Accurate Ubiquitous Localization with Off-the-Shelf IEEE 802.11ac Devices

Alejandro Blanco Pizarro
alejandro.blanco@imdea.org
IMDEA Networks Institute
Universidad Carlos III de Madrid
Madrid, Spain

Joan Palacios Beltrán
jbeltra@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Marco Cominelli
marco.cominelli@unibs.it
University of Brescia
Brescia, Italy

Francesco Gringoli
francesco.gringoli@unibs.it
University of Brescia/CNIT
Brescia, Italy

Joerg Widmer
joerg.widmer@imdea.org
IMDEA Networks Institute
Madrid, Spain

ABSTRACT

WiFi location systems are remarkably accurate, with decimeter-level errors for recent CSI-based systems. However, such high accuracy is achieved under Line-of-Sight (LOS) conditions and with an access point (AP) density that is much higher than that typically found in current deployments that primarily target good coverage. In contrast, when many of the APs within range are in Non-Line-of-Sight (NLOS), the location accuracy degrades drastically.

In this paper we present UbiLocate, a WiFi location system that copes well with common AP deployment densities and works ubiquitously, i.e., without excessive degradation under NLOS. UbiLocate demonstrates that meter-level median accuracy NLOS localization is possible through (i) an innovative angle estimator based on a Nelder-Mead search, (ii) a fine-grained time of flight ranging system with nanosecond resolution, and (iii) the accuracy improvements brought about by the increase in bandwidth and number of antennas of IEEE 802.11ac. In combination, they provide superior resolvability of multipath components, significantly improving location accuracy over prior work. We implement our location system on off-the-shelf 802.11ac devices and make the implementation, CSI-extraction tool and custom Fine Timing Measurement design publicly available to the research community. We carry out an extensive performance analysis of our system and show that it outperforms current state-of-the-art location systems by a factor of 2-3, both under LOS and NLOS.

CCS CONCEPTS

• **Networks → Location based services**.

KEYWORDS

Indoor localization, CSI, 802.11ac, AoA, ToF, Wireless Networks

ACM Reference Format:
Alejandro Blanco Pizarro, Joan Palacios Beltrán, Marco Cominelli, Francesco Gringoli, and Joerg Widmer. 2021. Accurate Ubiquitous Localization with Off-the-Shelf IEEE 802.11ac Devices. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24-July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3458864.3468850

1 INTRODUCTION

Wireless localization and sensing have become important applications of wireless communications, and the accuracy of such systems has improved substantially over the past two decades of research. While the first works that pioneered this field [6, 60] had an accuracy on the order of several meters at best, recent designs [5, 27] provide highly accurate location estimates with errors of a few decimeters. For this, location systems use a range of different approaches. With multi-antenna systems, Angle of Arrival (AoA) and/or Angle of Departure (AoD) information from incoming/outgoing signals can be estimated by means of array processing techniques. With sufficiently many Access Points (APs) or anchors with known location, target devices can then be located through triangulation [27, 31, 55]. When Time of Flight (ToF) or Time Difference of Arrival (TDOA) information is available for ranging, classical trilateration methods are applicable [41, 56, 57]. Combining these methods further improves accuracy, and some works even propose single-AP localization using both angle and ranging information [34, 40, 47].

For sub-meter accuracy, WiFi location systems typically extract radio signal features from the Channel State Information (CSI) to derive accurate angle and timing information of the Line-Of-Sight (LOS) path. While this is straightforward on software-defined radio systems, location systems that work on off-the-shelf devices are easier to deploy and have a much larger practical impact. The most prominent off-the-shelf devices that provide CSI information are the Intel 5300 cards [21] for the IEEE 802.11n standard. They are used by virtually all CSI-based off-the-shelf systems [8, 25, 27, 54]. Despite the good location accuracy, 802.11n is already a decade old and newer standards such as 802.11ac and 802.11ax can potentially provide even better performance.

At the same time, the good accuracy of prior systems is only achieved when the device to be located has unobstructed LOS to several APs (or at the very least one AP), and performance degrades

MobiSys '21, June 24-July 2, 2021, Virtual, WI, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8443-8/21/06…$15.00
https://doi.org/10.1145/3458864.3468850

considerably under Non-Line-Of-Sight (NLOS) conditions. Dealing with NLOS is extremely challenging. While angle and timing information from obstructed LOS paths that pass through obstacles may still provide useful information, it is very difficult to distinguish obstructed LOS paths from true NLOS paths coming from reflections. This is an important shortcoming since typical large scale WiFi deployments have a number of APs per building floor but fewer than one AP per room, and NLOS conditions are very common. Several works tackle NLOS scenarios using ultra-wideband technology [12, 37] or software-defined radios for through-wall imaging and mapping [1, 51]. However, to the best of our knowledge, there is no general-purpose WiFi location system that provides adequate performance under true NLOS conditions.[1]

In this paper we present UbiLocate, a ubiquitous WiFi location system that works both under LOS and NLOS conditions. We achieve good NLOS localization through the improvements brought about by 802.11ac in terms of bandwidth and number of antennas, in combination with novel signal processing for multipath decomposition, that jointly help to resolve multipath effects much more accurately. Our paper makes the following main contributions:

- **Optimized AoA extraction.** Classic algorithms such as MUSIC [46] and ESPRIT [43] have been widely used to analyze RF signals for path parameter estimation, especially AoA. [2, 31, 47]. Recently, compressed sensing techniques have been demonstrated to provide better accuracy [15, 33, 59]. However, their application can be computationally prohibitive in common scenarios. In order to reduce the computational complexity, UbiLocate iteratively determines a first estimate of the path parameters and then refines it through a Nelder-Mead search [30]. This minimization results in a more accurate multipath decomposition, and UbiLocate achieves an AoA accuracy improvement of a factor of 2 for LOS and 1.5 for NLOS settings compared to state-of-the-art algorithms [27].
- **Controlled Ranging.** Estimating the absolute ToF and thus the distance between client and AP requires timestamped packet exchanges, as standardized in the 802.11 Fine Timing Measurement (FTM) protocol [23, 24]. However, FTM is inaccurate in multipath-rich environments [26]. UbiLocate uses a custom protocol similar to FTM that has lower overhead and is more robust by decomposing the multipath channel to accurately determine the ToF of the first path. Again, UbiLocate improves the ToF estimation accuracy by a factor of 2 for LOS and and 1.5 for NLOS compared to plain FTM.
- **Filtering reliable APs.** Depending on the specific scenario, the estimates from different APs have different fidelity. When averaging the location information provided by all APs, low quality estimates may contaminate the overall location accuracy. UbiLocate therefore includes a mechanism to evaluate the quality of different estimates, giving more weight to the APs that provide good estimates.
- **Implementation on off-the-shelf devices.** We implement the UbiLocate system on off-the-shelf Asus AC2900 RT86U routers that support IEEE 802.11ac with 4x4 Multiple-Input Multiple-Output (MIMO) and 80 MHz of bandwidth. The improved hardware capabilities increase localization accuracy since the larger bandwidth and number of antennas allow for better time and space resolution. We can thus extract the path parameters more accurately than with the older IEEE 802.11n standard. We modify the router firmware to access CSI in order to estimate AoA, AoD, and ToF. (i) UbiLocate is the first location system implemented on off-the-shelf devices that works with 80 MHz WiFi channels and does not require a non-disclosure agreement (the existing 80 MHz location systems [5, 40] use Quantenna devices that require such a non-disclosure agreement). (ii) It is also the first IEEE 802.11ac-based location system that can simultaneously derive both angle and absolute distance to a target device, whereas prior work uses two separate co-located devices for this purpose [26, 40].

We deploy UbiLocate in a large office environment and test it with different AP densities and with both LOS to NLOS measurement points. Our performance evaluation shows that UbiLocate achieves meter-level median accuracy even for pure NLOS and low AP density scenarios. It outperforms current state-of-the-art systems by a factor of 2-3. Finally, we release our tool to extract CSI and perform FTM-like ranging to the research community to foster wireless systems research with 802.11ac. We believe that it will prove similarly useful as the widely used CSI tool for 802.11n [21], given the hardware improvements offered by 802.11ac. The CSI extractor tool with the modified firmware and documentation are available in a github repository [3].

2 UBILOCATE OVERVIEW

UbiLocate locates a wireless device using AoA, AoD, and ToF information. This is relatively straightforward when several APs with direct LOS are within range. However, typical indoor WiFi deployments do not provide ubiquitous LOS coverage since NLOS links can provide sufficiently high data rates.

In such complex environments with NLOS, the multipath channel and the resulting superposition of different signals at the receiver significantly affects the quality of the location estimate. Even under pure NLOS, good location accuracy is feasible as long as the location system can discriminate between obstructed LOS paths and the NLOS paths coming from reflections. The latter must be discarded, since they lead to erroneous angle and ToF estimates. By definition, obstructed LOS paths pass through an obstacle, and thus their signal power may be severely attenuated compared to other NLOS paths. Accurately detecting them requires a fine-grained multipath decomposition of the channel.

As is common for wireless location systems, we assume that the positions of the APs are fixed and known. To discriminate the multipath components, UbiLocate minimizes the norm of the difference between the observed received signal and estimated superimposed signals and their path parameters. The number of possible combinations of path parameters makes brute force minimization computationally prohibitive, but if an approximate estimate is known, the minimization can be sped up significantly. To this end, we first compute rough estimates of the path parameters and then refine them through a Nelder-Mead search [35]. This provides better accuracy than the widely used MUSIC and similar approaches [27] which resolve the paths in one round. UbiLocate iteratively estimates the parameters of the strongest path and then subtracts them from the received signal. This allows UbiLocate to estimate the parameters of weak paths that would otherwise be masked by stronger ones and is especially critical in NLOS environments. In contrast to prior

[1] While some systems [27, 45] claim to analyze "NLOS scenarios", these scenarios do in fact have LOS to one or more APs in almost all cases.

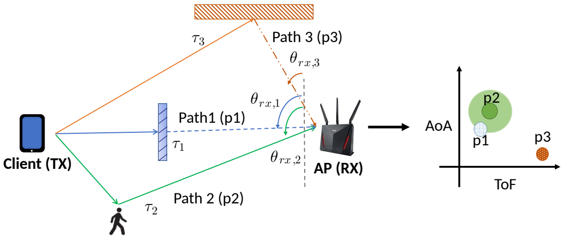


Figure 1: NLOS example with obstructed LOS path.

iterative approaches [13, 20, 48], we further refine the estimation to remove imperfections which leads to improved angle accuracy in the challenging cases we target in this paper.

Fig. 1 shows a typical NLOS scenario in which reflected paths may be stronger than the obstructed LOS one. In addition, as shown on the right in this example figure, path $p1$ and path $p2$ are close in time and angle, and the uncertainty around path $p2$ makes it hard to discriminate the two. In such a scenario, accurate multipath decomposition is key for good location system performance.

2.1 Path parameters

Consider a MIMO system where the transmitter and the receiver have uniform linear arrays of L and M antennas with antenna spacing of half a wavelength. The transmitter sends a set of OFDM signals $s[k] = [s_0[k], s_1[k], ..., s_{L-1}[k]]$ over K subcarriers and L antennas. The signals propagate through a multipath channel with P different paths and arrive at the receiver, characterized by:

• **Complex attenuation γ_p.** The signal suffers an attenuation of γ_p along path p.

• **Angle of arrival $\theta_{rx,p}$.** The signal arrives at each antenna with a phase delay determined by the antenna spacing. The phase shift $[\phi(\theta_{rx,p})]_m$ at the m^{th} receive antenna as function of the AoA for the p^{th} path is given by:

$$[\phi(\theta_{rx,p})]_m = e^{-j\pi(m-1)\sin(\theta_{rx,p})} \quad . \tag{1}$$

The vector of phase shifts for the whole array is:

$$\boldsymbol{\phi}(\theta_{rx,p}) = [\phi(\theta_{rx,p})]_0, ..., [\phi(\theta_{rx,p})]_{M-1} \quad . \tag{2}$$

• **Angle of departure $\theta_{tx,p}$.** Similarly, $[\phi(\theta_{tx,p})]_l$ is the phase shift for the l^{th} transmit antenna as a function of the AoD:

$$[\phi(\theta_{tx,p})]_l = e^{-j\pi(l-1)\sin(\theta_{tx,p})} \tag{3}$$

and we denote the vector of phase shifts for the whole array by $\boldsymbol{\phi}(\theta_{tx,p})$.

• **Path delay τ_p.** Each path p experiences a different propagation delay determined by its length. In the frequency domain, this delay represents a phase shift $\psi(\tau_p)[k]$ between adjacent subcarriers:

$$\psi(\tau_p)[k] = e^{-j2\pi k\Delta_f \tau_p} \quad , \tag{4}$$

where Δ_f is the spacing between consecutive subcarriers.

With the parameters above, we can express the channel as follows:

$$\mathbf{H}[k] = \sum_{p=0}^{P-1} \boldsymbol{\phi}(\theta_{rx,p})\gamma_p \boldsymbol{\phi}^{\mathrm{H}}(\theta_{tx,p})\psi(\tau_p)[k] \quad , \tag{5}$$

where $(\cdot)^H$ is the Hermitian operator. The received signal is:

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{s}[k] + \mathbf{w}[k] \quad , \tag{6}$$

where $\mathbf{w}[k]$ is L-dimensional white Gaussian noise in the frequency domain, i.e., $\mathbf{w}[k] = [w_0[k], w_1[k], ..., w_{L-1}]$. For a known $\mathbf{s}[k]$, we can then estimate the channel as:

$$\hat{\mathbf{H}}[k] = \mathbf{y}[k]\mathbf{s}^*[k] = \hat{\mathbf{H}}[k] = \mathbf{H}[k] + \hat{\mathbf{w}}[k] \quad , \tag{7}$$

where $(\cdot)^*$ is the conjugate operator. Since the channel provides spatial information about the location of the devices, it needs to be estimated as accurately as possible.

2.2 Angle estimation

For device localization, UbiLocate requires the angles for the direct or obstructed LOS path, provided such a path exists. This path is the one that typically arrives earliest in time before any of the NLOS paths coming from reflections, i.e., the one with the smallest τ_p. Note that the ToF τ_p is not an absolute value but reflects relative delay differences among paths. (For ranging, UbiLocate uses a customized FTM implementation.) While directly using AoD information is not useful due to potential rotation of the device to be located, estimating it jointly with the other path parameters considerably improves the path resolvability [54].

To extract parameters of all paths, our objective is to find an expression for $\mathbf{H}[k]$ that minimizes $\|\hat{\mathbf{H}}[k] - \mathbf{H}[k]\|$. $\hat{\mathbf{H}}[k]$ is the observed channel and $\mathbf{H}[k]$ contains the contribution of each path according to the estimated path parameters. However, minimization by brute force is computationally prohibitive due to the large number of combinations of path parameter. Hence, we split the minimization into two steps. We first perform a greedy matching projection to iteratively estimate the path parameters. We then perform a minimization through Nelder-Mead search based on the extracted path parameters from the first step to refine them.

2.2.1 Greedy estimation. Through greedy matching projection we iteratively compute the contribution of the strongest path, estimate its parameters, reconstruct it, and then subtract it from the overall measured channel. The output of the subtraction is the channel residual and using the residual we can then estimate the second strongest path's contribution, and so on, until the parameters of all significant paths are estimated. This allows to accurately estimate even the weak paths often found in NLOS scenarios, since we first remove the contribution of the stronger ones. As is illustrated in Fig. 1, depending on the properties of the reflectors, paths $p2$ and $p3$ may be significantly stronger than the obstructed LOS path $p1$.

We apply a matching projection to the observed channel and the path parameters that maximize it are the ones from the strongest path $p = 0$. We then remove this path from the observed channel and apply matching projection to the residual to obtain the second strongest path $p = 1$, and so on. In general, in iteration p we extract path p as the strongest path of the residual as:

$$(\tau_p, \theta_{rx,p}, \theta_{tx,p}) = \\ \underset{\tau_p,\theta_{rx,p},\theta_{tx,p}}{\arg\max} \sum_k \boldsymbol{\phi}^H(\theta_{rx,p})\hat{\mathbf{H}}^{\mathrm{r}}_p[k]\boldsymbol{\phi}(\theta_{tx,p})\psi^*(\tau_p)[k] \, , \tag{8}$$

The path parameters produce phase shifts, where $\boldsymbol{\phi}(\theta_{rx,p})$ and $\boldsymbol{\phi}^H(\theta_{tx,p})$ are the phase shifts introduced by the AoA and AoD at receiver and transmitter antennas, and $\psi(\tau_p)[k]$ that of the path length for subcarrier k. We multiply these phase shifts by their conjugates in the projection, so that only the correct path parameters maximize it. The residual in iteration p is given by

$$\hat{\mathbf{H}}^{\mathrm{r}}_{p}[k] = \hat{\mathbf{H}}[k] - \sum_{p'=0}^{p-1} \boldsymbol{\phi}(\theta_{rx,p'})\gamma_{p'}\boldsymbol{\phi}^{\mathrm{H}}(\theta_{tx,p'})\psi(\tau_{p'})[k]\,, \quad (9)$$

where the residual for $p = 0$ is the original channel $\hat{\mathbf{H}}^{\mathrm{r}}_{0}[k] = \hat{\mathbf{H}}[k]$.

To solve the optimization problem in (8), we first determine τ_p. To do so, we convert the channel from the frequency domain to the time domain $\mathbf{H}[t]$, by applying an over-sampled inverse discrete Fourier transform to the channel. In the time domain, the path delay τ_p of the strongest path is directly the time t value that maximizes $\|\mathbf{H}[t]\|$. This channel is given by a combination of *sinc* functions with maxima in the different delays. Now, given τ_p we have

$$\begin{aligned} \mathbf{H}[\tau_p] \quad &= \textstyle\sum_{p'=0}^{p-1} \boldsymbol{\phi}(\theta_{rx,p'})\gamma_{p'}\boldsymbol{\phi}^{H}(\theta_{tx,p'})(\psi^{\mathrm{H}}(\tau_p)\psi(\tau_{p'})) \\ &\simeq \boldsymbol{\phi}(\theta_{rx,p})\gamma_p\boldsymbol{\phi}^{H}(\theta_{tx,p}) \end{aligned} \,, \quad (10)$$

and $\hat{\mathbf{H}}[\tau_p] = \mathbf{H}[\tau_p] + \bar{\mathbf{w}}[\tau_p]$ with noise at the instant τ_p computed as $\bar{\mathbf{w}}[\tau_p] = \sum_{k=0}^{K} \hat{\mathbf{w}}[k]\psi(\tau_p)^{*}[k]$. With this, we can estimate the angle information. Instead of jointly estimating the $\theta_{rx,p}$ and $\theta_{tx,p}$, we first estimate $\theta_{tx,p}$ by a grid search assuming that $\theta_{rx,p}$ is unknown. This results in the following formulation:

$$\max_{\theta_{rx,p},\theta_{tx,p}} \; [1, 0, 0, \ldots]\hat{\mathbf{H}}[\tau_l]\boldsymbol{\phi}(\theta_{tx,p}). \quad (11)$$

Having estimated $\theta_{tx,p}$, we can iteratively refine either angle by a grid-search assuming that the other is known which increases the estimation accuracy. This individual estimation of two parameters is much faster than a joint estimation of two parameters. We refine the angle estimation by maximizing the following expression:

$$\max_{\theta_{rx,p},\theta_{tx,p}} \boldsymbol{\phi}(\theta_{rx,p})^{H}\hat{\mathbf{H}}[\tau_l]\boldsymbol{\phi}(\theta_{tx,p}) \; . \quad (12)$$

Once all parameters for one path are estimated, we recompute γ_p as a linear MMSE solution to minimize the error between the measured channel and the reconstructed one.

2.2.2 Refinement search. The previous estimation of the path parameters may contain imperfections since the paths are highly correlated. This may leak information of the parameters from weaker paths to stronger ones and vice versa. To refine the estimates, we carry out a Nelder-Mead search to minimize $\|\hat{\mathbf{H}}[k] - \mathbf{H}[k]\|$. This optimization method iteratively generates sets of points that compose a simplex polytope. The *gradient expression* for the refinement problem is very complex, whereas below we show how to obtain an *objective function* that is simple to evaluate. This makes Nelder-Mead search a much better fit for the specific problem of multi-path refinement than gradient descent. While Nelder-Mead search itself is well studied, to the best of our knowledge it has never been applied to the problem of path parameter estimation.

We use a vectorized version of the problem $\hat{\mathbf{h}}_{\mathrm{v}} = \Phi\gamma$, with $\hat{\mathbf{h}}_{\mathrm{v}} = [\mathrm{v}(\hat{\mathbf{H}}[0])^{\mathrm{T}}, \ldots, \mathrm{v}(\hat{\mathbf{H}}[K-1])^{\mathrm{T}}]^{\mathrm{T}}$, $[\Phi]_{:,p} = \psi(\tau_p) \otimes (\phi^{*}(\theta_{tx,p}) \otimes \phi(\theta_{rx,p}))$ and $[\gamma]_p = \gamma_p$. This way, we have $\hat{\mathbf{h}}_{\mathrm{v}}$ as the vector containing all measurement information, Φ as all path contributions and γ as their complex gains. Note that only Φ has a dependency on the path parameters and each column depends only on one path, while γ behaves as a weight vector for the different path contributions. Converting the formulation from $\min \|\hat{\mathbf{H}}[k] - \mathbf{H}[k]\|$ to the vectorized version of the problem $\min \|\hat{\mathbf{h}}_{\mathrm{v}} - \Phi\gamma\|^2$ makes it easy to evaluate the minimization. Now let $\Phi^{\perp}$ be the orthonormalization by Gram-Schmidt of Φ and $\mathbf{A}$ the invertible square matrix such that $\Phi = \Phi^{\perp}\mathbf{A}$ to simplify the incoming equations. Then

$$\begin{aligned} &\min \|\hat{\mathbf{h}}_{\mathrm{v}} - \Phi\gamma\|^2 = \min \|\hat{\mathbf{h}}_{\mathrm{v}}\|^2 + \|\mathbf{A}\gamma\|^2 - \mathcal{R}(\hat{\mathbf{h}}_{\mathrm{v}}^{\mathrm{H}}\Phi^{\perp}\mathbf{A}\gamma) \\ &= \min \|\hat{\mathbf{h}}_{\mathrm{v}}\|^2 - \|(\Phi^{\perp})^{\mathrm{H}}\hat{\mathbf{h}}_{\mathrm{v}}\|^2 + \|(\Phi^{\perp})^{\mathrm{H}}\hat{\mathbf{h}}_{\mathrm{v}} - \mathbf{A}\gamma\|^2 \\ &= \min \|\hat{\mathbf{h}}_{\mathrm{v}}\|^2 - \|(\Phi^{\perp})^{\mathrm{H}}\hat{\mathbf{h}}_{\mathrm{v}}\|^2 \end{aligned} \quad (13)$$

This formula is very fast to evaluate, making it amenable to a Nelder-Mead search over the path parameters $\theta_{rx,p}, \theta_{tx,p}, \tau_p$ in expression (13). Then, γ is recomputed as the linear MMSE solution using the refined parameters.

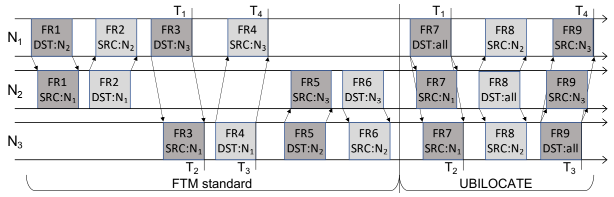


Figure 2: Standard FTM (left) sends dedicated messages per pair of nodes and UbiLocate (right) broadcasts a single frame per node for ranging with all other nodes.

The direct path p_{dp} corresponds to the index p with the smallest τ_p. To avoid spurious results, we add a power regularization term

$$p_{\mathrm{dp}} = \min_{p} \tau_p - 0.0001\frac{\gamma_p}{\max_{p'} \gamma_{p'}} \; . \quad (14)$$

Finally, the estimated AoA at the AP is given by

$$\hat{\theta} = \theta_{rx,p_{dp}} \; . \quad (15)$$

2.3 Ranging

Accurate ToF information is crucial for ranging and thus for localization. Unfortunately, locating a target node with multiple APs leads to several problems that must be addressed to achieve good performance. Each AP is running its own clock source, and since the different clocks are not synchronized, it is not possible to correct ranging estimates simply by post-processing the collected CSI data. Obtaining accurate ToF estimates between each AP and the client requires multiple packet exchanges with timestamps, as in the FTM protocol. While this protocol was standardized several years ago [23], the majority of current WiFi devices do not support it (including the ones we instrument for this work). At the same time, FTM measurements of devices that do support it show suboptimal performance in multipath-rich environments. We thus introduce in our framework the first implementation of an FTM-like protocol that obtains accurate ranging information on off-the-shelf 802.11ac devices that support CSI extraction.

In Fig. 2, we highlight the differences between the standard FTM and our implementation by showing how ranging is performed with three nodes $N_n, n \in \{1, 2, 3\}$ with time on the x-axis. Standard FTM uses unicast frame-ack exchanges, whereas UbiLocate broadcasts frames asynchronously to all other nodes. This significantly reduces the number of frames for ranging with multiple nodes.

For the FTM frames 1-6, we indicate the destination (at transmitter) and the source (at receiver) and the corresponding times. For instance, frame 3 ($FR3$) is transmitted at time T_1 by node N_1

(the initiator) to node N_3 (the responder) that receives it at time T_2. Afterwards, N_3 responds by transmitting frame $FR4$ to node N_1 at time T_3, which is received at node N_1 at time T_4. In addition to specifying the frames carrying the timestamps, FTM defines a mechanism to collect timestamps measured by the responder at the initiator. Then, FTM uses T_1, T_2, T_3 and T_4 to evaluate the Round-Trip Time (RTT) and thus the distance $\hat{d}$:

$$\begin{aligned} \text{RTT} &= (T_4 - T_1) - (T_3 - T_2) \\ \hat{d} &= (RTT/2) \cdot c \end{aligned} \quad , \tag{16}$$

where c is the speed of light. By using both transmit and receive times it is possible to remove the reaction time uncertainty, i.e., the delay between frames $FR3$ and $FR4$. The procedure can be repeated multiple times to average results and obtain a more accurate estimate [23]. For FTM, $N(N-1) = 6$ frames are required to compute the $N = 3$ distances, resulting in a quadratic overhead.

Instead, UbiLocate requires only $N = 3$ broadcast frames as shown in the right part of the figure. A frame includes the $N - 1$ timestamps when the last frame from each of the other nodes was received, as well as the transmit timestamp for the frame itself. These frames are transmitted asynchronously by each node, and are opportunistically reused by other nodes, resulting in a linear overhead. This also removes the need for a dedicated collection mechanism. The three frames $FR7$-$FR9$ are used to compute the three distances. We first use $FR7$ in place of $FR3$, and we call T_1 the time when $FR7$ is transmitted by node N_1, and T_2 the time when it is received at node N_3. We then use $FR9$ in place of $FR4$, sent and received at T_3 and T_4, respectively. As $FR9$ embeds T_2 and T_3 (among other timestamps), upon receiving it, node N_1 can use the same equation above to determine the distance. We can reuse $FR7$ together with $FR8$ to evaluate the distance between nodes N_1 and N_2. Similarly, we can reuse $FR9$ with $FR8$ to estimate the distance between N_2 and N_3.

2.4 Localization

With the information discussed previously, AP a can estimate the location $\hat{\mathbf{y}}_a$ of the target device in Cartesian coordinates using

$$\hat{\mathbf{y}}_a = \mathbf{x}_a + \hat{d}_a \begin{bmatrix} \cos \hat{\theta}_a \\ \sin \hat{\theta}_a \end{bmatrix} , \tag{17}$$

where $\mathbf{x}_a$ is the (known) position of AP a, $\hat{d}_a$ is the estimated distance of the target device from the AP, and $\hat{\theta}_a$ is the AoA estimated at the AP.

Since Eq. (17) holds for any AP, we have a system of A such equations, where A is the number of APs. However, not all APs provide equally useful location information and a simple strategy that averages all estimated positions $\hat{\mathbf{y}}_a$ with equal weights is suboptimal. To identify and filter out unreliable estimates, UbiLocate uses a metric that measures the dominance of multipath components with respect to the direct path in the received signal. The specific metric used by our system is the *mean excess delay* [39], given by the weighted average of the delays of every single multipath component with respect to the direct path, with relative path power as the weight. More precisely, assuming that we can discriminate $P > 1$ different paths, the mean excess delay $\tau_{m,a}$ for AP a is:

$$\tau_{m,a} = \frac{\sum_{p=0}^{P-1} \left\|\gamma_p\right\|^2 (\tau_p - \tau_0)}{\sum_{p=0}^{P-1} \left\|\gamma_p\right\|^2} , \tag{18}$$

where γ_p and τ_p are the complex attenuation and ToF of path p, respectively, and τ_0 is the ToF of the first received path.

If the contribution of the multipath components is small compared to the direct path, $\tau_{m,a}$ will tend to 0, whereas larger values of $\tau_{m,a}$ indicate stronger multipath. Hence, a large mean excess delay is an indication that the position estimate $\hat{\mathbf{y}}_a$ of AP a might be less reliable. UbiLocate uses a threshold τ_{th} and discards the estimates whose mean excess delay exceeds τ_{th}. Since this metric largely depends on the geometry of the scenario, obstacles and many other factors, fixing an absolute threshold for this metric to remove unreliable APs could lead to also removing useful APs. To address this, for each measurement point UbiLocate applies a dynamic threshold relative to the AP with the lowest mean excess delay, τ_{lw}. Specifically, τ_{th} is equal to two times τ_{lw}.

We denote by A' the set of APs for which the mean excess delay $\tau_{m,a}$ is below τ_{th}. Then, given $|A'|$ estimates along with their corresponding mean excess delay $\tau_{m,a}$, UbiLocate computes the final position of the target node with a weighted centroid approach:

$$\hat{\mathbf{y}} = \frac{\sum_{a=0}^{|A'|-1} \hat{\mathbf{y}}_a \cdot (\tau_{m,a})^{-1}}{\sum_{a=0}^{|A'|-1} (\tau_{m,a})^{-1}} . \tag{19}$$

This way, estimates with a small mean excess delay receive a higher weight. UbiLocate thus discards very unreliable estimates and gives higher importance to estimates from the most reliable APs. Furthermore, UbiLocate addresses the following issues:

Extreme angles. Extreme angles are defined as angles below -75° and above +75°. For these cases, UbiLocate's AoA estimator may takes the opposite solution (i.e., UbiLocate estimates -75° when the correct AoA is +75°), due to the fact that the relative phase differences become close when the angles approach ±90° and the system is affected by noise. To overcome this issue, UbiLocate considers both possible AoA values and computes the two resulting positions. UbiLocate then chooses the one that has the minimum distance to the position estimates from other APs.

Disagreement between position estimates. When UbiLocate combines estimates from very few APs, a single outlier may lead to large location errors. UbiLocate handles the specific case when only two APs are available for the localization. If the distance between location estimates is high, this indicates that the estimate of one of the AP is an outlier and thus combining the two estimates may degrade the location accuracy. UbiLocate then takes the AP with the lowest ToF estimate for the localization when the distance between the two position estimates exceeds 4 m.

3 IMPLEMENTATION

We build UbiLocate on the Nexmon project that provides a first step towards CSI extraction from several chipsets developed by Broadcom [17]. We largely improve over this prior work to consistently extract *accurate and reliable* CSI, implement features that make CSI extraction more flexible, and add support for timestamping both received and transmitted frames with very high accuracy.

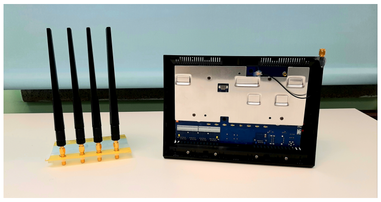
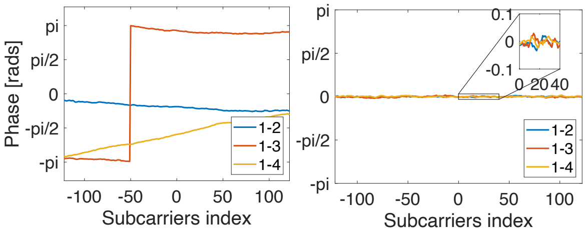


Figure 3: Asus 802.11ac router with custom antenna array.



(a) Before calibration **(b) After calibration**

Figure 4: Phase differences for the antennas pairs.

For our implementation we select the Asus AC2900 RT-AC86U router since it supports 80 MHz 802.11ac with up to four spatial streams in a 4x4 MIMO configuration. The firmware that we developed replaces the standard one by Broadcom and can capture the CSI matrix for frames with configurable MAC addresses. In addition, it recognizes the type of frame, including the spectral width and the spatial configuration, and collects the CSI matrix accordingly. Since the router exposes only three antenna SMA plugs externally, we remove the front panel to access the fourth internal UFL connector and attach a custom antenna array handler to the four antennas of the router as shown in Fig. 3.

We now discuss how we (1) validate the collected CSI and process it to estimate AoA and AoD, (2) provide the timestamping features, and (3) finally implement the enhanced FTM procedure.

3.1 Extracting accurate CSI

A range of preliminary measurements with our hardware platform reveal that calibration is needed to remove hardware imperfections that would otherwise affect the CSI and render it too unreliable for localization tasks. Specifically, we address the following problems:

Phase offset between antennas. While all the RF chains share the same sampling clock and reference signal (to tune to a given frequency), an unpredictable phase offset between each pair of antennas appears every time the system is tuned to a new WiFi channel. As a result, the measured phase delay may not correspond to the one measured by the AoA or AoD algorithms. This unpredictable phase offset then remains flat over time.

Echos. We observe that the router generates echos from a received signal, i.e., the signal is repeated in the time domain. The time distribution of such echos is fixed and they never change.

We devise a procedure to remove these two imperfections which consists of a calibration experiment that has to be repeated every time we configure the equipment. During the setup, we capture a full CSI matrix with the four antennas connected to the same single-chain transmitter. This can be easily achieved by connecting the output ports of a 4-way splitter to four short cables that are also used to connect the four external antennas during the localization experiment later. The splitter ensures that all the signals arrive in phase, and hence the AoA of the transmitted frame is at 0 degrees. Thus, all phase offsets measured between the receive chains depend only on the (random) configuration of the local oscillator. The rationale behind this experiment is that the full CSI matrix captured during the calibration phase represents a reference signal that can be used for *correcting* the CSI vectors captured afterwards. To this end, we perform the (element-wise) Hadamard division ⊘ between new CSI vectors and the reference one:

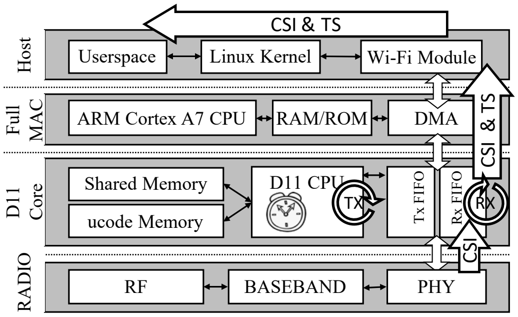


Figure 5: Enhanced CSI extraction platform with the modifications to collect ToF.

$$calibrated\ CSI = CSI \oslash reference\ CSI$$

Fig. 4 shows the phase offset between different pairs of antennas before (Fig. 4a) and after applying the calibration (Fig. 4b). The residual phase offset appears to be minor Gaussian noise, confirming that this procedure reliably removes the phase imperfections due to the hardware configuration.

3.2 Extracting timestamps

Implementing a ToF measurement procedure similar to standard FTM requires accurate time-stamping capabilities in both the transmit and receive directions, and the majority of the Wi-Fi chipsets, including the one in the chosen platform, simply lack them. We hence used the Nexmon firmware patching framework [36] to add these capabilities to the platform, following a similar approach to the one in [42]. The main modifications involve the software that runs in the D11 CPU, a microcontroller that manages all time-critical operations such as channel access, beaconing, generation of reply frames, etc. This software consists of a single main loop that i) can neither be interrupted by internal IRQs nor by the upper layer ARM Wi-Fi core; and ii) branches into secondary functions when the hardware reports conditions that require additional work. In particular, we modify two functions that belong to the reception and transmit paths, respectively.

The first function is invoked when a preamble is detected and performs multiple checks on the first bytes of the incoming frame to decide how to process it. The CSI extraction patch already adds a single instruction loop that spins until the frame is completely received and then it pushes both the CSI data and the frame to the host. We further customize this loop by adding instructions to sample the value of the high-frequency clock of the system whenever the frame ends. We represent this modification in Fig. 5 with the spinning wheel on the right data pipe (reception path). As shown, after the CSI data is retrieved from the PHY at the bottom, the RX timestamp travels up to the application that runs in user space and collects all the data.

The second function is invoked when the hardware verifies that all the conditions required for transmitting a frame are satisfied (i.e., the channel was idle long enough for the backoff counter to reach zero, no more energy is detected in the channel, no other operations are pending, etc). When this happens, the hardware is already transmitting the frame preamble. The function can then customize the transmission and monitor it until it terminates. We add two modifications here. The first writes the timestamps overheard during the previous transmissions as well as the device's own timestamp into the frame. The second consists of a new loop that waits until the end of the transmission and is represented in the figure with the spinning wheel on the left data pipe (transmit path). With this code we capture the transmit timestamp.

We obtain both timestamps by sampling the high-frequency clock that runs at the speed of the D11 CPU. For the chosen platform this corresponds to 192.6 MHz, and thus the receive timestamp has an uncertainty of 1.56 m when in perfect LOS without any multipath. In the presence of multipath, the timestamps are affected by the arrival of all the paths which leads to a bias. However, UbiLocate can extract an accurate time of arrival of the first path using the CSI data of the timestamp packet. Specifically, every path that arrives at the receiver introduces a phase rotation in the CSI of the subcarriers. By decomposing the channel in time domain, UbiLocate can eliminate the multipath bias and thus estimate a much more accurate time of arrival for the direct path.

3.3 Implementation of the FTM procedure

To evaluate the ToF between two nodes, we use the same set of equations as in Eq. (16) for standard FTM. We consider two frames traveling in opposite directions—relatively close in time—and we combine the four corresponding timestamps of two transmissions and two receptions. However, different from FTM, the frames do not belong to a specific *frame-ack* exchange. Instead, they are transmitted by the nodes asynchronously. In our experiments we transmit such frames frequently, so that frames from each AP are close in time to that of the client, but other strategies are possible: i.e., the client might initiate the procedure by transmitting a train of frames and all APs can schedule the same number of transmissions as soon as they receive the first frame from the client. We leave such modifications for future work.

To generate the ToF-related frames we use the injection capabilities available in the Nexmon CSI framework. We implement a user-space application that uses a PID controller to generate frames at a configurable rate, e.g., one frame every 4 ms. We also modify the D11 code to keep the same pacing at the access layer. This solution is key to avoid any DMA-related delay and ensure that at any moment in time there are enough "close" frames transmitted by all nodes, so that ToF estimation and thus ranging can be done. We further implement a back-pressure mechanism to avoid saturating the DMA memory when the queue holding injected frames starts to build up.

Another deviation from standard FTM is the fact that our implementation has no initiator and responder. For this reason, we cannot store timestamps at the responder and collect them later from the initiator. Hence, we modify the D11 code to store transmission timestamps directly inside the frame. To this end, we additionally modify one of the two functions described in the previous section. With this modification, we obtain all the necessary information for running the ranging procedure by capturing traffic traces at all nodes. In these traces we have the frames, corresponding reception timestamps and CSI data, and the transmission timestamps generated by the sender.

We finally describe the overall procedure for evaluating ToF between a pair of nodes N_1 and N_2. We start by processing the traces captured at each node, containing the frames received from the other one. We then align the clock of node N_1 to that of N_2. We extract from all frames collected by N_1 the reception timestamp (at N_1) and the transmission timestamp (generated by N_2). We then apply linear regression to remove the clock skew between the two nodes, adjusting both reception and transmission timestamps. For each frame transmitted by N_1 we associate the closest frame in time received from N_2 and we apply Eq. (16) to the four-tuple of timestamps, yielding a ToF estimate. Since each AP transmits these broadcast ToF packets asynchronously, collisions are avoided by means of the standard DCF channel access mechanism of IEEE 802.11. However, we observe a variability in the ToF estimates due to systematic delays introduced by WiFi packet processing similar to plain FTM [23]. These delay differences follow a Gaussian distribution which is centered approximately at the correct ToF value. We can thus remove this uncertainty by averaging over a certain number of estimates to compute a smoother ToF. We observe that 20 around estimates for good accuracy. As UbiLocate sends broadcast ToF packets every 4 ms, on average 80 ms are required to compute a smoothed ToF estimate. We also tested UbiLocate's ToF with different levels of background traffic and do not observe any degradation in raw ToF estimation accuracy. With fully backlogged background traffic, which corresponds background traffic rate of 500 Mbps, UbiLocate gets around 40 ToF estimates per second which results in smoothing ToF estimates over 500 ms.

4 EXPERIMENTAL EVALUATION

We now evaluate the location accuracy of UbiLocate in a realistic setup and compare it to several state-of-the-art location systems.

4.1 Testbed setups

To provide a comprehensive performance comparison of UbiLocate and state-of-the-art location schemes, we test three different deployments. The first is a simple scenario with high AP density, where all APs have a LOS path to the station. This corresponds to the benign conditions under which location systems are usually tested. Second, we evaluate a medium density scenario where the station usually sees several APs with a mix of LOS and NLOS conditions, which tests the systems under adverse conditions. Finally, we move to a much larger and more sparse environment where usually only two or three APs are available at a time. This corresponds to the most common real-world deployments that are optimized for WiFi coverage, rather than localization performance.

High density testbed The high density environment comprises four APs, each one placed in the corner of a room of size 85 m^2, as

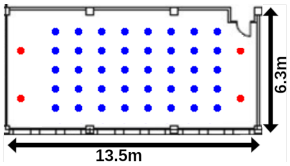


Figure 6: High density testbed.

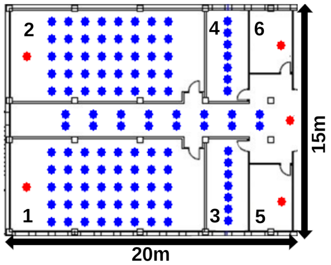


Figure 7: Medium density testbed.

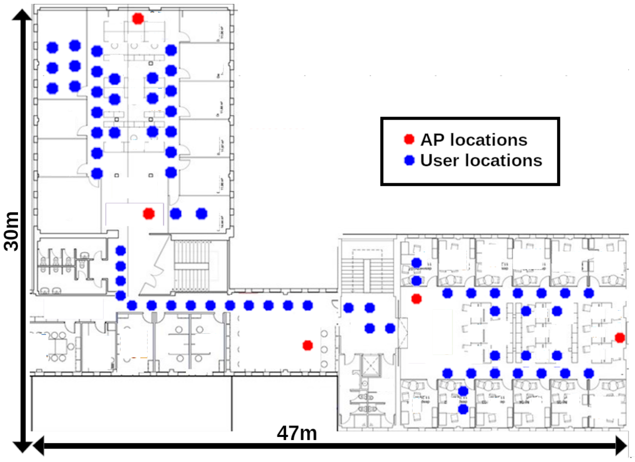


Figure 8: Low density testbed.

shown in Fig. 6. The deployment has an AP density of 1/21.25 m^2. We further ensure that each AP has a clear direct path to the station.

Medium density testbed The map of this testbed is depicted in Fig. 7. The area is approximately 300 m^2, contains 5 APs, and has an AP density of 1/60m^2. It has seven distinct areas: six rooms, not all of which contain an AP, and one central corridor.

We consider 110 measurement points located in rooms 1, 2, 3, 4, and in the corridor, shown as blue dots in Fig. 7. The five APs used to localize the target are shown as red dots and they are placed in rooms 1, 2, 5 and 6, and in the central corridor. With this deployment we ensure that: 1) the majority of target locations are in LOS with exactly one of the APs; 2) some of the target locations—the ones in rooms 3 and 4—are not in LOS with any of the APs; and 3) two APs—namely the ones in rooms 5 and 6—do not have a clear LOS to any of the target locations. Finally, for this deployment we also test different pure NLOS scenarios, where for each measurement point *we specifically remove the only AP that does provide LOS*, if any.

Low density testbed. This testbed pushes the location systems to their limit with a much more sparse deployment. This is in fact the most realistic scenario, with an AP density close to that of the actual production WiFi deployment in this office building. It comprises two wings of a building and one central area that connects them as shown in Fig. 8, with a total area of 578 m^2 and an AP density of 1/115 m^2. Each wing contains an open plan area with desks, as well as closed offices on either side. The dividing walls, furniture and the people moving around (measurements were taken during daytime) create a rich multipath environment and many areas without LOS. The scenario comprises 70 measurement points, and each point is usually covered by only two (in the best case by three) APs, whereas in the other scenarios most measurement points are covered by all APs. As a result, in this setting it is crucial to properly merge the location information from the few APs within range.

In all the considered scenarios, the APs are working in monitor mode, extracting one CSI matrix for every received frame. For ranging, each AP exchanges 802.11 frames with the target device following the procedure described in Section 2.3. A central controller connected to the APs via Ethernet gathers all the data to compute the AoA and the distance for every AP as described in the previous sections. Finally, the algorithm presented in Section 2.4 is executed on the controller to estimate the position of the device.

4.2 Comparison with other systems

We benchmark the performance of UbiLocate against the following three state-of-the-art indoor location systems.

Spotfi [27] is a WiFi location system which combines angle measurements from several APs to determine the device position. Spotfi computes AoA and path delay using a 2-dimensional MUSIC algorithm with spatial smoothing for accurate AoA estimates.

FUSIC [26] is based on ToF measurements to determine the device position. It relies on FTM ranging and uses the 1-dimensional MUSIC algorithm to reduce multipath effects.

SPRING [40] combines both AoA and ranging information to provide single AP localization. It uses the MUSIC algorithm for AoA and FTM for the distance. While SPRING was originally designed to work with only one single AP, in our experiments we average the estimates of all of the APs to provide a better position estimate.

We compare these systems against two different versions of UbiLocate, one that estimates the position using AoA, AoD and ToF, and a more basic version which only takes into account AoA and ToF. To distinguish different versions of UbiLocate and indicate the main features used by each system, we apply the following labeling scheme: letters A, D, and T identify a system using AoA, AoD and ToF information, respectively. For example, UbiLocate [AT] is used to refer to the basic implementation of UbiLocate that only uses AoA and ToF, whereas UbiLocate [ADT] refers to the full version that uses all information.

Before we delve into the overall performance of our location system, we first study the performance of the individual components of UbiLocate, i.e., the angle and ranging estimates, in isolation. In particular, the UbiLocate AoA estimator is compared against the 1D MUSIC AoA estimator and against the one used in Spotfi. We also compare UbiLocate's ranging subsystem to vanilla FTM and to the improved FTM-based ranging proposed in FUSIC.

While the majority of the systems described above were designed for and evaluated with the older IEEE 802.11n WiFi standard, we compare them against UbiLocate both for IEEE 802.11n and IEEE 802.11ac. SPRING also originally uses 80 MHz frames but is based on a proprietary Quantenna platform. In addition to the improved hardware capabilities of recent devices, the new 802.11ac standard supports 4x4 MIMO and up to 80 MHz of bandwidth. These features help significantly to resolve multipath effects. For reference, 802.11n systems are limited to 3x3 MIMO and up to 40 MHz of bandwidth.

4.3 High density scenario

The aim of this experiment is to localization performance of UbiLocate and the rest of the system in a benign multipath environment and to compare it against the state-of-the-art approaches in an environment similar to the one they have been designed for. This also allows to validate that the performance of the state-of-the-art algorithms matches the results reported in the respective papers. To this end, we first consider a simple LOS environment, as indicated in Fig. 6. It comprises four APs, with one AP placed in each of the corners of room 1. We use 40 location measurement points in an area of 85 m^2. In addition, to show how the improved capabilities offered by the 802.11ac standard impacts location accuracy, we evaluate all methods both for IEEE 802.11ac as well as IEEE 802.11n configurations. As shown in Fig. 9a, when using 802.11ac frames, UbiLocate achieves sub-meter localization accuracy for all the target points and a median error of 30 cm for [ADT] and 40 cm for [AT]. On the other hand, Spotfi and SPRING have a median error of 60 cm and 70 cm and a maximum error of 2.3 m and 3.6 m. FUSIC has the worst performance with a median error of 1.7 m. As expected, we observe that moving from 802.11ac to 802.11n leads to a performance degradation for all of the systems. For example, UbiLocate's median error increases from 30 cm to 60 cm for [ADT] and from 40 cm to 85 cm for [AT], respectively. Since the relative performance of the approaches does not differ substantially between 802.11n and 802.11ac, for the remaining experiments we only compare the performance of all systems with an 802.11ac configuration. It is worth highlighting that UbiLocate [ADT] is the only system that achieves sub-meter location accuracy for all measurement points in the high density testbed, making it an excellent fit for localization-based services that are sensitive to errors.

4.4 Medium density scenario

After evaluating UbiLocate in a simple LOS and dense environment, we now study how the individual features AoA and ToF behave in more complex settings with LOS and NLOS. Afterwards, we will show how these features translate into localization performance.

4.4.1 *Analysis of individual features.*

We test the performance of the different angle and ranging algorithms in the large deployment scenario shown in Fig. 7.

AoA. As shown in Fig. 10a, under LOS conditions UbiLocate achieves an excellent median error of 1 degree and 3 degrees for the [ADT] and [AT] versions respectively and a maximum error of 20 and 50 degrees, while Spotfi and MUSIC both have a significantly higher median error of 3.7 and 4.4 degrees and a maximum error of 65 and 55 degrees, respectively. For the measurement points that have NLOS, UbiLocate achieves a median error of 6 degrees while that of the other two approaches is above 10 degrees. A striking difference can be seen for the maximum error achieved 90% of the times: while UbiLocate has an error of at most 20 and 25 degrees for [ADT] and [AT] which is still partly usable, both Spotfi and MUSIC errors reach 40 degrees, which is indicative of significant outliers. We attribute our improvements to the Nelder-Mead search algorithm described in Section 2.2, which iteratively refines our estimates of the AoA by removing the effects of undesired multipath components. Note that the graph includes the raw estimates for all APs within range, whereas for the actual localization the AP estimates are weighted and filtered (i.e, not all estimates are used).

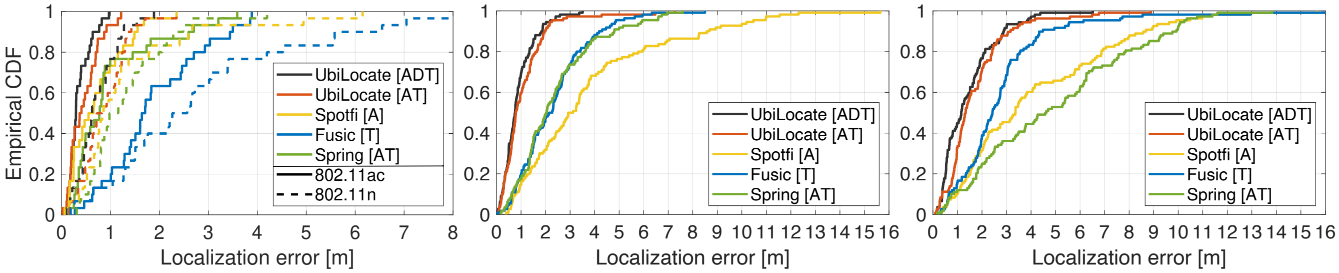
Ranging. Results for the ranging subsystem are shown in Fig. 10b. We verify that UbiLocate can perform ranging more accurately than FTM and FUSIC. Specifically, we measure a median error of 43 cm for UbiLocate (90% of the times below 1.3 m in LOS conditions), while FUSIC and FTM both achieve similar performance, with 0.8 m and 2 m for 50% and 90% of the cases, respectively. Also for NLOS conditions, UbiLocate ranging accuracy outperforms the other methods, with a median error of 1.1 m, while FUSIC and FTM have errors of 1.6 and 1.9 m. The key features of our system that enable this good performance are the accurate timestamping capabilities we added to the firmware of the devices (see Section 3).

4.4.2 *General localization.*

We now evaluate the overall localization accuracy of the different approaches in the medium density scenario. Specifically, we demonstrate the robustness of UbiLocate against NLOS and how UbiLocate deals with potentially contradictory location information from different APs in two spatial contexts: the LOS + NLOS deployment and a special case of only NLOS.

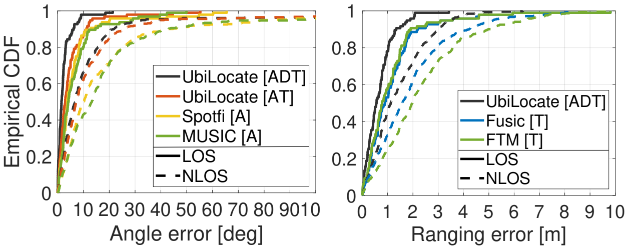
LOS + NLOS. This deployment scenario is shown in Fig. 7 and comprises five APs and 110 measurement points. In the best case, there is only one AP in LOS while the other APs are in NLOS. Thus, it is critical to exploit primarily the information extracted from this AP as it provides the most accurate location information, while minimizing the contribution of unreliable information from some of the NLOS APs. The results are shown in Fig. 9b. Clearly, UbiLocate achieves a significant median accuracy improvement of around a factor of 2 compared to state-of-the-art algorithms for both UbiLocate versions. Specifically, UbiLocate's median error is 0.75 m while for SPRING, FUSIC and Spotfi it is 2 m, 2.1 m and 3 m, respectively. Furthermore, the maximum error of UbiLocate is 3.5 m and 6 m for the [ADT] and [AT] versions, whereas the maximum errors of SPRING, FUSIC, and Spotfi are much higher at 7.5 m, 9 m and 15.5 m, rendering them unsuitable for many indoor location based services. While the median errors of [ADT] and [AT] are similar, the additional AoD information used in [ADT] significantly reduces the maximum error compared to [AT]. This superior performance is not only related to the more accurate AoA and ToF subsystems of UbiLocate, but also the particular localization strategy that identifies the most reliable APs and weighs their contributions based on their estimated quality. For completeness, we also tested UbiLocate [A], i.e., a pure AoA system which runs only on the APs without any station-side modifications. It achieves a median error of 1.2 m.

NLOS-only Finally, we evaluate the systems in a setting, where we force all measurement points to be in full NLOS. To this end, we remove the respective APs that does provide LOS information, if



(a) High density testbed. **(b) Medium density LOS+NLOS.** **(c) Medium density NLOS *only***

Figure 9: Localization performance of UbiLocate compared to state-of-the-art systems.



(a) AoA performance **(b) ToF ranging performance**

Figure 10: Empirical CDF for AoA and ToF error for all APs, and for LOS (solid lines) and NLOS (dashed lines).

any, i.e., for the measurement points in room 1 we remove the AP in room 1 and test the localization performance in that room with the remaining APs. This process is repeated for all other rooms as well. While this scenario is extreme and LOS will be available for at least some of the locations in a regular deployment, it gives a good indication of the expected performance when additional moving obstacles (such as persons) in the rooms obstruct and distort the only available LOS path.

The results are shown in Fig. 9c. There is a small performance degradation in localization accuracy, but UbiLocate still provides meter-level median accuracy with an error of 1.1 m and 1.2 m for the [ADT] and [AT], respectively. In contrast, the median errors for SPRING, FUSIC, and Spotfi are 4 m, 2.6 m and 3.5 m, respectively, around a factor of 2 to 3 worse than UbiLocate.[2] Finally, UbiLocate [A] has a median error of 2.2 m. This good overall performance of UbiLocate in NLOS indicates that path information from APs under obstructed LOS is valuable, if the paths can be resolved accurately.

4.5 Low density scenario

Compared to the previous scenarios, the low density scenario shown in Fig. 8 is much more sparse and for half of the points the client only sees two APs. This AP density is realistic for real-world deployments, where coverage depends very much on the geometry of the deployment. For the points with only two APs, Fusic and Spotfi cannot determine a location since they both need at least 3 APs within range to locate the user. In addition, the office furniture and the people moving around produce a rich multipath environment and dynamic channel conditions. We again first show the performance of the individual features and then the general localization performance.

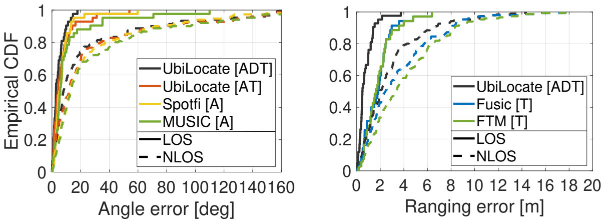
4.5.1 Individual features. We compare the AoA and ToF estimation in this challenging case to the previous scenarios. Again, the graphs include the raw estimates for all APs, whereas for the actual localization, UbiLocate filters out some of the outliers.

AoA. The AoA results are shown in Fig. 11a with UbiLocate obtaining a median error of 2.7 and 8.5 degrees for LOS and NLOS settings for the [ADT] version and 4.3 and 12 degrees for [AT]. Spotfi and MUSIC have similar performance and achieve a median error of 5.6 and 6.2 degrees for LOS and 12 and 13 degrees for NLOS, respectively. This degradation in the LOS and NLOS performance is caused by the larger distances and the rich multipath compared to the medium and high density scenarios.

Ranging. As shown in Fig. 11b, UbiLocate has an excellent median error of 0.5 m in LOS while for NLOS it achieves 2 m. FUSIC and FTM have the same median error of 1.8 m for LOS and 2.8 and 3.4 m in NLOS, respectively. UbiLocate has the lowest maximum error of 12 m, while Fusic and FTM errors reach 18 m.

4.5.2 General localization. The localization errors can be found in Fig. 12. Since Fusic and Spotfi cannot be applied for all of the measurement points, their CDF curves do not reach 1, whereas SPRING does since it works with just a single AP. As in the other evaluation, the [ADT] and [AD] versions of UbiLocate have similar performance with a median error of 1 m, while Spring achieves a 4 m error. Regarding the highest errors, UbiLocate [ADT] and [AD] reach 10 m while SPRING has up to 24 m. As expected, the low AP density and the rich multipath environment produce larger outliers compared to the medium density NLOS case. Similar to Spotfi, UbiLocate [A] only works in 55% of the points and achieves a median error of 12.4 m. The few large outliers with UbiLocate come from extreme points in far corners of the building that have large angles to the one or two APs within range under NLOS. In those cases, achieving better accuracy is only possible by deploying another AP. It is worth highlighting that UbiLocate generally deals very well even with such a sparse scenario with a complex channel environment, a low error in most cases. While the performance of UbiLocate is similar for the low density and NLOS-only scenarios, the reasons are different. The NLOS-only scenario is more dense with four APs in coverage, but none of them have a clear LOS, whereas the low density scenario has only 2 or 3 APs to localize, but there are cases with a clear LOS. These different effects happen to compensate each other in the specific scenarios under study.

[2] Note that in [27] a higher NLOS accuracy for Spotfi was reported. However, their NLOS deployment typically has around two APs with LOS per measurement point, whereas we consider as true NLOS only points for which *none* of the APs are in LOS.



(a) AoA performance

(b) ToF ranging performance

Figure 11: Empirical CDF for AoA and ToF error of UbiLocate compared to state-of-the-art systems for the low density scenario for all APs.

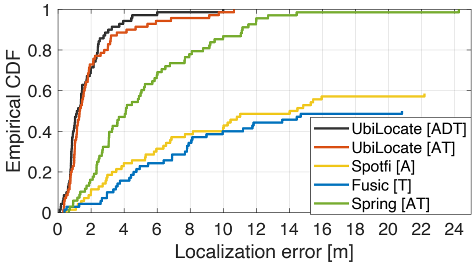


Figure 12: Localization performance of UbiLocate and state-of-the-art systems for the low density scenario.

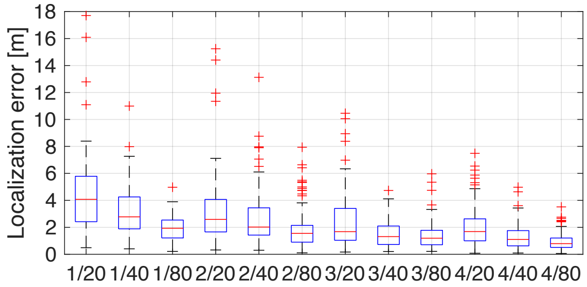


Figure 13: UbiLocate location accuracy of different configurations of (number of antennas/bandwidth)

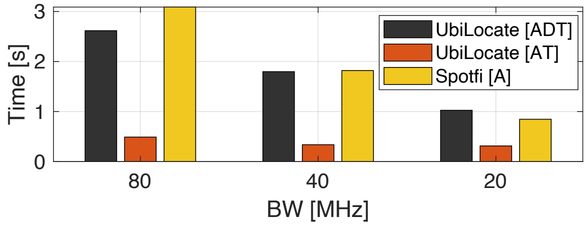


Figure 14: Time complexity for UbiLocate and SpotFi.

4.6 Additional Considerations

Impact of MIMO and bandwidth. All the experiments described up to this point are performed with the same 4x4 MIMO configuration with 80 MHz bandwidth. However, in principle UbiLocate can work with any hardware configuration. To characterize the localization performance for devices ranging from low-end to high-end hardware complexity, we evaluate UbiLocate for the following bandwidth and MIMO configurations in the medium density testbed (LOS + NLOS). We consider three bandwidth combinations (20/40/80 MHz) and four antenna configurations (1x1/2x2/3x3/4x4), resulting in 12 configurations overall.

The results of this evaluation are illustrated in the box plot in Fig. 13. Let us first consider the 4x4 MIMO configuration. As expected, the median error rises from 0.7 m when working with a bandwidth of 80 MHz to 1 m and 1.6 m when reducing the bandwidth to 40 MHz and 20 MHz, respectively. The worst performance is obtained with the single-antenna system and 20 MHz of bandwidth, with a median error of 4 m and a maximum error of 18 m. For comparison, the median error with one antenna and 80 MHz is only 1.8 m. Finally, the importance of AoA information can be seen from the sudden drop in the median localization error when moving from the 1x1 to the 2x2 MIMO configurations. However, decimeter-level median accuracy can only be achieved with 3x3 and 4x4 MIMO and 80 MHz, or with 4x4 MIMO and 40 MHz bandwidth, indicating that 802.11n hardware capabilities with 3x3 MIMO and 40 MHz are insufficient to achieve this very high accuracy.

Time complexity. Time complexity plays a crucial role especially in real-time processing. The dimensionality of the parameters and their granularity considerably affect the time complexity of the optimization algorithm. To evaluate it, we run the two versions of UbiLocate and Spotfi using the traces collected during the localization evaluation. UbiLocate applies Nelder-Mead search for the five most significant paths. The server used for this evaluation is an Intel(R) Core(TM) i7-6800K CPU with 3.40GHz and 16 GB of RAM running MATLAB 2019a. The results are illustrated in Fig. 14. We observe that for 80 MHz, UbiLocate [ADT], which estimates the three path parameters, is faster than Spotfi which only estimates two. UbiLocate [AT], which estimates two parameters, has an execution time of half second and reduces the time complexity by 85% compared to Spotfi. The significant difference in time complexity between [AT] and [ADT] comes from the combinatorial complexity with respect to the number of path parameters to be estimated. This is exacerbated by the high number of spatial streams of the MIMO system, since the channel complexity increases with the possible transmit/receiver antenna pairs.[3] While the time complexity of UbiLocate [AT] is significantly lower than that of [ADT], adding AoD information does reduce the maximum localization error, as discussed in Section 4.4. In addition, we observe that when we reduce the bandwidth by half, the time required to run the system is also approximately reduced by half. Our implementation uses unoptimized Matlab code using the predefined Matlab functions. We expect an improvement of the time complexity by a factor of 5-10 with an optimized implementation in native C.

5 RELATED WORK

Wireless localization is a very hot topic and has been widely studied both from theoretical and practical perspectives [4]. Below, we survey the most important approaches in the research area.

Path parameter estimators. Extracting the path parameters of the radio-frequency signal has been largely analyzed for positioning purposes, especially in the field of AoA estimation. Many classical algorithms such as MUSIC [46] and ESPRIT [43] are currently used but they do not well resolve AoAs of highly correlated signals [49]. Spatial smoothing techniques allow to decorrelate these signals [38,

[3] Note that if Spotfi were to consider AoD together with AoA and the path delays, its time complexity would increase over-proportionally, since the Nelder-Mead search we use deals with the complexity increase more efficiently than 3D MUSIC.

52] and provide better performance. Compressed sensing further improves over these algorithms [15, 33, 59]. These schemes rely on a search that minimizes the difference between the overall signal and the superimposed signals in terms of the path parameters. However, the complexity of the algorithms is computationally prohibitive when multiple path parameters are estimated jointly, due to the extremely high number of possible combinations. To deal with that, UbiLocate firstly estimates the path parameters iteratively, and then refines them using the Nelder-Mead search algorithm.

Active localization. Here, the goal is to estimate the position of the device which sends the radio frequency signal. We can distinguish the following main approaches:

RSSI-based: The propagation losses of the radio frequency signal is modeled to estimate the distance between transmitter and receiver. Many well-known models can be found in the literature [6, 9, 16, 18, 29, 62]. Unfortunately, this approach has been demonstrated to provide limited accuracy compared to other approaches, as the received power depends on many environmental factors.

ToF-based: Timestamps are used in the MAC layer together with echoing techniques to measure round trip time [11, 14, 41, 61], and consequently the distance between AP and the target device. This can be extended using dead reckoning [34] to provide the user location with only a single AP. This concept was later standardized as the FTM protocol. It was tested in [23] and in [24], where the claimed sub-meter accuracy was validated. However, this accuracy can usually not be achieved in rich multipath environments, whereas our approach is better able to deal with multipath.

AoA-based: Estimating the angle of arrival from an incoming signal is a well-known topic in the field of array processing [28]. Combining angle of arrival measurements from several APs can provide very good localization accuracy. This was validated in [55] where sub-meter accuracy was achieved with large antenna arrays that are not yet feasible in commercial off-the-shelf (COTS) devices. This work was extended to COTS devices in [27] and in [31], where a two-dimensional (2D) MUSIC implementation (AoA+ToF) is carried out, improving the performance of 1D MUSIC at the cost of increasing the computational complexity. It has been also extended to 3D (AoA+ AoD + ToF) in [50].

Hybrid (RSSI/ToF + AoA)-based: Several systems combine angle and distance measurements to localize a device from a single AP. SPRING [40] combines AoA and ToF data derived from two separates hardware devices. Also CUPID [47] extracts angle information from CSI but uses only coarse RSSI to estimate distance. UbiLocate exploits both angle and ToF information in the same device.

NLOS. The NLOS case was rarely tackled in the past because it is an extremely challenging problem. Having the main path partially or completely obstructed by an object significantly complicates accurate path parameter estimation. The majority of prior works dealt with NLOS using the high bandwidth available in ultra-wideband systems [1, 19, 37, 44]. For WiFi there are several proposals for imaging and mapping through walls [1, 2, 51], but they need flexible high-performance hardware such as software-defined radios and custom antenna arrays. In addition, active anchors [10, 58] and reconfigurable intelligence surfaces [22, 32] help dealing with NLOS and improve positioning accuracy in NLOS cases, but such special purpose hardware is not available in regular WiFi deployments. While several localization systems claim to tackle NLOS issues, many of them evaluated the localization accuracy in mixed LOS/NLOS environments with a very high fraction of available LOS paths [27, 45]. None of them were evaluated under pure NLOS conditions. To the best of our knowledge, UbiLocate is the first WiFi location system that not only works in pure NLOS scenarios, but even achieves sub-meter accuracy.

WiFi testbeds. The most widely used CSI extraction tool for localization is [21] and a lot of works build upon this platform. However, it uses the outdated IEEE 802.11n standard, which limits the potential performance and foregoes the hardware capabilities of new WiFi standards such as 802.11ac. Designs based on software-defined radios are appealing due to their high-quality RF hardware, flexibility, and powerful processing capabilities of FPGAs. There are even full stack WiFi implementations for 802.11a/g/n and 802.11a/g/p available through openwifi [53] and GNU Radio [7]. With such software-defined radio systems, the clock can be sampled more accurately and with the reduced dispersion the ToF measurements would need little or no averaging compared to UbiLocate, whereas the CSI and thus angle estimation accuracy would be largely the same. However, for practical real-world deployments, it is of critical importance that location systems can be implemented on COTS devices without modification to the underlying hardware.

6 CONCLUSIONS

In this paper, we tackle the challenges of accurate wireless localization in *realistic* indoor WiFi deployments. While many works in the recent literature achieve excellent performance under ideal conditions with high AP densities, we target two critical assumptions that are key to realistic environments: i) the prevalence of NLOS paths when estimating a device position, and ii) the scarcity of APs, i.e., "anchor" nodes with known location. Based on these assumptions we developed UbiLocate, an IEEE 802.11ac-based WiFi location system that works with realistic AP deployment densities. UbiLocate exploits both a refined AoA extractor and a fine-grained ToF ranging system to achieve sub-meter accuracy even in tough NLOS conditions. Our experimental evaluation in a number of common scenarios shows an overall improvement of the localization performance by a factor of 2-3 compared to state-of-the-art systems, both under LOS and NLOS conditions.

Besides the high-accuracy location system, the framework presented in this paper provides extremely useful tools to the research community for wireless experimentation with *recent* COTS WiFi platforms, for example for wireless sensing, through wall imaging, activity recognition, rate adaptation, etc., using the improved hardware capabilities of IEEE 802.11ac. Finally, the ability to associate very accurate timestamps to complete CSI information is valuable for novel applications such as fine-grained clock synchronization and optimal scheduling of interfering links.

ACKNOWLEDGMENT

This research work was sponsored in part by the European Union's Horizon 2020 research and innovation programme under Grant No.871249 (LOCUS), the Spanish Ministry of Science and Innovation (MICIU) grant RTI2018-094313-B-I00 (PinPoint5G+) and the Region of Madrid through TAPIR-CM (S2018/TCS-4496).

REFERENCES

[1] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via RF body reflections. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 279–292, 2015.

[2] Fadel Adib and Dina Katabi. See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 75–86, 2013.

[3] alejandroBlancoPizarro. Ubilocate. https://github.com/IMDEANetworksWNG/UbiLocate, 2021.

[4] James Aspnes, Tolga Eren, David Kiyoshi Goldenberg, A Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian DO Anderson, and Peter N Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.

[5] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[6] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, volume 2, pages 775–784. Ieee, 2000.

[7] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. An ieee 802.11 a/g/p ofdm receiver for gnu radio. In *Proceedings of the second workshop on Software radio implementation forum*, pages 9–16, 2013.

[8] Zhe Chen, Guorong Zhu, Sulei Wang, Yuedong Xu, Jie Xiong, Jin Zhao, Jun Luo, and Xin Wang. M^3: Multipath assisted wi-fi localization with a single access point. *IEEE Transactions on Mobile Computing*, 2019.

[9] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2010.

[10] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. Rf-echo: A non-line-of-sight indoor localization system using a low-power active rf reflector asic tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 222–234, 2017.

[11] M Ciurana, F Barcelo-Arroyo, and F Izquierdo. A ranging system with ieee 802.11 data frames. In *2007 IEEE Radio and Wireless Symposium*, pages 133–136. IEEE, 2007.

[12] Carmelo Di Franco, Amanda Prorok, Nikolay Atanasov, Benjamin Kempke, Prabal Dutta, Vijay Kumar, and George J. Pappas. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 235–246, 2017.

[13] Bernard H Fleury, Martin Tschudin, Ralf Heddergott, Dirk Dahlhaus, and K Ingeman Pedersen. Channel parameter estimation in mobile radio environments using the sage algorithm. *IEEE Journal on selected areas in communications*, 17(3):434–450, 1999.

[14] Domenico Giustiniano and Stefan Mangold. Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[15] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616, 1997.

[16] Abhishek Goswami, Luis E Ortiz, and Samir R Das. Wigem: A learning-based approach for indoor localization. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[17] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. Free your csi: A channel state information extraction platform for modern wi-fi chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 21–28, 2019.

[18] İsmail Güvenc. *Enhancements to RSS based indoor tracking systems using Kalman filters*. PhD thesis, University of New Mexico, 2003.

[19] Ismail Guvenc, Chia-Chin Chong, and Fujio Watanabe. Nlos identification and mitigation for uwb localization systems. In *2007 IEEE Wireless Communications and Networking Conference*, pages 1571–1576. IEEE, 2007.

[20] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 339–350, 2008.

[21] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, pages 53–53, 2011.

[22] Chongwen Huang, George C Alexandropoulos, Chau Yuen, and Mérouane Debbah. Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2019.

[23] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. Verification: Accuracy evaluation of wifi fine time measurements on an open platform. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 417–427. ACM, 2018.

[24] Mohamed Ibrahim, Ali Rostami, Bo Yu, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Fan Bai, and Richard Howard. Wi-go: accurate and scalable vehicle positioning using wifi fine timing measurement. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 312–324, 2020.

[25] Yue Jin, Zengshan Tian, Mu Zhou, and Heng Wang. Mutrack: Multiparameter based indoor passive tracking system using commodity wifi. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[26] Kevin Jiokeng, Gentian Jakllari, Alain Tchana, and André-Luc Beylot. When ftm discovered music: Accurate wifi-based ranging in the presence of multipath. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1857–1866. IEEE, 2020.

[27] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 269–282, New York, NY, USA, 2015. Association for Computing Machinery.

[28] Hamid Krim and Mats Viberg. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine*, 13(4):67–94, 1996.

[29] Praveen Kumar, Lohith Reddy, and Shirshu Varma. Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks. In *2009 Fifth international conference on wireless communication and sensor networks (WCSN)*, pages 1–4. IEEE, 2009.

[30] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.

[31] Xiang Li, Shengjie Li, Daqing Zhang, Jie Xiong, Yasha Wang, and Hong Mei. Dynamic-music: accurate device-free indoor localization. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 196–207, 2016.

[32] Teng Ma, Yue Xiao, Xia Lei, Wenhui Xiong, and Yuan Ding. Indoor localization with reconfigurable intelligent surface. *IEEE Communications Letters*, 2020.

[33] Dmitry Malioutov, Müjdat Cetin, and Alan S Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE transactions on signal processing*, 53(8):3010–3022, 2005.

[34] Alex T Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. Sail: Single access point-based indoor localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 315–328. ACM, 2014.

[35] John A. Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[36] NexMon Project. https://github.com/seemoo-lab/nexmon/.

[37] Pat Pannuto, Benjamin Kempke, Li-Xuan Chuo, David Blaauw, and Prabal Dutta. Harmonium: Ultra wideband pulse generation with bandstitched recovery for fast, accurate, and robust indoor localization. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):1–29, 2018.

[38] S Unnikrishna Pillai and Byung Ho Kwon. Forward/backward spatial smoothing techniques for coherent signal identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(1):8–15, 1989.

[39] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.

[40] Maurizio Rea, Traian Emanuel Abrudan, Domenico Giustiniano, Holger Claussen, and Veli-Matti Kolmonen. Smartphone positioning with radio measurements from a single wifi access point. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 200–206, 2019.

[41] Maurizio Rea, Aymen Fakhreddine, Domenico Giustiniano, and Vincent Lenders. Filtering noisy 802.11 time-of-flight ranging measurements from commoditized wifi radios. *IEEE/ACM Transactions on Networking*, 25(4):2514–2527, 2017.

[42] Fabio Ricciato, Savio Sciancalepore, Francesco Gringoli, Nicolò Facchi, and Gennaro Boggia. Position and velocity estimation of a non-cooperative source from asynchronous packet arrival time measurements. *IEEE Transactions on Mobile Computing*, 17(9):2166–2179, 2018.

[43] Richard Roy and Thomas Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing*, 37(7):984–995, 1989.

[44] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Transactions on instrumentation and Measurement*, 66(8):2106–2117, 2017.

[45] Thuraiappah Sathyan, David Humphrey, and Mark Hedley. Wasp: A system and algorithms for accurate radio localization using low-cost hardware. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2):211–222, 2010.

[46] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[47] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262, 2013.

[48] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Successive interference cancellation: A back-of-the-envelope perspective. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6, 2010.

[49] Tie-Jun Shan, Mati Wax, and Thomas Kailath. On spatial smoothing for direction-of-arrival estimation of coherent signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(4):806–811, 1985.

[50] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, pages 376–388, 2018.

[51] Bo Tan, Kevin Chetty, and Kyle Jamieson. Thrumapper: Through-wall building tomography with a single mapping robot. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, pages 1–6, 2017.

[52] Ronald T Williams, Surendra Prasad, Arijit K Mahalanabis, and Leon H Sibul. An improved spatial smoothing technique for bearing estimation in a multipath environment. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(4):425–432, 1988.

[53] Jiao Xianjun, Liu Wei, and Mehari Michael. open-source ieee802.11/wi-fi baseband chip/fpga design, 2019.

[54] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[55] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *10th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI)*, pages 71–84, 2013.

[56] Jie Xiong, Kyle Jamieson, and Karthikeyan Sundaresan. Synchronicity: pushing the envelope of fine-grained localization with distributed mimo. In *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pages 43–48, 2014.

[57] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 537–549, 2015.

[58] Mingyu Yang, Li-Xuan Chuo, Karan Suri, Lu Liu, Hao Zheng, and Hun-Seok Kim. ilps: Local positioning system with simultaneous localization and wireless communication. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 379–387. IEEE, 2019.

[59] Jihao Yin and Tianqi Chen. Direction-of-arrival estimation using a sparse representation of array covariance vectors. *IEEE Transactions on Signal Processing*, 59(9):4489–4493, 2011.

[60] Moustafa Youssef, Ashok Agrawala, and Udaya Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, March 2003.

[61] Moustafa Youssef, Adel Youssef, Chuck Rieger, Udaya Shankar, and Ashok Agrawala. Pinpoint: An asynchronous time-based location determination system. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 165–176, 2006.

[62] Gergely V Zàruba, Manfred Huber, FA Kamangar, and Imrich Chlamtac. Indoor location tracking using rssi readings from a single wi-fi access point. *Wireless networks*, 13(2):221–235, 2007.