TALLA: Large-scale TDoA Localization with Ultra-wideband Radios

Davide Vecchia, Pablo Corbalán, Timofei Istomin, Gian Pietro Picco
University of Trento, Italy
{davide.vecchia, p.corbalanpelegrin, timofei.istomin, gianpietro.picco}@unitn.it

***Abstract*—Time difference of arrival (TDoA) localization with ultra-wideband (UWB) radios is rapidly gaining interest. Nevertheless, TDoA requires tightly time-synchronized anchors, either via wireless synchronization in small-scale setups with all anchors in range or via expensive wired backbones, both ultimately hampering the use of TDoA in large-scale areas.**

In this paper we present TALLA, a novel wireless-only TDoA approach able to scale over large operational areas without sacrificing positioning accuracy. TALLA relies on a TDMA schedule enabling continuous, multi-hop operation of the anchor infrastructure. We evaluate TALLA in our 12-node UWB testbed and in much larger (>100 anchors) simulated areas, empowered by a technique that generates synthetic timing information faithfully reproducing the trends of the real one. Our real and simulated results show that TALLA achieves decimeter-level accuracy while tracking a moving target across several hops.

I. Introduction

Ultra-wideband (UWB) radios are rapidly gaining popularity among localization technologies. A new generation of UWB transceivers, spearheaded by the DecaWave DW1000 [1], has redefined the tradeoffs associated to this technology with radio chips that are tiny, cheap, low-power, yet capable of accurate (<10 cm error) distance estimation.

Industry and academia seized this new opportunity, with initial efforts focused on the two-way ranging (TWR) scheme relying on the time of arrival (TOA) estimation. However, in TWR-based schemes [2], at least 2 messages are required to estimate distance between a tag and an anchor; this must be repeated for each anchor and requires continuous neighbor discovery. Therefore, commercial and research systems increasingly rely on *time difference of arrival* (TDoA) estimation (§II), where *single* message transmission from the tag suffices to compute the difference in reception time at *all* localization anchors. Therefore, TDoA is more scalable than TWR, in terms of mobile tags and/or sample update rate supported [3].

Goal: Large-scale, Flexible TDoA Operation. However, TDoA requires tight time synchronization of the anchors. This can be achieved with an out-of-band wired network, often already present but otherwise expensive to deploy. Alternately, in-band UWB wireless schemes exist [4], [5] that greatly increase deployment flexibility and reduce cost. However, these have been applied only in small-scale settings (e.g., a room) with *all* anchors in range, where clock drift can be mitigated via one-hop wireless synchronization.

978-1-7281-1788-1/19/$31.00 2019 IEEE

In contrast, the TALLA (TDoA Localization for Large-scale Areas) system we present in this paper *i)* performs time synchronization in-band, using the wireless UWB link, and *ii)* is capable to *scale over large operational areas without sacrificing positioning accuracy*.

Contribution. Our design (§III) is based on TDMA, with time slots allocated for anchor and mobile tag transmissions. Each anchor is the potential (sub)ns-level time reference for TDoA localization; this feature enables a *continuous, multi-hop* operation of the wireless anchor infrastructure supporting localization. Further, reliance on TDMA prevents collisions and therefore improves reliability and update rate.

Anchors are allocated n slots in the TDMA frame, enabling their synchronization. Tags are allocated the remaining k slots for localization. These values are configured based on application and system requirements, e.g., catering for different update rates, anchor density and total number of tags. Distant anchors can safely share time slots, therefore n depends only on anchor density but is constant w.r.t. network size.

The evaluation of TALLA is non-trivial; the verification of our claims about large-scale operation directly depends on the number of nodes and the size and geometry of the operational area they are deployed in. We evaluate our prototype in a 12-node UWB testbed in a 100×60 m^2 corridor area at our premises. This is larger than what commonly reported [6]–[8]; yet, the number of nodes is relatively small, the network diameter of is only 3 hops, and the geometry very challenging. For this reason, we adopt an evaluation methodology (§IV) that exploits our prototype also to inform, via real packet traces, a simulation toolchain that faithfully reproduces the timing inaccuracies affecting TDoA positioning error (§V). This enables us to derive synthetic yet realistic traces for areas whose size is well beyond the one of our testbed, which we nonetheless use to validate our simulated results (§VI). Further, this mixed simulation-testbed strategy also allows us to easily explore the parameter space, which is key to analyze the design and configuration choices germane to our approach.

The paper ends by placing our work in the context of related ones (§VII) before offering concluding remarks (§VIII).

II. TDoA Localization: Fundamentals

We describe the necessary background on TDoA localization with UWB radios, with wireless time synchronization.

Infrastructure. We assume a network infrastructure with at least $N = 4$ anchors in range, enabling 2D positioning. Each anchor reports each transmission (TX) and reception (RX) along with their corresponding timestamps to a server, where the actual localization computation takes place, through a backbone network (e.g., WiFi). Mobile tags roaming in the covered area can be located with a single packet TX per tag.

Clock Synchronization. To compare the RX timestamps for TDoA localization, the server must have a clock model per anchor to translate the *local* anchor timestamps to the same clock domain. To this end, a reference anchor periodically transmits a synchronization beacon every $T_s = 1/f_s$. This beacon is received by anchors in range, which report their RX timestamps to the server that, in turn, computes the k^{th} clock offset of each anchor i w.r.t. the reference as

$$\theta_{i,k} = (t_{i,k} - \tau_{i,ref}) - t_{ref,k} \qquad (1)$$

where $\tau_{i,ref}$ is the known time of flight between anchor i and the reference anchor, determined based on the known positions $\mathbf{p_i}$ and $\mathbf{p}_{ref}$ and the speed of light in air c. The clock drift of anchor i w.r.t. the reference anchor can then be estimated based on the previous $k-1$ beacon TX/RX information as

$$\delta_{i,k} = \frac{\theta_{i,k} - \theta_{i,k-1}}{T_s} \qquad (2)$$

Based on $\theta_{i,k}$ and $\delta_{i,k}$, the server can translate each local anchor timestamp t_i to the reference clock domain as $t_i - \theta_{i,k} + t_e(1-\delta_{i,k})$, where t_e is the measured time elapsed since the last synchronization beacon RX.

Position Estimation. The server can estimate the true position $\mathbf{p}$ of a mobile tag in 2D based on the RX timestamps of at least $N = 4$ anchors, including the reference. The server first translates the local timestamps to the reference clock domain and then computes the TDoA estimates $\hat{\Delta} t_{i,ref} = t_i - t_{ref}$, each representing the equation of a hyperbola

$$\Delta t_{i,ref} = \frac{\|\mathbf{p} - \mathbf{p_i}\| - \|\mathbf{p} - \mathbf{p}_{ref}\|}{c} \qquad (3)$$

To estimate the tag position $\hat{\mathbf{p}}$ with $N-1 \geq 3$ non-redundant TDoA estimates, the server solves a non-linear least squares problem by minimizing the squared difference between the measured TDoA estimates $\hat{\Delta} t$ and the theoretical ones Δt as

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i}^{N-1} \left(\hat{\Delta} t_{i,ref} - \Delta t_{i,ref} \right)^2 \qquad (4)$$

III. Enabling TDoA over Large Areas

The wireless synchronization mechanism described requires all anchors to be in range of the reference, limiting scalability. We present our design enable TDoA localization across large areas requiring multi-hop communication. The design is based on a time-slotted approach that follows a periodic schedule. This schedule allows each node in the network (anchor or tag) to transmit its packets without collisions based on TDMA and, at the same time, enables (sub)ns-level clock synchronization for TDoA localization at the server side, where each anchor can serve as a potential time reference. The schedule is also configurable to cater for various requirements in terms of localization rate as well as number of tags and anchors.

Time-slotted Schedule Design. The schedule unfolds as a repetition of a pre-defined sequence of time slots, each of duration T ms, forming an *epoch* of duration D. The value of D is tied to the synchronization rate required to reliably compensate for clock drift. Within each epoch, the schedule provides each anchor and tag a dedicated time slot to transmit a beacon. Anchor transmissions serve to time-synchronize the network and enforce the schedule, while tag beacons provide the necessary TDoA measurements for positioning.

Figure 1 shows an example. Each of the n anchor slots is assigned to one anchor beacon, without repetitions. These beacons enable the network to build a time synchronization tree, described next, to avoid collisions among tags and/or anchors. They also enable the server to collect the information required to establish any anchor as a precise TDoA time reference. Tags listen periodically to anchor beacons to retain alignment with the schedule. Anchors are always listening except when they transmit their own synchronization beacons.

The remaining k slots are instead used by tags to transmit their localization beacon. Unlike anchor slots, the same tag can be assigned multiple slots. Tag beacons are received by anchors, which report the RX timestamps to the server, which in turn computes the tag position using the TDoA solver (§II).

Schedule Definition. The schedule is defined at compilation time, based on the expected localization accuracy and update rate. However, it can be tailored to application needs, e.g., by assigning multiple slots to a tag requiring a higher update rate. Similarly, the n anchor slots need not be consecutive and can be spread throughout the epoch. In any case, their number n detracts from the number k of those available for tags; this can be problematic in large areas with many anchors.

Nevertheless, the problem can be easily solved by enabling non-connected sets of anchors to re-use slots and transmit their beacons concurrently and safely, without possibility of collision. This requires a network connectivity assessment prior to operation, which can be easily performed by scheduling $n = N$ beacons (i.e., one slot per anchor). This enables each anchor to determine its neighbors, which can be reported to the server. The latter can use well-known graph coloring algorithms and communicate back to each anchor the actual full schedule to use, therefore reducing n while retaining a reliable and robust synchronization infrastructure.

Network-wide Synchronization. To enforce the schedule and avoid collisions, the network must be time-synchronized. Anchor beacons are used to build a synchronization tree where a pre-defined node (e.g., anchor 1) serves as the global time reference. Note that to follow the schedule, µs-level synchronization is sufficient. To build the synchronization tree

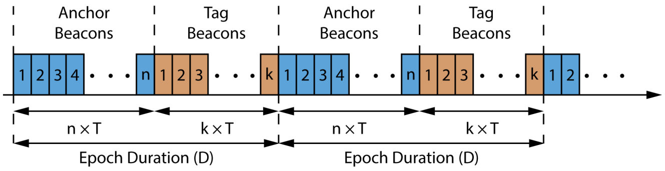


Figure 1. Example time-slotted schedule including n dedicated anchor slots for time synchronization and k slots for tag localization beacons.

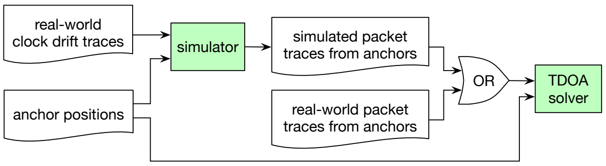

real-world clock drift traces
simulator
simulated packet traces from anchors
anchor positions
OR
TDOA solver
real-world packet traces from anchors

Figure 2. Prototypes and evaluation toolchain.

and avoid loops, nodes use a routing metric (e.g., hop count) to propagate the clock offsets, allowing nodes that are multiple hops away to follow the schedule. For anchors to be able to learn their time offset w.r.t. the reference, beacons must carry the node ID and metric. Based on the beacons received, nodes select an appropriate time source (e.g., the parent with the lowest hop count). Using the RX timestamp and the node ID of the parent, a node can learn its time offset and transmit its beacon in the corresponding time slot. As the schedule repeats itself, nodes can re-synchronize every epoch, reducing drastically the impact of clock drift on the schedule.

High-Precision Synchronization for TDoA Localization. Anchor beacons are also used to achieve (sub)ns-level time synchronization at the server, for TDoA localization. This requires each beacon TX/RX from an anchor to be precisely timestamped and sent to the server, which maintains a data structure containing the TX timestamp and sequence number of every beacon along with the ID of anchors that received it and their RX timestamps. Based on this information, the server can pick any anchor as a potential ns-level time reference to estimate the TDoA measurements required for positioning. With typical clock drifts, the synchronization frequency required to obtain accurate TDoA measurements must be $f_s \geq 1$ Hz. This in turn constrains the epoch duration, which should be kept short enough ($D \leq 1$ s) to reduce the positioning error.

Anchor Selection and Position Estimation. The solver accuracy is closely related to the choice of anchors involved. A first issue is selecting the appropriate TDoA reference among the many available in the large-scale areas targeted by TALLA. A tag beacon is received by many anchors, possibly synchronized with different candidate references; however, the anchors used for localization must be synchronized together. Our solver selects as reference the one that synchronized the highest number of anchors (among those that received the tag beacon) within the last epoch; in case of a tie, the anchor synchronized more recently is chosen. A second issue is validating the solver output. If the tag position is estimated at a distance from anchors greater than communication range, it is invalid. Localization is repeated with a subset of anchors, potentially exploring all combinations until a valid result is obtained.

IV. Evaluation Methodology

The specific focus of TALLA demands that its performance is evaluated on *large-scale areas* containing many anchors, e.g., tens if not hundreds. Unfortunately, publicly-accessible UWB testbeds of this size are not available, and ad-hoc setups are prohibitively effort-demanding. Similarly, simulators with realistic performance are unreported in the literature.

Hence, we adopt an evaluation methodology that combines a mid-scale testbed with a simulation approach yielding realistic and accurate estimates. Both are supported by a single evaluation toolchain (Figure 2) relying on two key software artifacts: a *simulator* of timing inaccuracies and a *TDoA solver*.

Real-world testbed experiments are executed by initializing the TDoA solver with knowledge about the anchor positions and then feeding as input the packet traces these anchors collect during a run. The solver operates based on the principles outlined in §II. In our case, as further detailed in §VI-A, the testbed is composed by 12 nodes deployed on the ceiling of corridors at our office premises. While the scale of this testbed is significantly larger than commonly reported setups [6]–[8], it only yields a network diameter of 3 hops. Further, the geometric characteristics of corridors are ill-suited to demonstrate the localization accuracy we can attain.

Therefore, we complement our testbed experiments with simulation, whose accurate modeling of timing inaccuracies—the main source of localization errors in TDoA—is nonetheless directly informed by real-world traces. The latter have the sole purpose of gathering enough data about the timing behavior of UWB nodes in the target scenario, e.g., due to temperature gradients. The simulator receives as input these real traces along with the actual anchor positions of the intended deployment, and outputs synthetically-generated packet traces with desired duration and frequency that faithfully represent the real traces one would observe in the same conditions.

We now describe in more detail the techniques we use to perform this accurate modeling of timing inaccuracies at the core of our simulation approach, along with experimental results confirming their accuracy.

V. Modeling and Reproducing Timing Inaccuracies

Timing inaccuracy is one of the main challenges in TDoA systems, as timing must be determined with a very fine granularity: a 1 ns timestamp estimation error translates in a 30 cm distance estimation error. At this granularity, inaccuracies arise from two main sources. The first one is the clock drift of typical COTS oscillators, which changes in time and can amount to several ppm. The other one is the error introduced by the timestamping of packets upon reception.

Impact of Packaging on Clock Drift. While performing the experiments in preparation of this paper, we were using both the fixed nodes in our testbed infrastructure as well as some spare nodes that we could position in various configurations to test different topologies and anchor densities.

However, when analyzing the clock drift, we noticed that the two exhibited very different trends, caused by the interplay between packaging and environmental factors. Indeed, testbed nodes are enclosed in a plastic box attached to the ceiling. This "boxed" configuration protects the electronics from temperature variations induced by air movement (e.g., caused by passersby) that, albeit minimal in absolute, do induce nanosecond-scale timing variations. In contrast, the nodes we used for our ad hoc experiments were "naked" (i.e., without

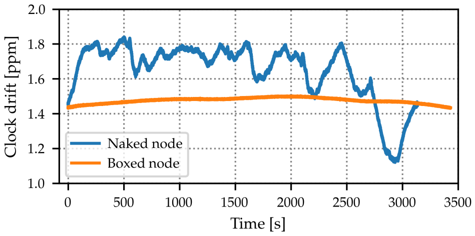


Figure 3. Impact of packaging on clock drift.

a protective case) and therefore prone to such environment-induced timing errors. We verified this phenomenon by placing each type of node in exactly the same position in our indoor office environment, observing dramatically different dynamics of clock drift (Figure 3). The clocks of boxed nodes are slow-changing and variations are within 0.1 ppm; instead, the clock of naked nodes varies abruptly and in a range of almost 1 ppm.

We argue that this finding is of practical interest to developers and users of TDoA systems, as it is common to have boxed nodes in the final deployment but use naked nodes during preliminary tests. In any case, our simulation framework can accommodate and faithfully reproduce both, as discussed next.

Modeling and Reproducing Clock Drift. To analyze and model the clock drift, we acquire hour-long traces from several pairs of UWB devices. In each pair, one device transmits packets at a 10 Hz frequency, and the other logs the corresponding RX timestamps reported by the radio. We then compute the clock drift between every two consecutive timestamps (§II), obtaining a *reference* clock drift signal capturing the real-world timing behavior of our UWB devices.

To generate a *synthetic* drift signal, we create a time signal with the same power spectral density (PSD) as the reference. To this end, our simulator performs the following steps:

i) obtain the frequency domain version of the reference drift signal by applying the Fast Fourier Transform (FFT) and compute the amplitude $A(f)$ for each frequency;

ii) assign a random phase $\phi(f) \in [0, 2\pi)$ from a uniform distribution to each frequency;

iii) build a new frequency signal as $A(f)e^{j\phi(f)}$ and apply the inverse FFT to obtain the time domain version;

iv) filter the generated synthetic signal with a 1 Hz digital low-pass Butterworth filter to eliminate the random RX timestamping error introduced by the radio (see below);

v) upsample the generated signal to match the timeslot frequency of the simulated system.

To generate an arbitrary number of signals, we repeat the process assigning different random phases $\phi(f)$ to each signal.

Modeling and Reproducing RX Timestamping Errors. Another source of timing error is induced by inaccuracies in determining the time of arrival of the received radio signal. The procedure we illustrated above intentionally filters out these inaccuracies to obtain undisturbed clock drift curves. Therefore, the simulator reintroduces the timestamping error for received packets that follows a zero-mean random normal distribution $\mathcal{N}(0, \sigma^2)$. The corresponding parameters, as with clock drift, are determined based on real-world experiments. We estimate the standard deviation σ of the error with several short tests using a higher sampling frequency of 100 Hz, logging the RX timestamps. We then isolate the RX timestamping error by using the same 1 Hz low-pass filter of step *iv)*, this time subtracting the trend imposed by the clock drift. The result of this analysis showed that the measured σ lies in the $[0.14, 0.18]$ ns range for our setup; hereafter, we use $\sigma = 0.18$ ns, as we verified experimentally that this yields the best match between simulation and reality, discussed next.

Simulation vs. Real-world. The techniques we described are at the core of the simulator we use in this paper. The simulator generates synthetic packet traces whose timing error is *different* from the real ones, but whose dynamics are *very similar.* This can be visually ascertained in Figure 4, showing that the simulated curves faithfully reproduce the trends of the measured ones for both boxed and naked nodes. The next section provides further evidence for this claim, as we show that indeed the localization accuracy obtained by our simulator is very close to the one of our real-world prototype, confirming that our model precisely reproduces timing errors.

As a final note, topology does not bear an impact on these timing errors. Therefore, the same reference curve can be used for simulated nodes in arbitrary positions; when generating the RX timestamp traces, the simulator accounts for the propagation time between the anchors, whose position is known (Figure 2). Of course, other deployment effects (e.g., non-line-of-sight propagation) may affect system performance; however these are outside the scope of this study.

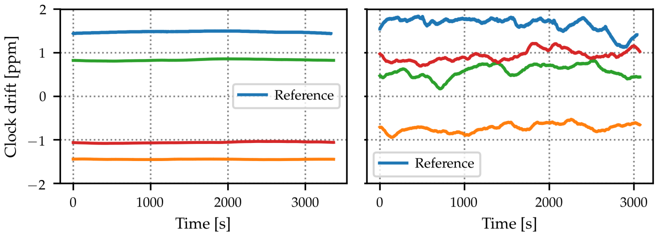
VI. Evaluation

We assess the ability of TALLA to provide continuous and accurate localization over large-scale areas, using simulation and testbed experiments as outlined in §IV.

A. Experimental Setup

Hardware, Firmware, Software. We employ the DecaWave EVB1000 platform [9], featuring an STM32F105 MCU and the DW1000 UWB transceiver with a PCB antenna. The firmware is implemented atop Contiki OS, using the port for the EVB1000 described in [10]. Finally, our TDoA solver and our simulator are implemented in Python, based on the techniques outlined in §II and §V, respectively.

RX/TX Timestamp Calibration. Our TDoA solver uses both the TX timestamps of the reference anchor and the RX timestamps of others. We noticed that, without calibration, location



(a) Boxed nodes.

(b) Naked nodes.

Figure 4. Measured (reference) and simulated clock drift curves.

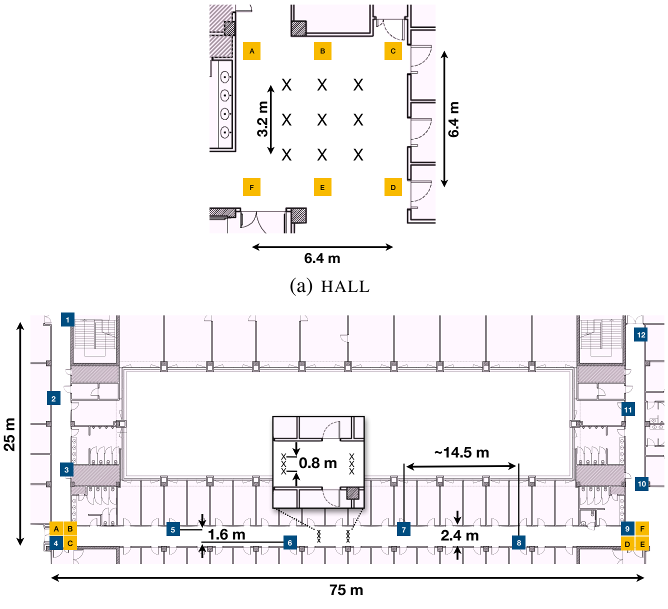
estimates show a decimeter-level bias towards the reference anchor. We attribute this to a discrepancy between TX and RX timestamps, probably caused by a difference between the TX/RX antenna delays. To compensate, we apply a constant correction of -53 ticks ($-0.83\,\text{ns}$) to the TX timestamps.

Experimental Facilities. Our testbed consists of 12 fixed nodes attached to the ceiling of our office building. Each UWB node is connected to a JTAG programmer and a Raspberry Pi. The ceiling nodes use a wired Ethernet infrastructure for automated experiment control and collection of the logs. This setup allows us to easily schedule and run many experiments without handling the nodes individually. Moreover, we used 7 additional portable nodes that could be placed anywhere in the building and/or used as the mobile tags being positioned.

Anchor Placement. We define two setups (Figure 5): *i)* HALL, a regular deployment of 6 portable anchors placed $70\,\text{cm}$ above floor on the perimeter of a $6.4\times$ m^2 square, and *ii)* CORRIDOR, a significantly sparser deployment along a U-shaped corridor whose ceiling hosts our fixed testbed nodes.

These two setups have very different characteristics. In HALL, all nodes are within communication range of each other. Further, the square deployment provides a good (low) dilution of precision (DOP) [11] along both axes. HALL is representative of typical scenarios for indoor localization (e.g., a room) and therefore we take it as our baseline for evaluating our system. Moreover, we also base our large-scale simulation in §VI-C on HALL, by replicating its topological characteristics over a much bigger area with over a hundred anchors, while faithfully reproducing the corresponding timing errors (§V).

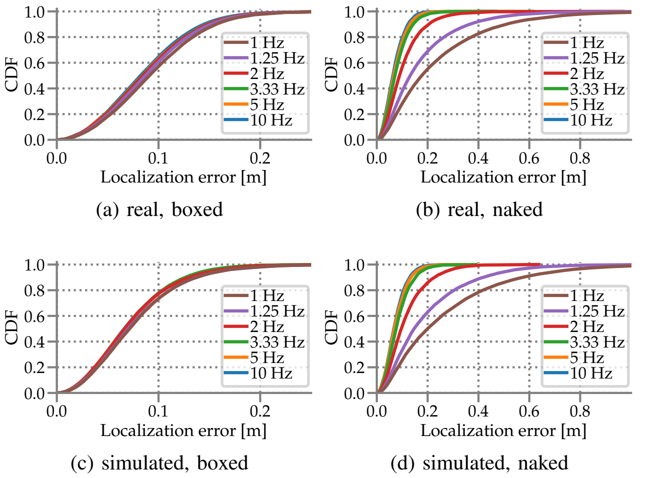
In contrast, the topology of CORRIDOR is definitely suboptimal. The width of the corridor area is only $2.4\,\text{m}$, generating significant multipath effects on the radio signal and yielding a high DOP (i.e., high uncertainty), albeit partially ameliorated by the zig-zag placement of anchors across the ceiling. On the other hand, CORRIDOR is the only "large-scale" testbed we have access to, with a diameter of 3 hops, essentially corresponding to each of the corridor segments (Figure 5b). Therefore, its value lies in the ability to run real-world experiments and *directly* validate our claims, although the localization accuracy we obtain is not indicative of the typical one that we can expect, instead represented by HALL. For these reasons, we evaluate TALLA in both setups.



(a) HALL

(b) CORRIDOR

Figure 5. Anchor deployments. A dark blue square denotes a stationary anchor attached to the ceiling, an orange square stands for a portable anchor, and an X represents a ground-truth landmark.



(a) real, boxed

(b) real, naked

(c) simulated, boxed

(d) simulated, naked

Figure 6. Localization error vs. synchronization rate in HALL (6 anchors).

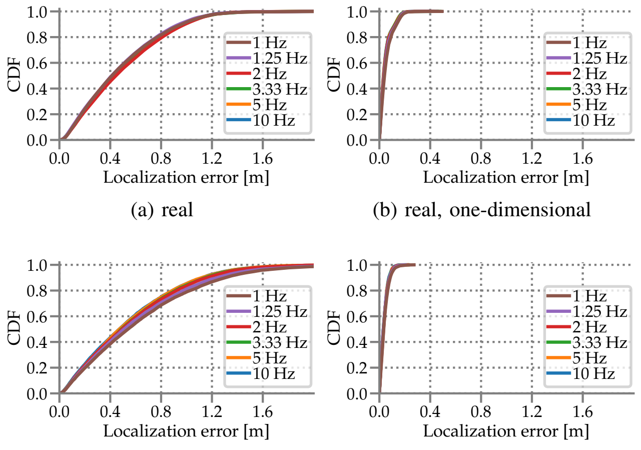
B. Small-Scale, Single-Hop Experiments

We establish a baseline to compare our large-scale scenario against by evaluating the localization accuracy in small-scale experiments representative of common TDoA deployments with all anchors in range. This serves also as a validation of our simulator, showing its ability to faithfully reproduce timing errors and therefore localization accuracy.

We consider HALL (Figure 5a) and the portion of CORRIDOR with nodes 4–9 (Figure 5b). We manually measure with a laser pointer ground-truth coordinates for 9 and 6 landmarks, respectively, and acquire location estimates at each of them at the rate of 90 samples/s for 5 minutes, yielding ≈27000 samples per landmark. In each setup, one anchor serves as time reference, periodically broadcasting synchronization beacons.

Synchronization Rate vs. Localization Accuracy. TDoA is very sensitive to clock drift; the rate at which time synchronization beacons are sent affects the system performance. Hereafter, we experiment with rates between 1 Hz and 10 Hz.

Figure 6 shows the corresponding CDFs of localization error in HALL, for both boxed and naked nodes. The setup with boxed nodes show excellent localization accuracy; 99% of the samples fall within $20\,\text{cm}$ of the ground-truth landmarks, regardless of the synchronization rate (Figure 6a). As mentioned in §V, the encasing protects nodes from temperature variations and yields very slow changes in the clock drift, which can be compensated precisely (§II). The same holds in the case of naked nodes, but only when the synchronization rate is $\geq$3.3 Hz; otherwise, the variations induced by the



(a) real

(b) real, one-dimensional

(c) simulated

(d) simulated, one-dimensional

Figure 7. Localization error vs. synchronization rate in CORRIDOR (6 anchors, boxed only). Note the different x-axis scale w.r.t. Figure 6.

Table I
LOCALIZATION ERROR (CM) FOR FIGURE 6–7 WITH A 3.3 HZ SYNCHRONISATION RATE.

			percentile			
			75	90	95	99
real	HALL	boxed	11	13	15	21
		naked	10	14	16	21
	CORRIDOR	2D	69	89	101	377
		1D	6	9	10	14
simulated	HALL	boxed	9	12	14	18
		naked	9	11	14	18
	CORRIDOR	2D	81	112	130	531
		1D	5	7	9	12

Table II
IMPACT OF THE NUMBER OF ANCHORS ON LOCALIZATION ERROR (CM).

		4 anchors				8 anchors			
percentile		75	90	95	99	75	90	95	99
HALL	boxed	11	14	17	22	8	11	12	16
	naked	11	15	17	23	9	11	14	18
CORRIDOR	boxed 2D	86	118	143	600	81	113	132	538
	boxed 1D	6	9	10	14	4	6	8	10

environment cannot be successfully compensated (Figure 6b). Figure 8 offers an alternate view for real experiments, showing the actual spread of sample points with error ellipses denoting the 3× standard deviation of the error, further reasserting the impact of packaging on clock drift and localization accuracy.

Figure 7 shows analogous results for the portion of CORRIDOR considered. In this case, we show only the case with boxed anchors, as they reflect the actual deployment. As in HALL, the synchronization rate does not bear a significant impact in the value range considered. However, as expected, the localization accuracy (Figure 7a) is significantly worse than in HALL, due to the very narrow geometry of the setup—a rectangle $2.4\,\mathrm{m}$ wide and $75\,\mathrm{m}$ long. Given this extreme shape, the localization accuracy is actually still reasonable, but not very indicative. Moreover, we observe that often what matters in a corridor is to localize where the target is along its length rather than a fine-grained localization including its width. Therefore, we also report the results for this one-dimensional case, which show high accuracy (Figure 7b).

Simulation vs. Reality. The bottom rows of charts in Figure 6–7 show the results output by our simulator when configured to replicate the real setup. Comparison between real and simulated results confirm that the simulator faithfully reproduces the real-world behavior of our TDoA localization. The small difference between real and simulated results can also be appreciated in Table I, for various error percentiles. The largest difference is found in the CORRIDOR with two-dimensional localization, a scenario that is severely challenged by geometry and multipath effects. In all other cases, simulated and real results are in very good agreement. This confirms that we can safely use the simulator to analyze the performance of TDoA systems in general, including large-scale deployments.

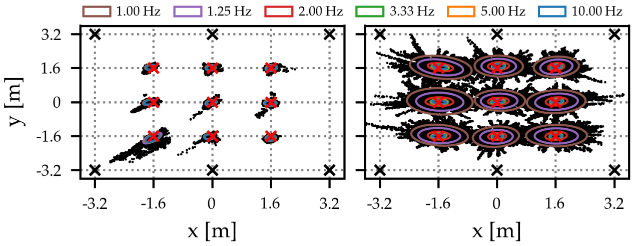


Figure 8. Localization error vs. synchronization rate in real HALL experiments with boxed (left) and naked (right) nodes. The external black crosses are anchors, the internal red ones are landmarks. Error ellipses denote the 3×standard deviation for a given rate. Black dots are individual samples.

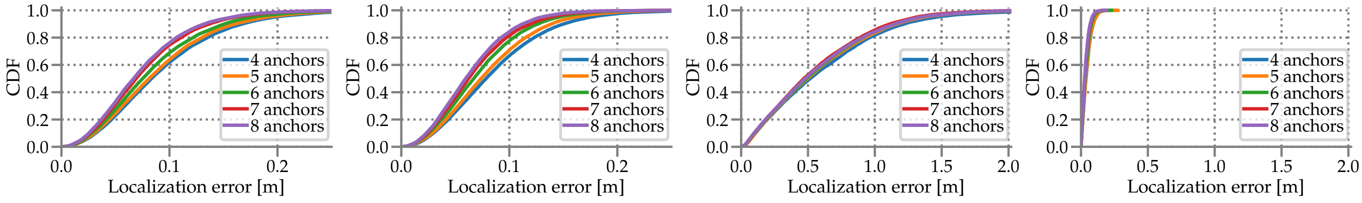
Number of Anchors vs. Localization Accuracy. Empowered by our simulator, we study an aspect crucial to deployments: the effect of the number of anchors on localization accuracy.

We explore a number of anchors ranging from 4 (i.e., the minimum required by TDoA) and 8. In HALL, we obtain the minimum by removing anchors B and E; dually, we obtain the maximum by inserting one anchor in the middle of A–F and C–D. In CORRIDOR we start from anchors 5–8 and mirror them by adding an anchor on the other side of the corridor, most outer ones first. Figure 9 shows the CDFs in various configurations, and Table II shows the corresponding values. Interestingly, both HALL and CORRIDOR appear to be only marginally affected by the number of anchors, even in the case of naked boxes. We verified that this is the case also in the corresponding real-world experiments.

C. Large-scale, Multi-hop Experiments

We evaluate our system in two large-scale, multi-hop scenarios. The first one, GRID, is a synthetically generated one, taking advantage of our simulator. GRID can be seen as a tiling of the HALL scenario across a much larger area. The other scenario is CORRIDOR, enabled by our real-world testbed.

Simulation Experiments: GRID. We model two large rectangular fields connected by a "passageway", and fill the area with a regular grid of anchors with a step of L. This configuration combines both wide, open areas and a narrow one, and is representative of situations found in several indoor and outdoor large areas, e.g., halls of large buildings, parking lots, etc. The rectangles fit 9×9 anchors each, while the passageway contains 9×2 anchors, totaling 180 nodes (Figure 10). To



(a) HALL, naked (b) HALL, boxed (c) CORRIDOR, boxed (d) CORRIDOR, boxed, 1-D

Figure 9. Effect of the number of anchors on the error distribution, 3.3 Hz synchronization rate, in simulation.

ensure uniform coverage through this anchor placement, we set the communication range to $3 \times L$; geometrical considerations guarantee that a tag is always in range of 4 to 9 anchors. This results in a network diameter of 26 hops.

In TALLA, all N anchors broadcast synchronization beacons; however, the number n of their time slots in an epoch can be defined to be $n \ll N$ (§III). Indeed, anchors 3 hops away from each other do not interfere and can safely reuse the same slot. Therefore, $n = 9$ slots can accommodate synchronization beacons from all $N = 180$ anchors in GRID.

Figure 10 shows an example of tracking a mobile tag in GRID; the positions estimated by TALLA are compared against a ground truth trajectory. As the tag moves, the set of anchors used for positioning changes, and so does the number of anchors (Figure 11a). Nevertheless, position estimates do not exhibit gaps, witnessing the ability of TALLA to support *continuous* localization across large areas. Moreover, Figure 11b also demonstrates that this result is achieved *without sacrificing localization accuracy*. Indeed, the median error is 6 cm and the maximum is below $<$ 30 cm across all 3000 points visited by the tag; these results are comparable to those we observed for HALL in both simulation and reality (§VI-B).

Testbed Experiments: CORRIDOR. We demonstrate the effectiveness of TALLA holds also in reality, although in a more limited setup w.r.t. the synthetic one in GRID. In these experiments, a person carrying the tag walks at a speed of ≈ 0.5 m/s in the center of the CORRIDOR U-shaped path, from one extreme to the other. As the person passes the corners, the system dynamically switches the sets of anchors it relies on.

The anchor placement in CORRIDOR is slightly different w.r.t. the one in §VI-B. Our ceiling testbed was originally motivated by the need to run experiments about communication quality. However, while experimenting with TALLA we discovered that the testbed node layout is problematic for localization; a few nodes (3, 5, 10) near the corners are in communication range with the tag but not in line of sight, causing a large positioning error. Therefore, we removed these from the analysis and added 6 portable anchors (A–F), 3 in each corner, to ensure line of sight (Figure 5b).

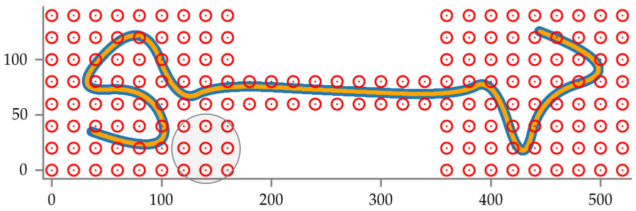


Figure 10. GRID: ground truth (blue) vs. estimated position (yellow).

Figure 12 shows the results obtained from our experiments, plotting the tag trajectory in the CORRIDOR area. Our testbed setup is not equipped to gather precise ground-truth information about the location of the tag. Therefore, we encode the time information associated with the estimated trajectory as a color gradient. This enables us to visually ascertain the correct and continuous operation of TALLA localization.

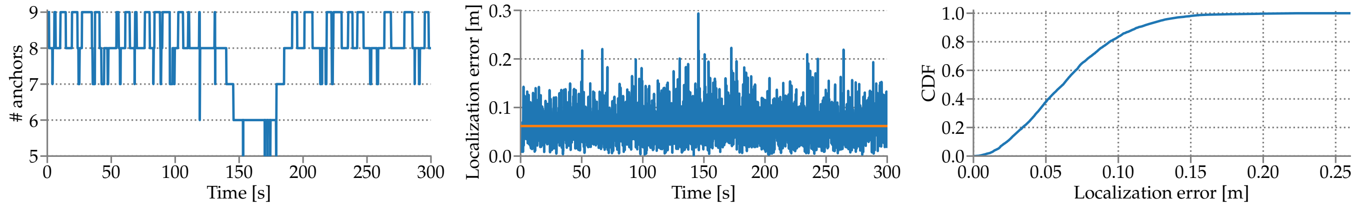
As for localization accuracy, we analyze different configurations of our TDoA solver. When configured without any knowledge of the environment, the solver outputs the results in Figure 12a, showing a high variance along the axis perpendicular to the corridor and suboptimal performance around corners due to residual non-line-of-sight and poor DOP in those areas. Nevertheless, the one-dimensional positioning—arguably the relevant one in a corridor area like ours—remains very accurate, as confirmed by the smooth color gradient.

However, in a practical deployment targeting a geometrically-challenged area like CORRIDOR, it is common (and easy) to configure the TDoA solver with knowledge of the area. Figure 12b shows that, by simply setting the outer boundaries of the corridor at the corners, the solver not only automatically drops unreasonable estimates but also improves them. This simple modification, whose nature has to do with the "local" TDoA localization and not with our multi-hop scheme, enables TALLA to successfully track the tag without gaps across the 3 hops in CORRIDOR, with a localization accuracy similar to what we observed in §VI-B. Finally, Figure 12c shows the results when one-dimensional localization along the corridor path is performed. As observed in §VI-B, the accuracy is even higher, and the correct and continuous operation of TALLA even more evident.

VII. Related Work

Recent research on UWB TDoA localization consists of *i)* conventional schemes, with tightly-synchronized anchors listening to mobile tag transmissions, and *ii)* reversed schemes, where tags are passive listeners of anchor transmissions.

Conventional TDoA. TALLA builds on the mechanisms in [4] and the ATLAS localization system [5], [6]. ATLAS is based on *i)* a 1-hop star topology, resulting in limited spatial scalability, and *ii)* random access to the wireless medium, leading to performance degradation as the tag number or positioning rate increase [3]. To avoid collisions between tags and anchors, ATLAS assigns a different UWB preamble code for

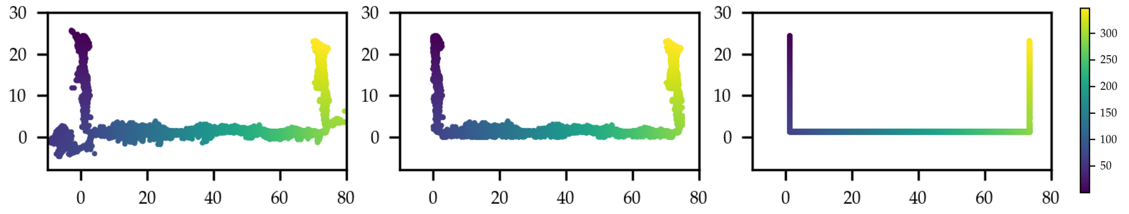


(a) Number of anchors used over time.

(b) Localization accuracy over time.

(c) Overall localization accuracy.

Figure 11. GRID: Number of anchors and localization accuracy.



(a) Unbounded

(b) Bounded

(c) One-dimensional

(d) Time, s.

Figure 12. Estimated trajectories in the CORRIDOR. The color gradient represents time, axis values are in meters.

synchronization, possibly leading to timestamping errors [12]. TALLA's TDMA schedule tackling large areas implicitly solves this issue, avoiding collisions altogether. Another significant difference, besides implementation details, is the ATLAS dependency on a dedicated external node for synchronization; this is not required in TALLA, which simplifies deployment by relying solely on the required anchors.

A recent extension, ATLAS-FaST [13], targets high-rate positioning also via a TDMA scheme. The authors mention that this could be exploited to scale to large areas, but do not provide experimental evidence for this claim. In contrast, we evaluate TALLA with both real-world and simulated experiments. Further, the methodology and techniques in the latter (§IV–§V) are a contribution per se, enabling further developments in the general field of TDoA localization.

Reversed TDoA. In Chorus [7] and SnapLoc [8] anchors transmit beacons concurrently, and tags self-position by estimating TDoA from anchor peaks in the channel impulse response (CIR). Loco [14] instead employs staggered beacons, transmitted sequentially. These systems support high update rates for unlimited tags, as these are passive listeners. Positioning information remains local to the tags, which is an asset for some applications and a drawback in others (e.g., logistics). None of these solutions addresses large-scale operation.

VIII. Conclusions and Future Work

We presented TALLA, a novel TDoA scheme enabling continuous and accurate localization across large-scale operational areas. We evaluated the TDMA design of TALLA both in a real-world and simulated setups; in the latter, we designed a novel technique enabling us to faithfully reproduce the timing inaccuracies of UWB devices causing TDoA positioning errors. Results show that TALLA achieves state-of-the-art accuracy in small-scale single-hop settings but also when targets move and change their set of anchors in range, confirming that TALLA enables uninterrupted, reliable positioning across large areas.

Future work on the topic revolves around the availability of a much larger testbed, currently under deployment at our premises. This will enable real-world experiments in much larger, and less geometrically challenged, setups than the CORRIDOR one used here, including also the ability to assess three-dimensional localization accuracy.

Acknowledgements. The authors are grateful to Francesco Giopp, whose M.Sc. thesis [15] explored the core ideas described here, and Davide Molteni, for his help in performing testbed experiments. This work is partially supported by EIT Digital via the D-TWIN project (18388) and by the Italian government via the NG-UWB project (MIUR PRIN 2017).

References

[1] DecaWave Ltd. DW1000 Data Sheet, 2016.

[2] Y. Jiang and V. C. M. Leung. An Asymmetric Double Sided Two-Way Ranging for Crystal Offset. In *Proc. of ISSSE*, 2007.

[3] M. Ridolfi et al. Analysis of the scalability of UWB indoor localization solutions for high user densities. *Sensors*, 18(6):1875, 2018.

[4] C. McElroy et al. Comparison of wireless clock synchronization algorithms for indoor location systems. In *Proc. of ICC*, 2014.

[5] J. Tiemann et al. Multi-User Interference and Wireless Clock Synchronization in TDOA-based UWB Localization. In *Proc. of IPIN*, 2016.

[6] J. Tiemann et al. ATLAS - An Open-Source TDOA-based Ultra-Wideband Localization System. In *Proc. of IPIN*, 2016.

[7] P. Corbalán et al. Chorus: UWB Concurrent Transmissions for GPS-like Passive Localization of Countless Targets. In *Proc. of IPSN*, 2019.

[8] B. Großwindhager et al. SnapLoc: An Ultra-fast UWB-based Indoor Localization System for an Unlimited Number of Tags. In *Proc. of IPSN*, 2019.

[9] DecaWave Ltd. DecaWave ScenSor EVB1000 Evaluation Board, 2013.

[10] P. Corbalán, T. Istomin, and G. P. Picco. Poster: Enabling Contiki on Ultra-wideband Radios. In *Proc. of EWSN*, 2018.

[11] R. B. Langley. Dilution of precision. *GPS world*, 10(5):52–59, 1999.

[12] D. Vecchia et al. Playing with Fire: Exploring Concurrent Transmissions in Ultra-wideband Radios. In *Proc. of SECON*, 2019.

[13] J. Tiemann et al. ATLAS FaST: Fast and Simple Scheduled TDOA for Reliable Ultra-Wideband Localization. In *Proc. of ICRA*, 2019.

[14] Bitcraze. The Loco Positioning System. https://www.bitcraze.io/loco-pos-system/, 2019. Online; last accessed May 25, 2019.

[15] F. Giopp. *Real-time Tracking of Multiple Users Using UWB: A TDOA-based Approach.* M.Sc. Thesis, University of Trento, October 2018.