;

KEYWORDS

Localization, TDOA, ultra-wideband, channel impulse response

ACM Reference Format:
Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. 2019. SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags. 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/10.1145/3302506.3310389

1 INTRODUCTION

Ultra-wideband (UWB) technology is becoming increasingly popular thanks to its robustness and outstanding localization accuracy. Spreading the signal over a wide bandwidth, indeed, results in: (i) greater immunity to multipath fading, (ii) better interference mitigation, (iii) higher throughput, as well as (iv) an improved timing resolution allowing for accurate localization and tracking [13].

Such a high time-domain resolution allows UWB-based solutions to significantly outperform narrowband RF technologies like Bluetooth Low Energy and Wi-Fi in terms of localization accuracy. These technologies, indeed, can hardly achieve a sub-meter accuracy [5, 18], and are hence unable to satisfy the requirements of location-aware Internet of Things (IoT) applications such as assisted living [42], robot navigation [15, 22], and smart manufacturing [20].

Because of the aforementioned properties, several works have used UWB technology to build indoor localization and tracking systems [1]. These works have shown that UWB-based solutions can achieve a localization accuracy up to a few cm [30], even in challenging conditions [24] and despite the use of a single anchor [14].

Existing solutions do not scale. Unfortunately, most of the existing solutions based on UWB technology focus on achieving a high localization accuracy, often disregarding properties such as *multi-tag support* and *high update rates* [33]. As a result, current systems typically support only a few tags and do not scale in terms of *tag density*, due to (i) the large number of messages exchanged, and (ii) the use of scheduling techniques for collision avoidance [33].

Large message overhead. A large number of UWB systems are indeed based on time of flight (TOF) techniques and make use of two-way ranging (TWR) schemes or a variant of it [24, 32]. These systems require the exchange of several consecutive messages, such that a mobile tag can derive the distance from multiple anchors and unambiguously determine its position. The large number of messages exchanged to carry out each distance estimation limits the overall update rate [25] and requires a tag to be heavily involved in the communication, which increases its energy-consumption. As mobile tags are typically battery-powered, their radio-on time should, instead, be minimized, in order to preserve their limited energy budget. Furthermore, sequentially estimating the distance to each anchor leads to inconsistent measurements in mobile and highly-dynamic settings (as one combines distances estimated at slightly different times), which limits the achievable accuracy.

Use of collision avoidance techniques. To reduce message overhead and avoid exchanging consecutive messages, a few UWB systems employ time difference of arrival (TDOA) techniques and allow a tag to broadcast only one message per position estimate [34, 41]. The latter is received from synchronized anchors, which compute the TDOA and communicate back the estimated position to the tag.

IPSN '19, April 16–18, 2019, Montreal, QC, Canada
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6284-9/19/04…$15.00
https://doi.org/10.1145/3302506.3310389

Whilst this allows to minimize the number of transmissions carried out by a tag and to shift the computational burden to more powerful anchors, one still needs to allocate *specific timeslots* to each tag in order to avoid collisions. Such scheduling techniques, which are also needed in systems based on TOF [24, 32] and single anchors [14], however, limit the number of tags that can be supported and, consequently, the scalability of a localization system.

Need for a tight synchronization. TDOA-based systems typically require anchors to be synchronous. For example, in anchor-initiated solutions [28], the tags estimate their position based on signals received from synchronized anchors. Whilst this approach allows tags to carry out self-localization without the need to transmit information, it requires a tight (ns-range) synchronization between anchors. However, this results in an overhead [41, 44] and is challenging [43]. Furthermore, the anchors still send messages sequentially, which requires also the tag to compensate for clock deviations [28].

Concurrent ranging still immature. Recent work on concurrent ranging has the potential to significantly reduce message overhead by exploiting simultaneous responses to a ranging request issued by an initiator [6]. However, concurrent ranging is still inapplicable in practical UWB systems due to: (i) the inability to identify responders and to discern them from strong multipath components [12], (ii) the high amount of payloads lost when responders are located at similar distances, as well as (iii) the limited transmit timestamp resolution of off-the-shelf UWB transceivers (see Sect. 2.3).

This state of affairs represents a significant problem, as increasing the density of tags in existing UWB systems results in a significant reduction of the localization update rate [26, 34, 37], due to the large message overhead and the use of collision avoidance techniques. In order to create UWB-based indoor localization systems that scale regardless of the number of tags, one would ideally (i) address the aforementioned limitations of concurrent ranging, and (ii) apply the latter to TDOA-based anchor-initiated approaches, in such a way that anchors are not required to be tightly synchronized. This would keep tags away from actively transmitting messages, minimize their radio-on time, and avoid the use of collision avoidance schemes. More importantly, this would enable an *unlimited number* of tags to passively self-localize at *fast update rates*.

Contributions. In this paper we present SnapLoc, a UWB-based indoor localization system that achieves exactly this. SnapLoc solves the limitations of existing concurrent ranging techniques and allows tags to obtain responses from multiple anchors *simultaneously*. Instead of scheduled sequential anchor messages [28], in SnapLoc multiple anchors reply *(quasi-)simultaneously* to an initialization message sent by a reference anchor. Based on these responses, a tag can quickly derive the time difference of arrival between anchor pairs and accurately estimate its position.

Therefore, in SnapLoc, a tag does not require to actively transmit messages, and its radio-on time can be reduced to a *single* read operation. This removes the need for a tight clock synchronization between anchors and eliminates the clock correction at the tag completely. Furthermore, SnapLoc's approach allows tags to passively localize themselves, ensuring that the performance does not degrade with high node densities. Theoretically, SnapLoc requires just 434 μs to provide the tag with all the necessary information to estimate its location. Thus, SnapLoc enables an unlimited number of tags to self-localize at position update rates up to 2.3 kHz.

Besides the reception of a single packet, a key property of SnapLoc is the use of information that is only contained in a packet's preamble for computing the actual position (see Sect. 3). In fact, a tag extracts the (quasi-)simultaneous responses from the anchors by analyzing the estimated channel impulse response (CIR) provided by standard-compliant UWB transceivers upon reception of a preamble. This avoids the need to correctly receive a payload and bypasses an intrinsic limitation of concurrent ranging.

To associate each response in the CIR to the correct anchor and to counteract the impact of strong multipath components, SnapLoc assigns an individual delay in the nanosecond range to each anchor (see Sect. 4). This, however, limits the maximum number of anchors due to the finite length of the estimated CIR. To overcome this restriction, we propose to partition the area of operation in multiple cells and let each cell operate using orthogonal *preamble codes*.

We have implemented SnapLoc on a low-cost platform based on the Decawave DW1000 radio (see Sect. 4.5). As discussed in Sect. 2.3, this transceiver constrains the transmit timestamp resolution to 8 ns, which severely limits the accuracy. We devise two techniques to restore a resolution of 15.65 ps and 1 ns, respectively, allowing SnapLoc to sustain a high localization accuracy (see Sect. 5).

An experimental evaluation in a common office and in a larger laboratory classroom shows that SnapLoc exhibits a 90% error and a median error in the order of 33.7 cm and 18.4 cm, respectively. Therefore, SnapLoc enables decimeter-level localization accuracy at fast update rates for an unlimited number of tags.

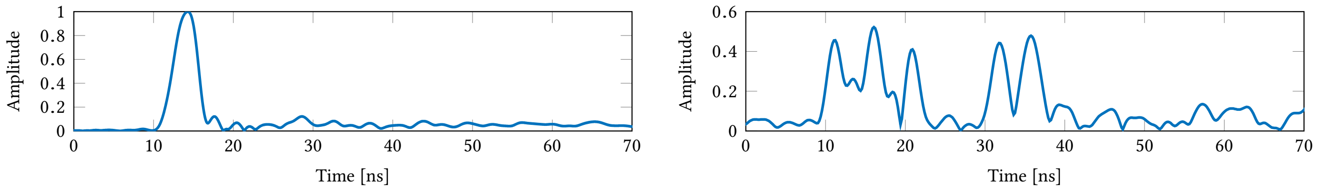
After describing the limitations of existing approaches in Sect. 2, this paper makes the following contributions:

- We introduce the design of SnapLoc, a UWB-based indoor localization system that allows an unlimited number of tags to self-localize at very high update rates (Sect. 3);
- We describe SnapLoc's principle, and detail on: how to reliably detect the anchors' responses, how to associate each response to the corresponding anchor, as well as how to derive the TDOA and compute a tag's position (Sect. 4);
- We implement SnapLoc on a low-cost platform based on the DW1000 radio and propose a technique to overcome the transceiver's limited transmit timestamp resolution (Sect. 5);
- We evaluate SnapLoc in common office environments and show that it enables decimeter-level localization accuracy at fast update rates also for high tag densities (Sect. 6);

After describing related work in Sect. 7, we conclude our paper in Sect. 8, along with a discussion on future work.

2 LIMITATIONS OF EXISTING APPROACHES

We begin our discussion by describing existing UWB-based localization approaches and highlighting their limitations. We first point out how solutions based on TOF and two-way ranging schemes typically incur a large message overhead (Sect. 2.1). We then discuss how TDOA-based approaches require a tight synchronization across anchors, which is complex to attain and may introduce errors lowering the localization accuracy (Sect. 2.2). Finally, we discuss recent work on concurrent ranging, and elaborate on the limitations that make it still inapplicable to practical UWB systems (Sect. 2.3).



(a) CIR in an environment with limited multipath

(b) CIR in the presence of strong multipath components

Figure 1: Example of CIRs estimated by a UWB transceiver. Whilst the first path component is prominent in an environment with limited multipath (a), the same does not necessarily hold true in the presence of strong multipath components (b).

2.1 TOF-based Approaches

Time-based localization systems rely on measuring the travel time of a radio signal between two nodes (typically an *anchor* and a *tag*). Among time-based systems, the most popular approaches are *time of flight* (TOF) and *time difference of arrival* (TDOA).

TOF-based systems – often also referred to as time of arrival based (TOA-based) systems – determine the absolute distance between sender and receiver by measuring the time of flight of a packet multiplied with the propagation speed (i.e., speed of light in air c). Allowing a tag to passively self-localize in a TOF scheme (i.e., performing only one-way communications from anchors to tag) requires anchors and tags to be tightly synchronized, which implies a significant overhead and is often infeasible [44]. To avoid this, several UWB systems let a tag estimate the TOF from multiple anchors by making use of two-way ranging (TWR) or similar schemes [24, 32]. These schemes do not require synchronization between anchors and tags, but envisage the exchange of multiple messages, such that a tag can derive its distance to several anchors.

Limitations. Such an approach incurs a large communication overhead, considering that at least three (2D) or four (3D) anchors are necessary to unambiguously determine a tag's position. Furthermore, common systems typically make use of up to eight [28, 41] or fifteen [22] anchors to increase the redundancy and robustness of localization, e.g., to mitigate non-line-of-sight (NLOS) conditions. This may results in tens of messages transmitted and received by a tag for each localization attempt, which limits the achievable update rate [25] and heavily affects its energy consumption.

2.2 TDOA-based Approaches

TDOA approaches do not require the absolute time of flight of a packet, but exploit the difference Δt in the arrival time of a signal at two reference points. Based on Δt, the difference in distance Δd between tag and reference points can be calculated as $\Delta d = \Delta t \cdot c$.

The advantage of TDOA approaches is that sender and receiver do not need to be synchronized, as the offset of the tag's clock is canceled out [8]. This simplifies the system design and removes the need of exchanging several messages between tags and anchors.

Indeed, in most UWB-based TDOA localization systems, a tag broadcasts only one message per position estimate [34, 41]. The tag's position is then estimated in a central localization engine computing the TDOA, which allows to shift the computational burden from the tag to other (more powerful) devices [35]. This is especially advantageous in applications where a central entity monitors the position of all users, and where the tags do not necessarily need to know their own position, e.g., asset- or sports tracking [4, 29].

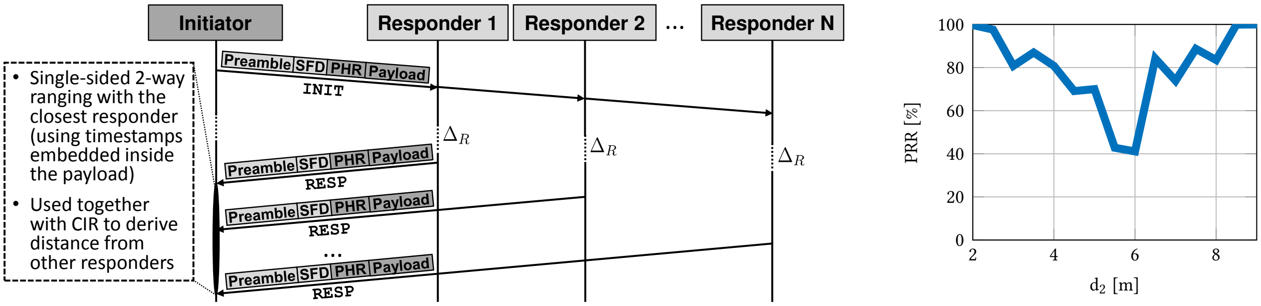
Limitations. Whilst such an approach minimizes the number of transmissions carried out by tags, one still needs to allocate *specific timeslots* to each tag in order to avoid collisions. Such scheduling techniques, however, limit the number of tags that can be supported and, consequently, the scalability of the system. TDOA-based approaches allowing tags to carry out passive self-localization exist [28], in which synchronized anchors subsequently send signals that are received by a tag to estimate the time differences. However, besides the need for a tight ns-range synchronization between anchors (which is hard to achieve [43], and increases message overhead [41, 44]), one needs to correct the tags' clock skew due to the long reception phase of the sequential messages.

2.3 Concurrent Ranging

Corbalán and Picco [6] have recently introduced the concurrent ranging primitive, which enables the simultaneous distance estimation between an initiator and an arbitrary number of responders. By doing so, concurrent ranging potentially allows to reduce the number of messages required to estimate the distance from N neighbors to a single transmit and receive operation.

Channel impulse response (CIR). Concurrent ranging exploits the CIR estimated by standard-compliant UWB transceivers, such as the Decawave DW1000, to extract simultaneous responses from an arbitrary number of nodes. Fig. 1a shows an exemplary CIR estimated with the DW1000 radio in an environment with limited multipath: one can clearly note the first path or line-of-sight (LOS) component. The latter is typically used to precisely estimate the arrival time of a packet (and consequently the distance between two nodes). A CIR further contains information about the multipath propagation consisting of reflections from surfaces as well as scattering. This feature has been exploited, among others, to derive the presence of NLOS conditions [31], destructive interference [13], as well as to perform multipath-assisted single-anchor localization [14, 27].

Principle of operation. To perform concurrent ranging, an initiator broadcasts an `INIT` message to all its neighbors (responders), who answer simultaneously with a `RESP` message after a constant delay Δ_R, as shown in Fig. 2a. The principle foresees the computation of the distance to the closest neighbor using single-sided two-way ranging. Thus, it is assumed that the timestamp included in the payload of the closest neighbor's `RESP` message is reliably detected. After completing this step, one can estimate the distance to all other responders by analyzing the CIR of the received `RESP` message.



(a) Concurrent ranging principle

(b) Reliability in decoding packets

Figure 2: Concurrent ranging principle (a) and reliability in decoding packets when keeping a first responder at a fixed distance d_1 = 5 meters and varying the distance d_2 of a second responder (b). Up to 60% of the packets are not received correctly.

Limitations. Unfortunately, concurrent ranging is currently inapplicable in practical UWB systems due to: (i) the inability to identify responders and to discern them from strong multipath components, (ii) the high amount of payloads lost when responders are located at similar distances, as well as (iii) the limited transmit timestamp resolution of common UWB radios, as discussed next.

Identifying responders. One of the key challenges hindering the feasibility of concurrent ranging in real-world systems is the inability to associate a distance estimate to a specific responder. Corbalán and Picco have shown the feasibility of concurrent ranging in artificial setups where the initiator had prior knowledge about the order in which the signal peaks associated to the responders are received in the CIR [6]. In practical situations, however, one does not know the relative locations of nodes beforehand [12].

Discerning responses from strong multipath components. In multipath-rich indoor environments, several strong multipath components (MPCs) may appear in the CIR and overlap with the concurrent responses. Fig. 1b, for example, shows a CIR estimated by the DW1000 radio in a University office while a single responder answers with a RESP message. One can clearly identify five peaks, four of which correspond to MPCs that have an amplitude similar to the LOS component. The presence of such strong MPCs makes it impossible to differentiate between a desired response and a strong multipath component due to reflections from walls and solid surfaces.

Unreliability of correctly decoding timestamps. Concurrent ranging assumes that the timestamp included in the payload of the closest neighbor's RESP message is reliably received. However, the probability to lose a packet or to decode a corrupted payload is very high when one or more responders are located at similar distances. To illustrate this problem, we perform an evaluation in a corridor using one initiator node and two concurrent responders R_1 and R_2, all using the DW1000 radio. R_1 is placed at a fixed position whose distance from the initiator is d_1 = 5 m. We execute different measurements while varying the distance of R_2 between $d_2 = 2, \ldots, 9$ m in steps of 50 cm. For each step, we perform 1000 concurrent rangings and log the number of RESP messages successfully decoded at the initiator, which we denote as packet reception rate (PRR). Fig. 2b shows the PRR as a function of d_2: concurrent ranging as suggested in [6] does not perform reliably when two responders are close to each other. In practice, the PRR would decrease even further if more than two responders are located at a similar distance.

Limited transmit timestamp resolution. As shown in Fig. 2a, all responders dispatch a RESP message after a constant delay Δ_R. To this end, one can use the *delayed transmission* feature of the Decawave DW1000 radio. The latter enables to set a future timestamp at which the transceiver actually sends a RESP message. This allows to align a pre-calculated timestamp with the real transmit timestamp and embed it in the message being transmitted. Unfortunately, the Decawave DW1000 ignores the low-order 9 bits of the timestamp, limiting the transmission resolution to approximately 8 ns [9, p. 26]. This is not an issue in the classical single-sided two-way ranging scheme, as the real transmit timestamp is anyway embedded in the message. However, this aspect has a severe impact on the precision of concurrent ranging, as it negatively affects the concurrency of the RESP messages of the neighbors.

SnapLoc mitigates the aforementioned limitations of concurrent ranging and applies a modification of the latter to a TDOA-based approach, allowing the creation of an indoor localization system that scales regardless of the tags density, as elaborated in Sect. 3.

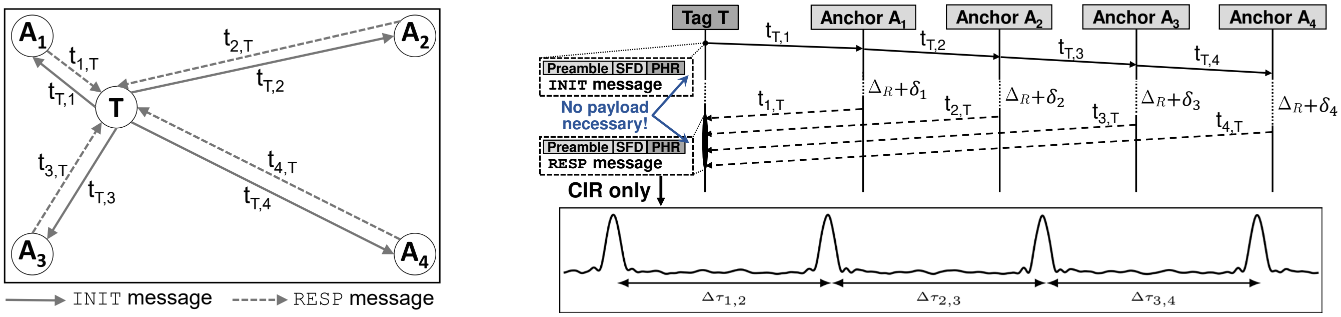
3 SNAPLOC: DESIGN RATIONALE

In SnapLoc, we tackle the limitations of concurrent ranging and allow tags to reliably obtain and identify simultaneous responses from multiple static anchors. To this end, we assign an individual delay in the nanosecond range to each anchor, which avoids misclassification of responses due to overlapping responses or multipath components (Sect. 3.1). This allows tags to derive the TDOA between anchors by *only* reading and analyzing the CIR. Hence, it removes the need to carry out a single-sided two way ranging and to correctly receive the timestamp embedded in a RESP payload (Sect. 3.2). We finally show how embedding these key principles into a TDOA-based anchor-initiated approach allows to create a scalable UWB-based localization system (Sect. 3.3).

3.1 Correctly Identifying Multiple Responses

As discussed in Sect. 2.3, concurrent ranging fails in situations where responders are located at a similar distance from the tag. Furthermore, in multipath-rich indoor environments, several strong MPCs may be present and overlap with responses from the anchors, making it hard to correctly recognize desired anchor responses.

To address this problem, instead of making use of just a fixed Δ_R as in Fig. 2a, we set an additional individual delay δ_i for each



(a) Tag initiating the position estimation

(b) Deriving the TDOA between anchors using CIR information only

Figure 3: Tag initiating the position estimation by sending an INIT message to the surrounding anchors, who respond simultaneously (a). Based *only* on the CIR embedded in the response, the tag can derive the TDOA between anchors (b).

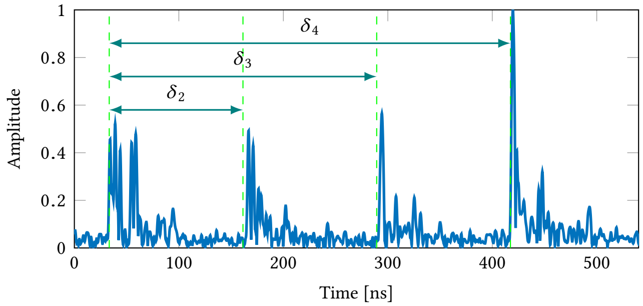


Figure 4: Introducing an additional individual delay δ_i for each responder A_i allows to identify responses and discern them from strong multipath components.

responder A_i in the nanosecond range. Consequently, the anchors do not respond simultaneously, but rather quasi-simultaneously. This allows to obtain responses that are separated in time and spread over a wider range of the CIR, as well as to avoid the overlap of MPCs and desired responses. Fig. 4 shows the resulting CIR with four responders: although the first response exhibits a peak due to a strong MPC, it is possible to distinguish it from the other responses, thanks to the additional individual delay δ_i.

3.2 Exploiting CIR Information Only

The approach described in Sect. 3.1 allows tags to seamlessly derive the TDOA between anchors by *only* reading and analyzing a *single* CIR – a novel approach allowing ultra-fast TDOA estimations. Fig. 3a illustrates a scenario with four anchors A_i ($i = 1 \ldots 4$) and one tag T. The latter broadcasts an INIT message that is received by all anchors (solid arrows), which simultaneously respond with a RESP message after a constant delay $\Delta_R + \delta_i$ (dashed arrows).

Due to the individual delay δ_i and the different TOF, the responses in the CIR are separated in time, as shown conceptually in Fig. 3b. The distances of the responses $\Delta\tau_{i,j}$ in the CIR contain position-related information of the tag, namely the TDOA between the anchors A_i and A_j ($i \neq j$):

$$\Delta\tau_{i,j} = \delta_j - \delta_i + 2 \cdot (t_{j,T} - t_{i,T}). \tag{1}$$

Given that the individual delay δ_j of anchor A_j is known, the time difference of arrival $\Delta t_{i,j}$ follows as:

$$\Delta t_{i,j} = t_{j,T} - t_{i,T} = \frac{\Delta\tau_{i,j} - (\delta_j - \delta_i)}{2}. \tag{2}$$

Note that this approach removes the need to carry out a single-sided two way ranging and to correctly receive the timestamp embedded in a RESP payload – one of the key limitations outlined in Sect. 2.3. Therefore, as highlighted in Fig. 3b, one can estimate the TDOA between anchors using *only* information that is contained in the CIR estimated from a *single* read operation.

3.3 Allowing the System to Scale

The novel approach described in Sect. 3.2 allows an ultra-fast estimation using only information contained in a single CIR. In principle, by having the tag initiating the localization, this approach allows tags to trigger a position update individually and aperiodically[1]. However, it requires a tag to initiate the location estimate by actively sending an INIT message. In order to avoid collisions between tags, one would hence still need to allocate *specific timeslots* to each tag, as well as elect one anchor responsible to periodically broadcast the fixed position and ID of all involved anchors. This would decrease the scalability of the system, as described in Sect. 2.2.

Therefore, we design SnapLoc as an *anchor-initiated* approach in which an anchor is selected to act as the initiator broadcasting the INIT message (called *reference anchor* or A_{ref} in the remainder of this paper). The key advantage of such an anchor-initiated approach is that the tag is not actively involved in the communication and thus no scheduling between multiple tags is required. Furthermore, similar to Global Navigation Satellite Systems (GNSS), this approach allows *passive self-localization*. This enables tags to remain anonymous and maximize their privacy, as well as to achieve a high scalability regardless of the tag density.

4 SNAPLOC: INNER WORKING MECHANISMS

SnapLoc consists of two types of nodes: *anchors* and *tags*. N anchors are placed at known positions $\mathbf{a}^{(i)} \in \mathbb{R}^3$ (with $i = 1, \ldots, N$) to

[1] Furthermore, by overhearing the INIT message and the anchors' responses it is possible to compute the position of other tags or the position of all tags at a central entity, which is valuable for smart factories as well as people- and asset-tracking [21].

localize N_t tags located at an unknown positions $\mathbf{p}^{(n)} \in \mathbb{R}^3$ (with $n = 1, \ldots, N_t$). One of the anchors, A_{REF}, is selected as reference to broadcast the INIT message, as described in Sect. 3.3.

We discuss next how to estimate the unknown positions of the tags $\mathbf{p}^{(n)}$. We first assign an individual delay δ_i to anchors in order to avoid misclassification of responses (Sect. 4.1). We then present a mechanism to reliably detect responses within a CIR (Sect. 4.2), show how to derive the TDOA from the detected responses (Sect. 4.3), and how to use this information to estimate the position of the tags $\mathbf{p}^{(n)}$ (Sect. 4.4). We finally describe SnapLoc's implementation on a low-cost UWB platform and present a clock correction scheme to support constrained anchors (Sect. 4.5).

4.1 Setting Individual Anchor Delays

To avoid the overlap of anchor responses and MPCs in the CIR, we suggest to use an individual delay δ_i at each anchor to separate the responses in time, as discussed in Sect. 3.1. Due to the limited length of the CIR register in common UWB transceivers, there is a trade-off between how much the anchors' responses can be separated in time (i.e., the ability to avoid overlaps between strong MPCs and actual responses), and the supported number of anchors. For example, the DW1000 radio limits the CIR to a maximum length of 1016 samples with a sampling period of $T_s = 1.0016$ ns [9].

In SnapLoc, we set the individual delay $\delta_i = (i-1) \cdot \alpha$, where α represents the size of the slot assigned to each anchor. We suggest to use $\alpha = 128\,ns$, which relates to a distance offset of ≈ 38.5 m and makes it very unlikely that a strong MPC of an earlier response interferes with the current response[2]. This allows to use up to eight anchors when using the DW1000 transceiver [9]. In case this anchor density is insufficient, one needs to reduce α to increase the number of supported anchors. In multipath-rich enviroments, this is not advisable, and we suggest instead to support an unlimited number of anchors using a cellular approach similar to the one employed in mobile networks, e.g., GSM. Instead of multiple frequencies, one can use orthogonal *preamble codes* between neighboring cells, which enables the re-use of slots in the channel impulse response[3].

Note that the use of an individual delay δ_i to separate the responses of a CIR in time is, in spirit, similar to the one proposed by Großwindhager et al. [12]. However, in that solution, slots are assigned to mobile tags: this highly limits the number of users that can be supported and hence the scalability of the system, even when using techniques such as *pulse shaping*. In SnapLoc, instead, we allocate slots to anchors, whose number is limited and known beforehand, which allows to keep the overall design simple.

4.2 Reliable Response Detection

Reliably detecting anchor responses in the CIR is key to achieve a high performance. To this end, in SnapLoc we follow these steps:

(1) Upsample the estimated CIR denoted as $\mathbf{r}$ using fast Fourier transform by a factor of $L = 30$. This improves the time granularity for further processing.

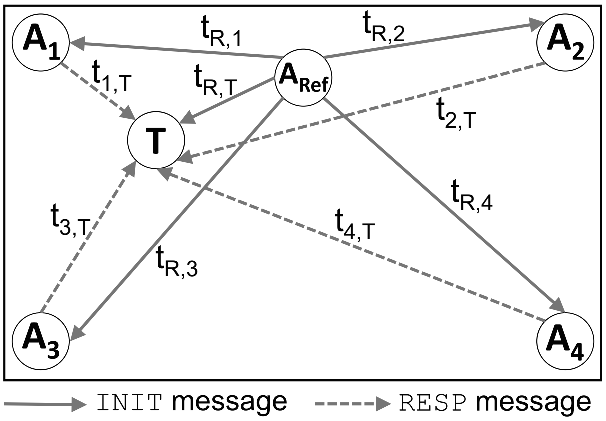


Figure 5: In SnapLoc, a reference anchor A_{REF} broadcasts an INIT message, to which all surrounding anchors reply (quasi-)simultaneously with an empty RESP message.

(2) Use the estimated channel impulse response to compute the matched filter output $\mathbf{y} = \mathbf{h}_{MF} * \mathbf{r}$, where $*$ marks the discrete convolution and $\mathbf{h}_{MF}$ is the time-discrete impulse response of the matched filter. The latter is defined as the time-reversed transmitted pulse shape $s(t)$ [10], which is derived in a measurement campaign according to [14]. This operation optimizes the signal-to-noise ratio of $\mathbf{r}$.

(3) Within each slot i defined by the individual delay δ_i described in Sect. 4.1, the first sample m_i of the matched filter output $\mathbf{y}$ exceeding a certain threshold TH indicates the first path of each anchor response. The threshold TH is chosen experimentally as the 10-fold power of the noise floor.

(4) The estimated time difference of the responses $\Delta\hat{\tau}_{i,j}$ is determined by $\Delta\hat{\tau}_{i,j} = (m_j - m_i) \cdot (T_s/L)$.

4.3 Deriving Time Difference of Arrival

As discussed in Sect. 3.3, SnapLoc employs an *anchor-initiated* approach, where a reference anchor A_{REF} broadcasts an INIT message. The remaining anchors (marked as $A_1 \ldots A_4$ in Fig. 5) respond simultaneously with a RESP message after a delay $\Delta_R + \delta_i$ (with $i = 1 \ldots 4$). A nearby tag T can listen to the signals sent from the anchors and detect the responses in the CIR using the algorithm described in Sect. 4.2. Similarly to the approach discussed in Sect. 3.2, the responses encode information related to the time difference of arrival between the anchors. However, due to the different setup, the distances of the responses $\Delta\tau_{i,j}$ follow as:

$$\Delta\tau_{i,j} = (\delta_j - \delta_i) + (t_{R,j} + t_{j,T}) - (t_{R,i} + t_{i,T}). \tag{3}$$

Due the static nature of the anchors, $t_{R,i}$ and $t_{R,j}$, respectively, are known, and the TDOA $\Delta t_{i,j}$ of the anchors A_i and A_j follows as:

$$\Delta t_{i,j} = t_{j,T} - t_{i,T} = \Delta\tau_{i,j} - (\delta_j - \delta_i) - t_{R,j} + t_{R,i}. \tag{4}$$

Selection of reference anchor. In principle, any anchor within the communication range and optimally in line-of-sight of all other anchors in the same area can be selected as reference (A_{REF}). The selection of an anchor as initiator allows tags to *self-localize*, as discussed in Sect. 3.3. Furthermore, it also increases the robustness of the localization system. Indeed, anchors are typically installed

[2] This holds true also for large areas as the MPCs are attenuated due to path- and reflection loss and will have a negligible impact on the response of the next anchor.

[3] According to the IEEE 802.15.4 standard [39], up to 24 orthogonal preamble codes can be used to extend SnapLoc with this approach. The implementation of such an extension is, however, out of the scope of this paper and left as future work.

in corners and well above objects in a room. Thus, it is less likely that there is a degraded link between anchors. For this reason, the probability to lose the *INIT* message is lower in the *anchor-initiated* approach (Fig. 5) than with the *tag-initiated* one (Fig. 3).

Broadcast anchor positions. As for every anchor-based system allowing self-localization of tags, also in SnapLoc a tag needs to know the ID and location of the anchors to compute its position. To avoid the need of additional infrastructure or packet exchanges, we propagate (i) the ID of the anchors, (ii) their individual delay δ_i, and (iii) their position within the INIT message sent by the reference anchor. Furthermore, the INIT message contains the initialization interval T_{init} between two consecutive INIT messages, as well as transmit timestamp correction values as discussed in Sect. 5.

4.4 Localization Algorithm

As described in Sect. 4.3, the time difference of arrivals between the anchors are derived from the CIR. Based on these estimates, we are able to directly derive the unknown position of the tags $\mathbf{p}^{(n)}$ using TDOA trilateration. For simplification and due to space limitations, we tackle in this section just the two-dimensional case ($\mathbb{R}^2$) and a single tag (i.e., N_t=1) at position $\mathbf{p}^{(1)} = \mathbf{p} = [x, y]^{\mathrm{T}}$. The anchor nodes are positioned at $\mathbf{a}^{(i)} = [x_i, y_i]^{\mathrm{T}}$ (with $i = 1, \ldots, N$). The distance d_i between the tag and an anchor A_i is defined by:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{5}$$

Therefore, the distance differences between anchors Δd_{ij} (with $i \neq j$) – derived by multiplying the time difference of arrivals $\Delta t_{i,j}$ with the propagation speed c – is:

$$\Delta d_{i,j} = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}. \tag{6}$$

The use of N anchors results in $N - 1$ non-redundant nonlinear equations. In the two-dimensional space, at least $N = 4$ anchors are required, i.e., three non-redundant equations, for finding the unambiguous position of a tag [2]. Even with $N = 4$ anchors, just with zero measurement noise we are guaranteed to get a single solution, which corresponds to the real tag position $\mathbf{p}$. Adding white Gaussian measurement noise $\boldsymbol{n}$ results in the signal model in vector notation in Eq. (7), expressing the relationship between the position of anchor/tag and the estimated time difference of arrivals $\Delta\hat{t}_{i,j}$. We obtain the latter by applying equation (4) to the estimated time differences of the responses $\hat{\tau}_{i,j}$ derived from the CIR (see Sect. 4.2). Please note that we relate the TDOA estimates to the first anchor.

$$\hat{\boldsymbol{d}} = \boldsymbol{s}(\mathbf{p}) + \boldsymbol{n} \tag{7}$$

with

$$\boldsymbol{s}(\mathbf{p}) = \begin{bmatrix} \Delta d_{2,1} \\ \vdots \\ \Delta d_{N,1} \end{bmatrix} = \tag{8}$$

$$= \begin{bmatrix} \sqrt{(x_2 - x)^2 + (y_2 - y)^2} - \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ \vdots \\ \sqrt{(x_N - x)^2 + (y_N - y)^2} - \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \end{bmatrix}$$

and the observation vector

$$\hat{\boldsymbol{d}} = c \cdot [\Delta\hat{t}_{2,1}, \Delta\hat{t}_{3,1}, \ldots, \Delta\hat{t}_{N,1}]^{\mathrm{T}}.$$

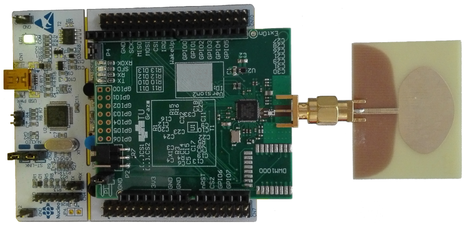


Figure 6: Low-cost UWB platform based on a Decawave DW1000 radio with an omni-directional dipole antenna.

Based on Eq. (7), the nonlinear least squares (NLS) cost function $J_{NLS}(\tilde{\mathbf{p}})$ follows as [38]:

$$J_{NLS}(\mathbf{p}) = (\hat{\boldsymbol{d}} - \boldsymbol{s}(\mathbf{p}))^{\mathrm{T}}(\hat{\boldsymbol{d}} - \boldsymbol{s}(\mathbf{p})). \tag{9}$$

Therefore, the NLS position estimate follows as:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} J_{NLS}(\mathbf{p}) = \arg\min_{\mathbf{p}} (\hat{\boldsymbol{d}} - \boldsymbol{s}(\mathbf{p}))^{\mathrm{T}}(\hat{\boldsymbol{d}} - \boldsymbol{s}(\mathbf{p})). \tag{10}$$

To find $\hat{\mathbf{p}}$, we use the quasi-Newton method [36] with an initial position estimate chosen at the center of the room.

4.5 Implementation

We implement SnapLoc on a low-cost UWB platform described in Sect. 4.5.1. In general, the hardware employed to build a localization system severely affects the minimum response delay ΔR that can be used, as discussed in Sect. 4.5.2. This affects the update rate that can be achieved by SnapLoc, and raises the need for a simple clock correction scheme at the anchor, which we present in Sect. 4.5.3.

4.5.1 Hardware. We employ a self-made low-cost UWB platform based on the IEEE 802.15.4-compliant DW1000 transceiver for both anchors and tags. The platform, shown in Fig. 6, is based on the STM32 Nucleo-64 board, which employs an ultra-low power ARM Cortex-M3 based STM32L152RE microcontroller. The RF front-end contains a low-cost EPSON TSX-3225 oscillator with a frequency of 38.4 MHz, a tolerance of 10 ppm, and with no temperature compensation. To overcome quartz imperfections and varying temperatures we used a built-in functionality of the DW1000 to tune the clock of the anchors depending on the INIT message received from the reference anchor. Every node uses an off-the-shelf omni-directional UWB dipole antenna. The DW1000 is configured to use channel 4 (i.e., a bandwidth of 900 MHz and a carrier frequency of 3.9936 GHz), maximum data rate (6.8 Mbps), a pulse repetition frequency of 64 MHz, as well as a preamble symbol repetition of 128.

4.5.2 Minimum response delay. In SnapLoc, anchors respond to an INIT message broadcasted by the reference anchor after a delay $\Delta_R + \delta_i$. Since the first symbol of the physical header (PHR) determines the transmit timestamp [9], the minimum applicable response delay $\Delta_{R,min}$ is defined by the duration of PHR and payload of the INIT message, as well as the duration of preamble and start-of-frame-delimiter (SFD) of the RESP message. Overall, this corresponds to a delay of 178.5 μs. Additionally, we also need to account for the minimum time necessary to switch the DW1000 radio from receive to transmit mode. We have evaluated this minimum switching time (due to the SPI communication overhead and delays introduced by the processing of a packet) experimentally using two different platforms. The powerful Decawave EVB1000 board, which embeds the STM32F105 ARM Cortex M3 microcontroller, exhibits a minimum switching time of roughly 100 μs, which results in $\Delta_{R,min} = 278.5\,\mu s$. When employing a more constrained microcontroller with lower CPU and SPI speed to control the DW1000 (e.g., the UWB platform presented in Sect. 4.5.1), the minimum response delay increases to $\Delta_{R,min} = 850\,\mu s$. This delay affects the update rate that can be achieved by SnapLoc, as shown in Sect. 6.2.

4.5.3 *Clock correction.* SnapLoc requires that all anchor nodes send their RESP message at well-defined time instances. Therefore, variations of the response delay Δ_R due to imperfections of low-cost oscillators driving the UWB transceiver can potentially degrade the performance of the system. This problem is exacerbated when using a highly constrained hardware causing a large minimum response delay $\Delta_{R,min}$, as discussed in Sect. 4.5.2. Thus, to allow a flexible selection of the response delay Δ_R, we suggest a simple technique to correct the response time $t_{k,i}^{TX,\,R}$ at each anchor A_i.

Consider that the oscillators of the reference anchor and a fixed anchor A_i are running at different speeds due to imperfections, i.e., also the reported time $C_{REF}(t_k)$ and $C_i(t_k)$ vary. The relative skew $a_{REF,\,i}$ between them can be calculated as [17]:

$$a_{REF,\,i} = \frac{C_i(t_{k+1}) - C_i(t_k)}{C_{REF}(t_{k+1}) - C_{REF}(t_k)}, \tag{11}$$

where $C_{REF}(t_k) = t_k^{TX,\,I}$ is the transmission time of the k^{th} INIT message and $C_i(t_k) = t_{k,i}^{RX,\,I}$ denotes the reception time of the k^{th} INIT message at the anchor A_i neglecting the time of flight. In SnapLoc, the reference anchor broadcasts the INIT message with the interval $T_{init} = t_{k+1}^{TX,\,I} - t_k^{TX,\,I}$. Eq. (11) hence follows as:

$$a_{REF,i} = \frac{t_{k+1}^{RX,\,I} - t_k^{RX,\,I}}{T_{init}}. \tag{12}$$

The common response time Δ_R and the individual anchor delay δ_i are defined in the common time of the reference node. Thus, they have to be brought into the time domain of the corresponding anchor A_i using the relative skew $a_{REF,\,i}$ between them. The corrected transmit time $t_{k,i}^{TX,\,R}$ of the RESP message at A_i follows as:

$$t_{k,i}^{TX,\,R} = t_{k,i}^{RX,\,I} + a_{REF,i} \cdot (\Delta_R + \delta_i). \tag{13}$$

5 IMPROVING TIMESTAMP RESOLUTION

To implement SnapLoc on the UWB Decawave DW1000 transceiver, we employ the *delayed transmission* feature. The latter allows to program a future timestamp in a register and lets the DW1000 initiate a packet transmission at this defined timestamp. In SnapLoc, this allows each anchor to set the timestamp at which the RESP message needs to be transmitted upon reception of the INIT message. Although the DW1000 radio represents receive (RX) and transmit (TX) timestamps as 40-bit values with a resolution of 15.65 ps [9], it ignores the lower 9-bit when performing delayed transmissions. This lowers the effective transmission resolution from (theoretical) 15.65 ps to $4/(499.2 \cdot 10^6) \approx 8\,ns$. Without correction, in SnapLoc, this transmission uncertainty results in a uniformly distributed and memoryless error $e_{TS} \sim \mathcal{U}(-8\,ns \cdot c, 0)$. Considering that an error of 1 ns in the time domain results in an error of ≈ 30 cm in the distance domain, it is evident that this error highly affects the localization performance, as we show experimentally in Sect. 6.3. Thus, to sustain a decimeter-level accuracy in SnapLoc, we propose two techniques to increase the transmit timestamp resolution.

Wired correction. We first propose an optimal correction scheme that tracks the lost 9-bit at each anchor and sends these correction values back via a wired backbone to the reference anchor A_{REF}. Such a wired connection is typically available in localization systems, in order to power, reprogram, and reconfigure the anchors. In this scheme, the reference anchor broadcasts the missing transmit timestamp information in the next INIT message to all tags. The latter then correct the timestamps of the anchor responses derived from the previous CIR. In this way, the correction does not require additional messages to be transmitted, as the correction values are embedded in the INIT message. Nonetheless, the tag applies the correction values sent in the latest INIT message to correct the timestamps of the previous position estimate, which causes a delay by one initialization interval T_{init}. Due to the high update rate of SnapLoc, this trade-off is tolerable, as discussed in Sect. 6.2.

Wireless correction. In case a backbone network is not available, we propose a second scheme to increase the timestamp resolution that does not require a wired connection between anchors and the reference anchor. In principle, so far, the latter was used to initiate a position estimation by sending an INIT message and could act as a regular anchor by responding to its own initialization message. In the *wireless correction* scheme, instead, the reference anchor listens to the responses of the anchors and derives the estimated CIR. As the anchors are static and their positions are known, the distance information estimated from the CIR can be compared with the true values. Deviations of the estimations from the true values are treated as errors due to ignoring the least significant 9-bits in the transmit timestamp. To recover the lost precision, we differentiate between the correction at anchor A_1 and the remaining anchors. This is due to the fact that, in SnapLoc, anchor A_1 has an individual time delay $\delta_1 = 0$ and its response hence corresponds to the first peak in the CIR. Thus, the timestamp of its response $t_{RESP,1}^{RX}$ is detected with the highest possible resolution of 15.65 ps by the embedded leading edge detection of the DW1000 [9]. Instead, the resolution of the remaining anchor responses is limited by the sampling period $T_s = 1.0016\,ns$ of the CIR (see Sect. 4.1). For anchor A_1, we define the transmit error due to the limited timestamp resolution e_{A1}^{TX} as the difference between the true round trip time t_{RT} of A_1 and A_{REF} and the estimated one $\hat{t}_{RT}$:

$$e_{A1}^{TX} = t_{RT} - \hat{t}_{RT}. \tag{14}$$

The true round trip time t_{RT} is defined by

$$t_{RT} = 2 \cdot t_{ref,1} + \Delta_R + \delta_1 + 2 \cdot \Theta_a, \tag{15}$$

where $t_{ref,1}$ is the time of flight between reference node and A_1, Δ_R the common response delay at all anchor nodes, δ_1 the individual response delay of A_1, and Θ_a an antenna delay. The latter is required to correct for delays introduced by the antenna, PCB, and internal and external components [9, p.205 ff.]. To measure the antenna delay Θ_a, we have performed 5000 two-way ranging trials between two nodes placed 3 m apart from each other. The antenna delay Θ_a is calibrated such that the difference between the reported distance and the true distance $d_0 = 3\,m$ is minimized. The estimated round trip time $\hat{t}_{RT}$ is determined by the difference between the timestamp $t^{RX}_{RESP,1}$ of A_1's response and the transmission time of the INIT message at the reference anchor t^{TX}_{INIT}. Therefore, the TX timestamp error of A_1 follows as:

$$e^{TX}_{A_1} = (2 \cdot t_{ref,1} + \Delta_R + \delta_1 + 2 \cdot \Theta_a) - (t^{RX}_{RESP,1} - t^{TX}_{INIT}). \quad (16)$$

The transmit timestamp resolution error of the remaining anchors $e^{TX}_{A_i}$ $(i = 2, \ldots, N)$ is defined as the true TDOA $\Delta t_{i,1}$ between A_i and A_1 and the one estimated from the CIR $\Delta\hat{t}_{i,1}$:

$$e^{TX}_{A_i} = \Delta t_{i,1} - \Delta\hat{t}_{i,1}. \quad (17)$$

The true TDOA $\Delta t_{i,1}$ is derived from the known positions of the reference node and anchors and follows as:

$$\Delta t_{i,1} = t_{ref,i} - t_{ref,1} \quad (18)$$

where $t_{ref,i}$ is the time of flight between A_{REF} and A_i. The estimated TDOA $\Delta\hat{t}_{i,1}$ is derived from the CIR according to (2) and has to be corrected by the previously acquired transmit error of A_1 $e^{TX}_{A_1}$. Thus, the resulting error of the anchor A_i is:

$$e^{TX}_{A_i} = 2 \cdot (t_{ref,i} \cdot - t_{ref,1}) - (\Delta\hat{t}_{i,1} + e^{TX}_{A_1}). \quad (19)$$

As discussed, the resolution of the error value $e^{TX}_{A_i}$ is restricted by the sampling period of the CIR $T_s = 1.0016\,ns$. Thus, 3-bits in the INIT message broadcasted by the reference anchor are enough to represent the error correction value. Therefore, the overhead due to a longer packet size is slightly shorter in the *wireless correction* method compared to the *wired correction*.

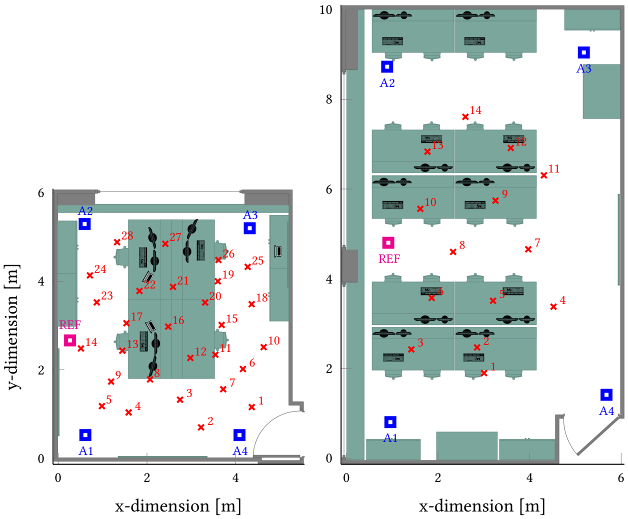
6 EVALUATION

We evaluate SnapLoc experimentally in a challenging office environment (Room A, see Fig. 7a) and a larger laboratory classroom (Room B, see Fig. 7b). We describe the experimental setup in Sect. 6.1, followed by an analysis of the energy consumption in terms of over-the-air time and the potential update rate in Sect. 6.2. We then extensively evaluate the performance of SnapLoc in Sect. 6.3, showing that it can achieve decimeter-level localization accuracy.

6.1 Experimental Setup

To evaluate SnapLoc in a realistic indoor environment, we use a common office for three employees with a size of $5.2 \times 6.03\,m \approx 31.36\,m^2$ (see Fig. 7a) and a larger laboratory classroom with $6.05 \times 10\,m = 60.5\,m^2$ (see Fig. 7b). The rooms contain several scattering and reflecting objects such as monitors, desks, and chairs. The reference anchor (magenta square) and the remaining anchors (blue squares) are placed on tripods at known positions. The height of all tripods is 1.60 m, which puts all nodes in the same 2D plane. For all evaluations, we employ just the *minimum amount of anchors necessary*, i.e., $N = 4$: this allows to examine the performance of SnapLoc using just minimal infrastructure. The number of evaluation points ($N_{EP} = 28$ in Room A and $N_{EP} = 14$ in Room B) are randomly distributed in the rooms to evaluate the performance of SnapLoc. At each evaluation point, $N_P = 500$ position estimates are derived. The absolute error of each trial is calculated as the Euclidean distance between the position of the evaluation point $\mathbf{p}_{EP}$ and the i-th position estimate $\hat{\mathbf{p}}_i$:

$$\text{Err}_i = \|\hat{\mathbf{p}}_i - \mathbf{p}_{\text{EP}}\|. \quad (20)$$



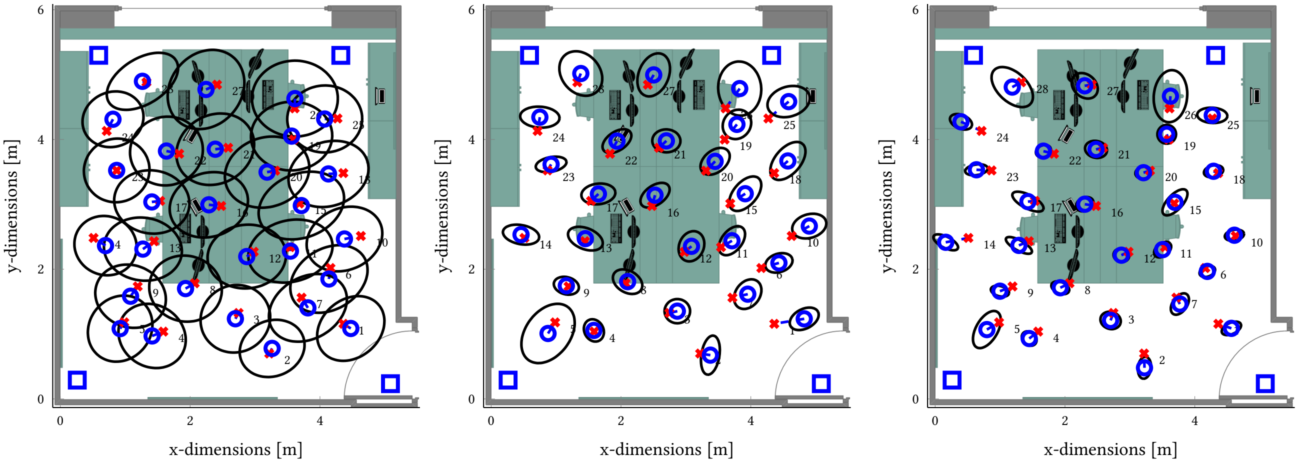
(a) Room A

(b) Room B

Figure 7: Evaluation setup: we consider two different environments with 28 and 14 evaluation points, respectively.

6.2 Position Update Rate and Efficiency

Due to the high current consumption of the DW1000 in the transmit and especially in the receive mode [13], it is critical for UWB-based localization systems to minimize the radio-on time at the tag. Due to the simultaneous acquisition of all the anchor signals, SnapLoc excels in this regard. Indeed, the tag does not have to send any packet, but just listens to a single message. Thus, the number of anchors does not affect the system's energy consumption in terms of packet reception and transmission. This is in contrast to state-of-the-art UWB-based localization systems, where the energy consumption increases – typically linearly – with the number of anchors [24, 28]. We measure the energy consumption of SnapLoc with the settings described in Sect. 4.5.1 with a Keysight MSOS-254A oscilloscope. Acquiring the simultaneous anchor responses requires only approximately $82.4\,\mu J$. Besides a low energy consumption, simultaneously responding anchors also highly affect the achievable position update rate, as the latter relates to the total time needed to provide the tag with the necessary information to estimate its position. In SnapLoc, this total time consists of the duration of INIT and RESP messages, as well as the time to switch between receive and transmit mode at the anchors. As discussed in Sect. 4.5.2, this switching time is approximately $100\,\mu s$ when using the Decawave EVB1000 board and the duration of the two messages is roughly $334\,\mu s$. Thus, deriving the information to estimate the tag's position just takes $434\,\mu s$ overall. Theoretically, this enables an update rate of more than 2.3 kHz for SnapLoc, without any limitation on the number of tags. Even when using the highly constrained microcontroller with low SPI and CPU speeds described in Sect. 4.5.1, we still achieve an update rate of about 996 Hz. This high update rate makes SnapLoc highly suitable for feedback control systems and enables the precise tracking of highly-dynamic objects. Note that the update rate is also influenced by (i) streaming the CIR via SPI from the DW1000, (ii) deriving the actual TDOAs, as well as (iii) executing the algorithm to estimate the tag's position. However, these values are strongly



(a) Without correction **(b) With wireless correction** **(c) With wired correction**

Figure 8: Error ellipses showing the bias (blue circles) and the standard deviation (black ellipses) of the position estimation without correction of the transmit timestamp (a), with the wireless correction (b), and with the wired correction (c).

hardware-dependent and could significantly be reduced by integrating a UWB transceiver together with a performant microcontroller in a system on chip solution. Furthermore, when using the techniques proposed in Sect. 5 to increase the timestamp resolution of the DW1000, the uncertainties of the timestamps have to be either sent back via wire (*wired correction*) or estimated at the reference anchor (*wireless correction*), which decreases the update rate.

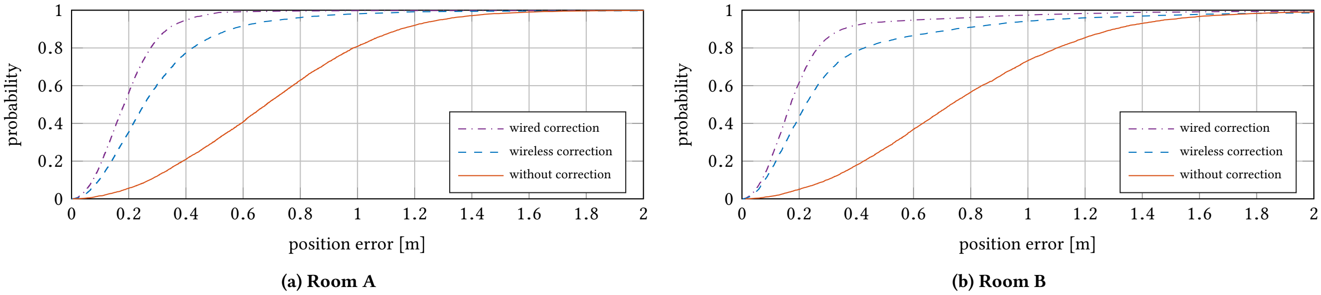
6.3 Localization Accuracy

We evaluate next the performance of SnapLoc and the effectiveness of the methods to overcome the limited transmit timestamp resolution proposed in Sect. 5.

Individual evaluation points. We start by investigating the localization accuracy of SnapLoc in a smaller room (Fig. 7a) using $N_{EP} = 28$ evaluation points. Fig. 8 shows the impact of the transmit timestamp correction techniques presented in Sect. 5. The mean (blue circle) and the standard deviation (black ellipses) for $N_P = 500$ position estimates are shown for each evaluation point. Fig 8a shows the accuracy of SnapLoc's position estimation *without transmit timestamp correction*. Fig 8b shows the accuracy of SnapLoc's position estimation with the *wireless correction*, whilst Fig. 8c with the *wired correction*. As expected, the latter performs best, as it recovers the least significant 9-bits of the transmit timestamp at all anchors. The *wireless correction*, instead, restores a time resolution of 15.56 ps for anchor A_1 and a resolution of 1 ns for the remaining anchors; thus, its performance is slightly worse compared to the one obtained with the *wired correction*. Without any correction, each of the anchor timestamps has a resolution of just 8 ns, which induces a high error, as shown by the larger ellipses in Fig. 8a. Moreover, it is noticeable in Fig. 8b and 8c that the evaluation points within a distance of 1.5 m to an anchor ($EP \in \{1, 5, 24, 25, 26, 28\}$) perform worse than those located further away from the anchors. This is due to the high signal strength of the close anchor, which causes the CIR register to saturate. As the amplitude of the other anchors' responses remains relatively low, a correct response detection is impaired. Thus, when deploying SnapLoc, a distance of at least 1.5 m between the tag and the anchors should be ensured. This is often already the case in indoor localization systems, as anchors are typically mounted close to the ceiling.

Overall localization accuracy and precision. To investigate the overall performance of SnapLoc, we derive its accuracy and precision statistically using the cumulative distribution function (CDF) over the error Err_i of all position estimates. Due to the saturation effects at tag positions close to the anchors, we have ignored the corresponding evaluation points $EP \in \{1, 5, 24, 25, 26, 28\}$ for this analysis. Fig. 9a shows the performance of SnapLoc depending on the used method to correct the limited TX timestamp resolution of the Decawave DW1000. *Without correction* (solid orange line), a 90% error of 1.15 m and a median error of 0.68 m was achieved. Instead, the use of *wireless correction* allows to reduce the 90% error to 55.8 cm and the median error to 25.4 cm (dashed blue line) and the *wired correction* even reaches a 90% error of just 33.7 cm and a median error of 18.4 cm (magenta dash dotted line). Thus, by using the proposed correction methods, SnapLoc easily achieves decimeter-level accuracy despite the limited transmit timestamp resolution of 8 ns and the CIR resolution of about 1 ns.

Performance in larger room. To validate the accuracy of SnapLoc also in other environments, we carry out an evaluation in a laboratory classroom (Fig. 7b) that is significantly larger than the previously employed office room (31.36 m^2 vs. 60.5 m^2). Fig. 9b shows the CDF of all position estimates in the $N_{EP} = 14$ evaluation points shown in Fig. 7b. Without using a transmit timestamp correction, the 90% error is at 1.30 m and the median error at 0.73 m. The *wireless correction* allows SnapLoc to sustain a 90% error of 74 cm and a median error of 22.3 cm. With the *wired correction*, the median error is reduced to 17 cm and the 90% error to 35.2 cm. The slight differences compared to the evaluation in room A are due to the presence of a few more outliers with a position error above 0.5 m, as shown in Fig. 9b. Still, the results are consistent to the evaluation in Room A despite the use of a larger area.



(a) Room A

(b) Room B

Figure 9: Performance of SnapLoc depending on the method used to correct the limited transmit resolution of the DW1000 transceiver in the two rooms used in our evaluation.

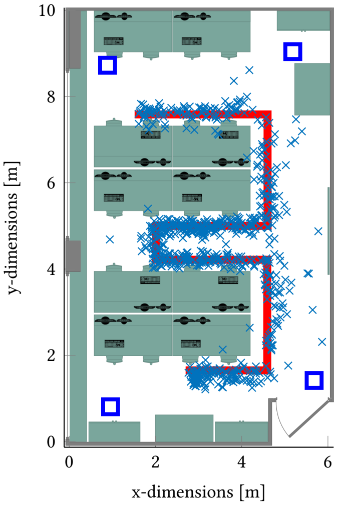


Figure 10: Performance of SnapLoc when a subject follows a pre-defined track (red solid line). The positions estimated by SnapLoc are marked with light blue crosses.

Free movement. The previous evaluations were performed at randomly chosen, but static evaluation points to deliver reproducible results. To investigate the performance of SnapLoc also while moving around freely, we mounted a tag on a rolling stand and asked a subject to follow a marked line in a slow but continuous fashion. While moving, we have continuously estimated the tag's position using SnapLoc combined with the *wired correction* method. Fig. 10 shows the results of the experiment. It is evident that the position estimates (light blue crosses) follow the pre-defined track (red solid line). Especially in the middle of the room, SnapLoc shows reliable results due to equally strong responses of the anchors. Instead, close to the anchors and at the border of the envelope curve spanned by the anchors, the number of outliers increases. This is coherent with the observations made in the previous evaluations. Please note that we did *not* use any (tracking) filter on the measurements, such as Kalman filter, particle filter, or moving average. The results are solely *raw position estimates*. Due to unavailability of a optical tracking system to provide ground truth data, we could not determine the absolute error properly. Still, estimating the shortest distance to the desired track reveals a mean deviation of just 14.8 cm.

7 RELATED WORK

UWB localization systems. Ultra-wideband technology enables decimeter-level localization accuracy in multipath-rich indoor environments without the need of extensive infrastructure [30, 47]. Several practical implementations using low-cost UWB radios exist, e.g., based on the Decawave DW1000 [3, 14, 24, 28, 32, 37, 41], on Time Domain's PulsOn module [11, 45], or on self-made hardware [23]. The main objective of these systems is to achieve a *high localization accuracy*: as a result, the update rate at which the position can be computed has often not been discussed. Amongst works explicitly mentioning the supported update rates, Kempke et al. [24] achieve a 99% error in 3D of 53 cm with an update rate of 12 Hz. However, the latter is divided by the number of supported tags (e.g., 6 Hz for two tags). Silva et al. [37] report average errors between 5 and 40 cm in 2D, and achieve an update rate of 10 Hz for a single tag. Hartmann et al. [16] report an average error of 27 cm in 2D and update a single tag every 50 Hz. SnapLoc achieves similar accuracies (90% error of 33 cm), but *at much higher update rates* and with the ability to support an *unlimited number of tags*.

Passive self-localization. One of the main features of SnapLoc is that it gives tags the ability to carry out passive self-localization and remain anonymous (i.e., tags are not actively transmitting data). This allows to build localization systems that scale regardless of the density of tags, given that an unlimited number of tags can, in principle, localize at the same time. Passive self-localization is the same principle adopted by GNSS systems [19] and one of the key reasons for their long-lasting and enduring success. However, GNSS satellites require the use of atomic clocks to maintain synchronization of anchors. An UWB-based system comparable to GNSS has been presented in [28], but it requires a tight synchronization at the anchors and clock skew correction at the tag due to the use of sequential messages, which is often hard to attain [43, 44]. SnapLoc, instead, *removes the need for tight synchronization* and *does not need a correction at the tag* due to the use of (quasi-)simultaneous responses, which ultimately enables very high update rates.

Concurrent passive localization. Similarly to SnapLoc, also Chorus [7] exploits the concept of concurrent transmissions to perform passive localization. Both works are developed independently and published simultaneously in the same venue: differences include implementation details, evaluation methodology, and the (complementary) slant of the contribution. While Chorus focuses on modeling the impact of the limited timestamp resolution, SnapLoc proposes a technique to overcome this limitation and implements it on platforms making use of state-of-the-art UWB transceivers (Sect. 5): this allows to achieve decimeter-level accuracy, as demonstrated experimentally (Sect. 6) and showcased at public events [40]. SnapLoc also counteracts the clock drift between INIT and RESP messages, enabling also highly-constrained devices (such as the low-cost UWB platform presented in Sect. 4.5.1) to make use of the proposed TDOA-based localization concept.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we present SnapLoc, an ultra-fast localization system for an unlimited number of tags – actually faster than a finger snap, which typically takes 1 to 3 ms [46]. SnapLoc derives simultaneously all the information required to estimate a tag's position, which is enabled by extracting concurrent anchor responses from a single estimated CIR. Based on the detected responses, SnapLoc estimates the TDOA between anchors, removing the need to derive the distance to the closest anchor using multiple messages as in previous solutions, and allowing tags to anonymously self-localize. Furthermore, in contrast to classical TDOA systems, SnapLoc does neither require tight synchronization of anchors, nor correction of clock deviations at the tag. We implement and evaluate SnapLoc experimentally on a low-cost platform based on the Decawave DW1000 UWB radio, as well as mitigate the intrinsic limited transmit timestamp resolution of this transceiver. Our results show that SnapLoc sustains decimeter-level positioning accuracy, with a 90% error of 33.4 cm and a median error of 18.4 cm, and that it is highly suited for supporting mobile applications.

Future work includes the evaluation of SnapLoc in three dimensions, as well as the installation of SnapLoc in a multi-room multi-level building to investigate its performance when introducing multiple cells operating with orthogonal preamble codes.

9 ACKNOWLEDGMENTS

This work was supported by the TU Graz LEAD project ”Dependable Internet of Things in Adverse Environments”.

REFERENCES

[1] A. Alarifi et al. 2016. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors* 16, 5 (2016).

[2] R. M. Buehrer and S. Venkatesh. 2012. Fund. of Time-of-Arrival-Based Position Locations. *Handbook of Position Location: Theory, Practice, and Adv.* (2012).

[3] W. Chantaweesomboon et al. 2016. On Performance Study of UWB Real Time Locating System. In *Proceedings of the 7th IC-ICTES Conference.*

[4] Y.K. Cho et al. 2010. Error Modeling for an Untethered Ultra-Wideband System for Construction Indoor Asset Tracking. *Automation in Construction* 19, 1 (2010).

[5] G. Conte et al. 2014. BlueSentinel: a First Approach using iBeacon for an Energy Efficient Occupancy Detection System. In *Proc. of the 1st ACM BuildSys Conf.*

[6] P. Corbalán et al. 2018. Concurrent Ranging in UWB Radios: Experimental Evidence, Challenges, and Opportunities. In *Proc. of the 15th EWSN Conference.*

[7] P. Corbalán et al. 2019. Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets. In *Proc. of the 18th IPSN '19 Conference.*

[8] W. Dargie and C. Poellabauer. 2010. *Fundamentals of Wireless Sensor Networks: Theory and Practice.*

[9] Decawave Ltd. 2017. *DW1000 User Manual. Version 2.13.*

[10] Chiara Falsi et al. 2006. Time of Arrival Estimation for UWB Localizers in Realistic Environments. *EURASIP Journal on Advances in Signal Processing* 1 (2006).

[11] Javier González et al. 2009. Mobile Robot Localization based on Ultra-Wide-Band Ranging: A Particle Filter Approach. *Robotics and Auton. Syst.* 57, 5 (2009).

[12] B. Großwindhager et al. 2018. Concurrent Ranging with UWB Radios: From Experimental Evidence to a Practical Solution. In *Proceedings of the 38th International Conference on Distributed Computing Systems (ICDCS).*

[13] B. Großwindhager et al. 2018. Enabling Runtime Adaptation of PHY Settings for Dependable UWB Communications. In *Proceedings of the 19th Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM).*

[14] B. Großwindhager et al. 2018. SALMA: UWB-based Single-Anchor Localization System using Multipath Assistance. In *Proceedings of the 16th ACM International Conference on Embedded Networked Sensor Systems (SenSys).*

[15] K. Guo et al. 2016. Ultra-Wideband-Based Localization for Quadcopter Navigation. *Unmanned Systems Journal* 4, 1 (2016).

[16] F. Hartmann et al. 2015. Design of an Embedded UWB Hardware Platform for Navigation in GPS Denied Environments. In *Proceedings of the IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT).*

[17] J. He et al. 2014. Time Synchronization in WSNs: A Maximum-Value-Based Consensus Approach. *IEEE Trans. Automat. Control* 59, 3 (2014).

[18] Suining He and S-H Gary Chan. 2016. Wi-Fi Fingerprint-based Indoor Positioning: Recent Advances and Comparisons. *IEEE Comm. Surveys & Tutorials* 18, 1 (2016).

[19] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. 2007. *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more.*

[20] S. Huang et al. 2017. A Real-time Location System Based on RFID and UWB for Digital Manufacturing Workshop. *Procedia CIRP* 63, 1 (2017).

[21] L. Jiang et al. 2010. Integrated UWB and GPS Location Sensing System in Hospital Environment. In *Proceedings of the 5th ICIEA Conference.*

[22] B. Kempke et al. 2015. PolyPoint: Guiding Indoor Quadrotors with Ultra-Wideband Localization. In *Proc. of the 2nd HotWireless Workshop.*

[23] B. Kempke et al. 2016. Harmonium: Asymmetric, Bandstitched UWB for Fast, Accurate, and Robust Indoor Localization. In *Proceedings of the 15th International Conference of Information Processing in Sensor Networks (IPSN).* IEEE.

[24] B. Kempke et al. 2016. SurePoint: Exploiting Ultra Wideband Flooding and Diversity to Provide Robust, Scalable, High-Fidelity Indoor Localization. In *Proceedings of the* 14^{th} *ACM Int. Conf. on Embedded Network Sensor Systems (SenSys).*

[25] Hakyong Kim. 2009. Performance Comparison of Asynchronous Ranging Algorithms. In *Proceedings of the Global Telecommunications Conference (GLOBECOM).*

[26] M. J. Kuhn et al. 2011. A Multi-Tag Access Scheme for Indoor UWB Localization Systems Used in Medical Environments. In *Proc. of the International Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireless).*

[27] J. Kulmer et al. 2017. Using Decawave UWB Transceivers for High-accuracy Multipath-assisted Indoor Positioning. In *Proc. of the* 5^{th} *IEEE ANLN Workshop.*

[28] A. Ledergerber et al. 2015. A Robot Self-Localization System using One-Way Ultra-Wideband Communication. In *Proceedings of the IROS Conference.*

[29] R. Leser et al. 2014. Accuracy of an UWB-based Position Tracking System Used for Time-Motion Analyses in Game Sports. *Eur. Journ. of Sport Sc.* 14, 7 (2014).

[30] D. Lymberopoulos et al. 2017. The Microsoft Indoor Localization Competition: Experiences and Lessons Learned. *IEEE Signal Processing Magazine* 34, 5 (2017).

[31] S. Maranò et al. 2010. NLOS Identification and Mitigation for Localization Based on UWB Experimental Data. *IEEE J-SAC* 28, 7 (2010).

[32] Y. Qin et al. 2015. A Distributed UWB-based Localization System in Underground Mines. *Journal of Networks* 10, 3 (2015).

[33] M. Ridolfi et al. 2018. Analysis of the Scalability of UWB Indoor Localization Solutions for High User Densities. *Sensors* 18, 6 (2018).

[34] N. C. Rowe et al. 2013. A UWB Transmit-Only Based Scheme for Multi-Tag Support in a Millimeter Accuracy Localization System. In *Proceedings of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet).*

[35] J. L. Rullan-Lara et al. 2013. Indoor Localization of a Quadrotor Based on WSN: A Real-Time Application. *Int. Journal of Adv. Robotic Systems* 10, 1 (2013).

[36] David F Shanno. 1970. Conditioning of Quasi-Newton Methods for Function Minimization. *Math. Comp.* 24, 111 (1970).

[37] B. Silva et al. 2014. Experimental Study of UWB-based High Precision Localization for Industrial Applications. In *Proc. of the Int. Conf. on Ultra-WideBand (ICUWB).*

[38] Hing Cheung So. 2011. Source Localization: Algorithms and Analysis. *Handbook of Position Location: Theory, Practice, and Advances* (2011).

[39] IEEE Computer Society. 2015. Standard for Low-Rate Wireless Networks.

[40] M. Stocker et al. 2019. Demo Abstract: SnapLoc: An Ultra-Fast UWB-Based Indoor Localization System for an Unlimited Number of Tags. In *Proceedings of the 18th ACM/IEEE IPSN Conference.*

[41] J. Tiemann et al. 2016. Atlas: An Open-Source TDOA-based Ultra-Wideband Localization System. In *Proceedings of the IPIN Conference.*

[42] K. Witrisal et al. 2016. High-Accuracy Local. for Assisted Living: 5G systems will Turn Multipath Channels from Foe to Friend. *IEEE Signal Proc. Mag.* 33, 2 (2016).

[43] Y. Wu et al. 2011. Clock Synchronization of Wireless Sensor Networks. *IEEE Signal Processing Magazine* 28, 1 (2011).

[44] B. Xu et al. 2013. High-Accuracy TDOA-Based Localization Without Time Synchronization. *IEEE Transactions of Parallel Distributed Systems* 24, 8 (2013).

[45] Wang Yan et al. 2015. The Designing of Indoor Local. System Based on Self-Organized WSN Using PulsON UWB Sensors. In *Proc. of the 2nd ICISCE Conf.*

[46] Yanni Yang et al. 2016. Person Authentication Using Finger Snapping — A New Biometric Trait. *Biometric Recognition* (2016).

[47] Jinyun Zhang et al. 2009. UWB Systems for Wireless Sensor Networks. *Proc. of the IEEE* 97, 2 (2009).