Wi-PRO: Accurate Indoor PNT using FTM with Multipath Suppression

William Hunter*, Roshan Ayyalasomayajula†, and Dinesh Bharadia*
*Department of Electrical and Computer Engineering, University of California, San Diego, USA
†Department of Computer Science and Engineering, University at Buffalo-SUNY, USA
Email: wshunter@ucsd.edu, roshana@buffalo.edu, dineshb@ucsd.edu

***Abstract*—Accurate indoor localization remains a significant challenge as GPS signals fail to penetrate indoor environments. While the IEEE 802.11mc standard introduced Fine-Timing Measurement (FTM) to enable Wi-Fi-based ranging, commercial off-the-shelf (COTS) implementations of FTM often suffer from ranging errors exceeding several meters due to proprietary hardware limitations and multipath interference. Existing solutions typically rely on crude statistical corrections to standardized 802.11mc ranging measurements using channel state information (CSI). In this paper, we propose Wi-PRO (Wireless Positioning using Range Offsets), a system designed to achieve high-accuracy indoor positioning using distributed, low-cost ESP32 nodes. Unlike previous statistical correction methods, Wi-PRO addresses the root causes of FTM inaccuracies in single-antenna systems. We identify and mitigate two primary sources of error: non-linear phase jumps across channel-bonded CSI measurements and incorrect start-time detection in hardware timestamping. By developing algorithms that calibrate low-quality CSI and compensate for multipath profiles, Wi-PRO significantly refines Time-of-Arrival (ToF) estimates. Experimental results demonstrate that our approach corrects FTM ranging offsets to provide meter-level localization accuracy, offering a scalable and cost-effective PNT solution for the modern IoT ecosystem. We release Wi-PRO's codebase and dataset to the open-source community[1].**

***Index Terms*—Indoor Localization, Fine-Time Measurement, Channel State Information (CSI), Multipath Mitigation, Internet-of-Things (IoT), Spectrum Sensing**

I. Introduction

Accurate indoor positioning, navigation, and timing (PNT) is a foundational capability for a wide range of emerging applications, including mobile robotics, augmented and virtual reality, asset tracking, and distributed sensing systems [1], [2]. While Global Navigation Satellite Systems (GNSS) have enabled precise outdoor PNT, their signals are unreliable or unavailable in indoor and dense urban environments [3]–[5]. This gap has motivated significant interest in radio-based indoor localization technologies that can leverage existing wireless infrastructure. Wi-Fi, due to its ubiquitous deployment, has emerged as a promising infrastructure solution for indoor PNT [1], [2]. This push towards indoor Wi-Fi localization has been standardized through IEEE's 802.11mc standard, also known as Fine-Time Measurement (FTM). FTM provide s range estimates between two Wi-Fi radios using a fast two-way exchange of packets. FTM promises accurate ranging for indoor environments to enable positioning and tracking.

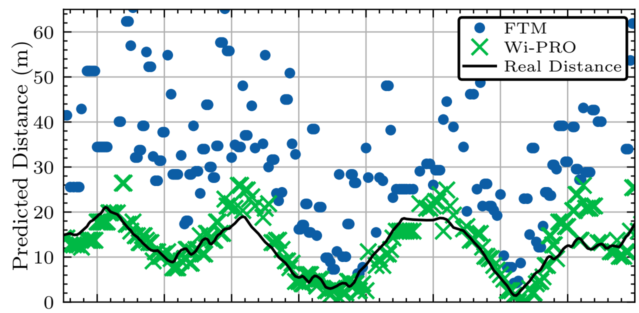


Fig. 1: Time series of a moving target's actual range vs. the ESP32's predicted range with FTM in an indoor environment. Standard implementations significantly overestimate actual range by as much as $2-3\times$. Wi-PRO reduces this overestimation significantly by compensating for the effect of reflections.

However, despite its theoretical appeal and widespread availability, practical deployments of FTM on commercial off-the-shelf devices have consistently reported large ranging errors in indoor environments [6]–[8], limiting the effectiveness of FTM for accurate and robust localization [9].

This lack of accurate and robust localization stems from a more fundamental mismatch between the physical propagation of wireless signals and the way commodity Wi-Fi radios generate timestamps. Existing FTM [8]–[10] implementations implicitly assume that the packet reception timestamp corresponds to the arrival time of the Line-of-Sight (LoS) signal. These works adjust the final range prediction using statistical models conditioned on the received signal strength (RSSI) and range estimate. However, in multipath-rich indoor environments, this assumption does not hold. Due to synchronization and energy detection mechanisms in commodity radios, the reported timestamp is often aligned with the centroid of the channel impulse response (CIR)[2], rather than the first-arriving path. This presents a fundamental barrier to accurate range estimation, because reflected paths' longer delays cause the reception timestamp to shift backwards, causing FTM to over-estimate the range. This bias can be seen in figure 1, where we see very consistent and severe overestimation of range with a standard FTM implementation.

[1]https://github.com/ucsdwcsng/wi-pro-esp32-ftm

[2]The time-series Fourier inverse of CSI

Consequently, many state-of-the-art Wi-Fi ranging frameworks use Channel State Information (CSI), which allows access to the raw wireless signal's phase and amplitude that can be used to measure range [11], enabling better ranging algorithms. Works like FUSIC [6] and FSI [7] utilize CSI in tandem with FTM to refine ranging, with the observation that ranging errors are chiefly caused by the Wi-Fi signal propagating along paths that are not a straight line, breaking the FTM algorithm's core assumption. This can be observed in the fact that nearly all erroneous FTM measurements overestimate the target's range. These methods estimate the relative strength of different propagation paths to isolate the direct-path (straight line) signal, thereby correcting errors in the final range estimate. However, these works are only able to obtain 40% improvement over standard FTM, and require averaging of many packets to suppress the effect of multipath. Moreover, these averaging-based techniques sacrifice temporal resolution and mobility support, making them unsuitable for dynamic scenarios.

Our Contributions:

In Wi-PRO, we optimally solve the timing and multipath problem in the FTM protocol, IEEE 802.11mc [12], using (a) a propagation path recovery algorithm which corrects non-idealities in the channel measurement that reduce the accuracy of the LoS path, and (b) a bandwidth-compensated leading edge detection algorithm to make packet timestamp measurements consistent with the LoS path. We combine these algorithmic innovations to achieve state-of-the-art ranging accuracy and implement them into a localization platform which runs on an off-the-shelf ESP32-S3 embedded platform costing less than 6$ per node, which we plan to open-source.

CIR reconstruction – When the Wi-Fi radio measures the channel, certain frequencies are nulled since they are not used for communication. This distorts the measured channel, widening peaks and introducing ripple artifacts that reduce ranging accuracy. Prior work does not account for these errors in its estimate of the propagation delay, reducing performance. Wi-PRO instead recovers the nulled subcarriers by minimizing the L_1 norm of the CIR, which naturally reduces the artificial side lobes created by guard-band nulling. This convex optimization approach is lightweight enough to implement at the edge, and allows for more accurate detection of the line-of-sight propagation path.

Line-Of-Sight Recovery – Secondly, we need to identify the correct start time of the direct path from the multipath riddled CIR plot as shown in Fig. 2. Most of the current works focus only on detecting peaks of received power for time-stamping packet reception. We observe that this method causes significant bias on low-bandwidth systems such as Wi-Fi, where multiple paths up to 6 meters apart can merge into a single peak, making it impossible to recover which one is the shortest through peak detection alone. Wi-PRO develops a simple compensated leading-edge detection strategy which significantly increases ranging accuracy.

Real-World Deployments – Finally, we deploy Wi-PRO across a distributed set of ESP32s (IoT devices) spread across

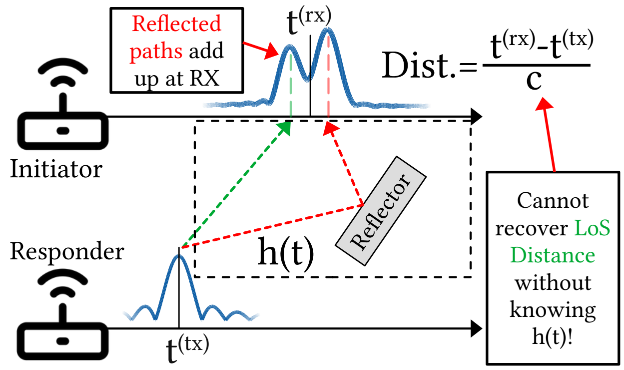


Fig. 2: How the channel affects time-delay measurement in FTM. Reflectors in the environment cause delayed copies of the packet to add up at the receiver, and the receiver places its received timestamp in the middle of these delays, causing an overestimation of the distance between the nodes.

a total area of $1440m^2$ over two scenarios. We record the ground-truth position of a mobile robot [13] equipped with a LiDAR, allowing us to test Wi-PRO in mobile scenarios. Using traditional trilateration algorithms, we demonstrate that Wi-PRO's range estimates achieve a median 1.5m localization error, a 2-3× improvement over the state-of-the-art FTM compensation system [6].

II. Primer: Wi-Fi Sensing Model

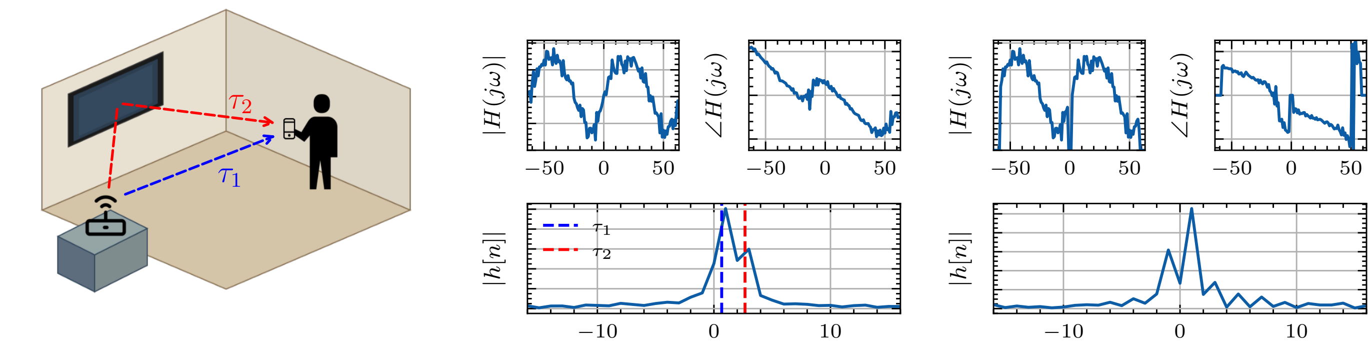
We first give a quick primer on how fine-timing measurement (FTM) measures range between two Wi-Fi modules, and how multipath affects this range.

A. Two-way Ranging

In 802.11mc Fine Time Measurement (FTM) [12], two Wi-Fi radios, the 'initiator' and 'responder,' exchange packets and record the time at which the exchanged packets are transmitted and received. With a two-way exchange in which both devices transmit and receive packets, the propagation delay, τ, and clock offset, Δ_t, between the initiator and responder clocks can be measured. If the signal propagates along the line-of-sight (LoS) path, the distance between the initiator and responder is τ times the speed of light. Denoting $t^{(0)}$ as the time of the responder's transmission to the initiator, $t^{(1)}$ as the time of the initiator's reception of the responder's packet, and $t^{(2)}, t^{(3)}$ as the corresponding TX/RX times for the response packet, we can compute:

$$t^{(1)} = t^{(0)} + \Delta_t + \tau \tag{1}$$
$$t^{(3)} = t^{(2)} - \Delta_t + \tau \tag{2}$$
$$\Rightarrow \tau = \frac{1}{2}[(t^{(3)} - t^{(2)}) + (t^{(1)} - t^{(0)})] \tag{3}$$
$$\Rightarrow \Delta_t = \frac{1}{2}[(t^{(3)} - t^{(2)}) - (t^{(1)} - t^{(0)})] \tag{4}$$

This technique works well in the case where there is only a single propagation path, such as in a wired channel or in



(a) Example scenario with two propagation paths.

(b) Real CSI and CIR.

(c) CSI and CIR measured at the Wi-Fi radio.

Fig. 3: A demonstration of the effect of multipath on the CIR and the radio receiver's corruption of the CIR. 3a: In indoor environments, strong reflected paths alongside the direct path are typical. 3b: Real CSI and CIR magnitude for this scenario. The CIR has two peaks at delays corresponding to the two paths. 3c: Actual CSI and CIR measured at the ESP for this scenario. The peaks in the CIR no longer correspond to the real propagation paths.

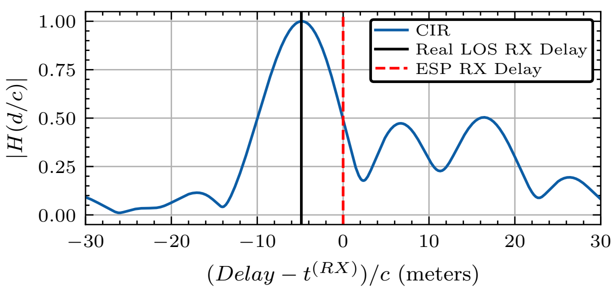


Fig. 4: Real example of the measured CIR after phase corrections and interpolation. The baseband hardware places $t = 0$ near the center of the impulse response, despite the direct path clearly arriving several meters before this point.

free space, but in the indoor environment, assigning a single propagation delay τ to the propagation is insufficient to capture the effect of multipath.

B. Multipath's Effect on the RX Timestamp

Consider the initiator and responder from section II-A. A complex baseband waveform $x(t)$ sent at time $t^{(0)}$ sent from the responder will be convolved with the CIR $h(t)$, and with the clock offset of Δ_t will be received at the initiator as:

$$y(t) = x(t) * h(t) \tag{5}$$

In the case when there are no reflectors, $h(t)$ is a single delta function at $\tau = \frac{d}{c}$, and the receiver will detect the received packet $y(t)$ exactly τ seconds after it is transmitted. However, when $h(t)$ has several propagation paths, $y(t)$ will be several shifted copies of $x(t)$. In this case, we observe that the Wi-Fi hardware does not detect the received packet exactly when the LoS path arrives. Instead, we observe that it tends to place the transmit timestamp near the 'center of mass' of $h(t)$, causing an overestimation of the LoS distance. We demonstrate this effect in figure 2.

We have now identified the critical source of timestamping error on the Wi-Fi radio – Multipath, causing the packet reception timestamp to not trigger exactly when the packet is first received. We have also demonstrated that these errors affect the received CSI, so by measuring the CSI, we can attempt to correct them. Therefore, we design an algorithm, Wi-PRO, to estimate the CIR $h(t)$ from the Wi-Fi channel, and then recover the true LoS reception time from the CIR.

III. Design

In order to estimate accurate range between two Wi-Fi radios, the measured channel state-information (CSI) is corrected to reconstruct a more robust Channel Impulse Response (CIR). Then, the reconstructed CIR is used to detect the correct timestamp for the peak corresponding to the direct path in Section III-B. Finally, Section III-C provides the details of Wi-PRO's deployment in real-world settings.

A. Reconstructing the CIR

Recall that we must obtain the CIR, $h(t)$, in order to adjust the received timestamp to align with the line-of-sight path. We can model the CIR as:

$$h(t) = \sum_{i=1}^{P} \alpha_i \, \delta(t - \tau_i) \tag{6}$$

Where τ_i are the path lengths and α_i are complex gains of each path. We cannot directly measure the CIR, but the radio hardware measures the CSI, a band-limited, frequency-domain version of the CIR. A sufficient model for the CSI is

$$\mathbf{H}[j\omega_n] = \frac{\mathbf{Y}[j\omega_n]}{\mathbf{X}[j\omega_n]} = \sum_{i=1}^{P} \alpha_i e^{-j\omega_n \tau_i} \tag{7}$$

Where ω_n are the base-band subcarrier frequencies. Consider $\mathbf{H}[j\omega]$[3] in the case where there are two propagation

[3] CSI for all N subcarriers $\mathbf{H}[j\omega] = [H[j\omega_1], H[j\omega_2], \cdots, H[j\omega_N]]$

paths, a line-of-sight path of delay τ_1 and a reflected path of delay τ_2. We should expect a CIR with two peaks centered at τ_1 and τ_2. However, since the typical Wi-Fi radio is designed for communication and not for sensing, some additional transformations are applied to the CSI, corrupting the ideal CIR. Without undoing the effect of these transformations, we will not be able to find the line-of-sight path length and recover the user's distance from the access point. Figure 3 demonstrates the cumulative effect of these errors.

Observing figure 3, we see several problems: There is a $\pi/2$ phase offset between the upper and lower half of the CSI due to channel-bonding [14], several subcarriers near DC and at the edges are nulled, and the entire CSI has been shifted so its centroid in time lies at $t = 0$. We un-rotate the upper half-band to remove the first issue, but removing the zeroed subcarriers and recovering the true time-shift in the CIR is nontrivial. We briefly discuss the reason some subcarriers are nulled, then discuss how we undo the effect of this nulling to construct an accurate CIR.

40MHz Wi-Fi splits a symbol into 128 orthogonal subcarrier frequencies. Each of these frequencies carries an independent data stream, so the receiver must measure CSI separately for each subcarrier frequency. In practice, the Wi-Fi standard does not use all 128 subcarriers. 5 carriers on the high and low ends of the spectrum are unused, to reduce the effect of interference between adjacent channels. 3 carriers around the center frequency are also left unused to simplify the receiver circuitry, and 6 carriers called 'pilots' transmit dummy data. The Wi-Fi radio reports $\mathbf{H}[j\omega]$ for the data and pilot subcarriers, but the other subcarriers are left at zero. This zeroing in the frequency domain causes a ripple-like effect in the CIR which causes adjacent paths to interfere slightly with each other, reducing the accuracy to which we can detect the location of the line-of-sight path.

Now, we discuss Wi-PRO's simple algorithm to reconstruct the nulled subcarriers. Here we denote the set of zeroed subcarriers as Ω_Z, and the known data subcarriers as Ω_D. The Fourier Transform of the CSI with the zeroed subcarriers cause a distortion of the CIR which manifests as a widening of its envelope and an increase in side-lobe ripples over the ideal $sinc$ function. This distortion can be better visualized through up-sampling the CIR by taking an IFFT of the CSI using sub-sample spacing between output time points. This allows us to sample the continuous-time $h(t)$ at a finer resolution as shown in 4. To avoid these distortions our first goal is to choose interpolation values for the unknown nulled subcarriers' channels, $\mathbf{H}[j\Omega_Z]$, given the known $\mathbf{H}[j\Omega_D]$, that best approximates the real $\mathbf{H}[j\omega]$.

A classical solution to this problem is the linear-phase-fitting method, where we fit a linear slope to $\angle\mathbf{H}[j\omega]$, and choose values for $\angle\mathbf{H}[j\Omega_Z]$ that fit this slope. The corresponding magnitudes can be chosen as the average magnitude of the adjacent known portions of $\mathbf{H}[j\omega_D]$. This method performs well when $h(t)$ is dominated by a single path, but causes significant distortion when there are paths with similar strength in $h(t)$. We observe this effect in figure 5.

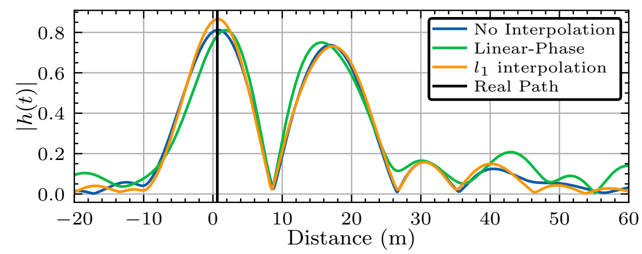


Fig. 5: Estimated CIR for a 2-path channel using 3 techniques: No interpolation, linear-phase interpolation, and our method. The top plot shows the entire non-zero part of the CIR, the bottom left shows a zoom-in on the first peak, and the bottom right shows a zoom in on the rising portion of the first peak. Linear-phase interpolation causes large distortions in peak location up to 2 meters, and no interpolation results in a wider 'sinc' function, hurting our ability to detect the CIR's rising edge and reducing ranging accuracy by 10cm. (see figure 9a.)

The linear-phase approximation fails because the recovery of $h(t)$ from incomplete frequency measurements is inherently underdetermined, and any interpolation method implicitly encodes assumptions about the underlying channel structure. Linear-phase fitting assumes a single dominant path, which produces a linear phase response across frequency, but this assumption breaks down under multipath. We propose a simple interpolation method which better matches the distribution of real world channels, which is to choose v_ω, interpolation values for $\mathbf{H}[j\Omega_Z]$, which minimize the L_1 norm of the CIR:

$$\hat{\mathbf{H}}[j\Omega_Z] = \arg\min_{v_\omega} \sum_{n=0}^{N} \left|\mathcal{F}^{-1}\{\mathbf{H}[j\omega]\}[n]\right| \tag{8}$$

$$\mathbf{H}[j\omega] = \begin{cases} v_\omega & \omega \in \Omega_Z \quad \text{(optimization variable)} \\ \tilde{\mathbf{H}}[j\omega] & \omega \in \Omega_D \quad \text{(known)} \end{cases} \tag{9}$$

Where $\mathbf{H}[j\Omega_D]$ are the known CSI values and $\mathbf{H}[j\Omega_Z]$ are the unknown optimization variables, and $N = 128$.

This method effectively constrains the CIR to have a lower number of peaks. Since nulling tends to increase the side lobes of every peak and increase the L_1 norm of the CIR, we choose interpolation values that minimize this norm. The minimization problem in equation 8 is convex and converges in 5-10 steps of gradient descent in our implementation. The Wirtinger gradient of the loss term in the RHS of equation 8 is:

$$\nabla_{\mathbf{H}[j\omega]} = \mathcal{F}\{s(\mathcal{F}^{-1}\{\mathbf{H}[j\omega]\})\}[j\omega] \tag{10}$$

$$s(x) = \begin{cases} x/|x| & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{11}$$

Where $s(x), x \in \mathbf{C}$ is the "complex sign" operation, which returns a unit-magnitude complex number pointing in the same direction as the input. We use this expression to optimize eq. 8

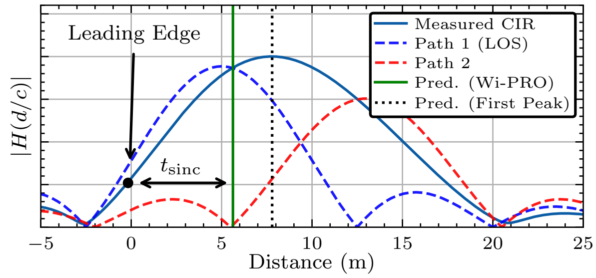


Fig. 6: Simulated CIR with two paths: one at 5 meters and one at 13 meters. The contributions to $h(t)$ from each path are shown in blue and red dotted lines. Since the paths are close together, they combine into a single peak, causing normal first-peak-index detection (black dotted line) to fail. Compensated leading edge detection (green) recovers the true first path.

via gradient descent. With an exact estimate of the CSI across all subcarriers, we can next correct for the effect of multipath.

B. Re-alignment and Edge Detection

For the purpose of ranging, our goal is to find the time of the LoS path in the CIR. By the triangle inequality, we know the LoS path must be the shortest path. The canonical method of detecting the shortest path is the 'First-Peak-Index' method. In this method, the first peak in $|h(t)|$ which is above an empirically-chosen threshold is selected as the LoS path. This method works well when all paths are widely separated, but fails when there are paths with similar distances, because nearby paths can merge together to create a single large peak. Figure 6 demonstrates a simulated channel in which this is the case. Two propagation paths are present, one at 5 meters and one at 13, but their sum results in only a single large peak at 8 meters, causing the first-peak-index method to fail.

The Ultra Wide-Band (UWB) literature [15], [16] has an alternative solution to this problem, which is to measure when the CIR first rises above a threshold, known as 'Leading-Edge Detection.' In this method, we choose the reception timestamp as the point at which the channel magnitude first rises above a threshold value T. The leading edge of our example CIR from before is demonstrated in figure 6. This method works well for UWB, but in a lower-bandwidth system like Wi-Fi, every propagation path is convolved with a sinc function due to the low sampling rate. Directly using the leading edge causes us to choose the earlier portion of the sinc, and significantly underestimate the range, as shown in figure 6. We can estimate what the error term t_{sinc} should be by computing when a sinc function with magnitude 1 crosses the threshold value T, and add this bias to the measured leading edge, leaving us with the final channel rise time:

$$\mathrm{t}_{\text{rise}} = t_{\text{sinc}} + \inf_t\{|h(t)| > T\} \tag{12}$$

In our evaluations, we normalize the CIR by setting its maximum magnitude to 1, and demonstrate that values of T between $[0.2, 0.5]$ are near-optimal (Sec IV-B4.)

We have now determined the correction term, t_{rise}, which picks out the LoS path in the measured CIR. The final step is to consider how it can be applied to the FTM algorithm. Recall that the Wi-Fi radio records the timestamps $t^{(1)}, t^{(3)}$ at $t = 0$ in the respective CIR. We do not have the CIR for $t^{(3)}$, but we have assumed $h_{\text{u}}(t) = h_{\text{d}}(t)$, so we assume $t^{(1)}_{\text{rise}} = t^{(3)}_{\text{rise}}$. Thus, we have:

$$t^{(1)}_{\text{LoS}} = t^{(1)} + t^{(1)}_{\text{rise}} \tag{13}$$
$$t^{(3)}_{\text{LoS}} = t^{(3)} + t^{(1)}_{\text{rise}} \tag{14}$$
$$\tau_{\text{LoS}} = \frac{1}{2}[(t^{(3)}_{\text{LoS}} - t^{(2)}) + (t^{(1)}_{\text{LoS}} - t^{(0)})]$$
$$\Rightarrow \tau_{\text{LoS}} = \tau + \mathrm{t}^{(1)}_{\text{rise}} \tag{15}$$

Thus, we can compensate for multipath-induced bias simply by using the CIR at one end of the FTM exchange, and adding the rise-time of the CIR to it.

C. Implementation

We deploy our algorithm on commercial-off-the-shelf ESP32-S3 devices [17], with the entire CSI interpolation and range compensation process running entirely on the ESP32. As our algorithm requires paired CSI and FTM measurements, we extract CSI data from the ESP32 every time it runs an FTM exchange. The ESP32s are configured to run a callback function every time a Wi-Fi packet is received, which gives us access to the CSI, transmitting MAC address, RSSI, and other features. We also patch the ESP's firmware to additionally emit the high-accuracy timestamp which is used in FTM with every CSI packet, allowing us to associate CSI measurements to FTM exchanges. When raw CSI and FTM data have been associated, the ESP32 performs the subcarrier interpolation, upsampling, and rise time detection steps, and outputs the final range estimate over USB to a host device. Total processing takes 24ms on the ESP32-S3, allowing Wi-PRO to process packets in real time. The ESP additionally outputs the raw CSI and FTM measurements for benchmarking purposes. All source code and data are available at https://github.com/ucsdwcsng/wi-pro-esp32-ftm.

IV. Evaluation

We evaluate Wi-PRO's positioning performance by deploying it in 2 different environments (DS1 and DS2), spanning a total area of over $1400m^2$. In each environment we deploy 3-6 ESP32-S3 **anchor** nodes each connected to a raspberry Pi 4. We deploy another **tag** ESP32-S3 on a wheeled Turtlebot 3 robot with a LiDAR running ROS2's Slam Toolbox [13] for mapping, which we use to generate ground-truth position data as well as a 2-d map of the environment. In total DS1 contains 213 robot locations and DS2 296. DS1 is designed to be an 'easy' dataset in which LoS is maintained to all 3 anchors at all times, and DS2 is a 'hard' dataset in which there is typically only LoS to 2-3 of the 6 anchors, and strong metal reflectors to NLoS anchors. Additionally, in DS2 the tag moves over an entire office building floor, and at times goes

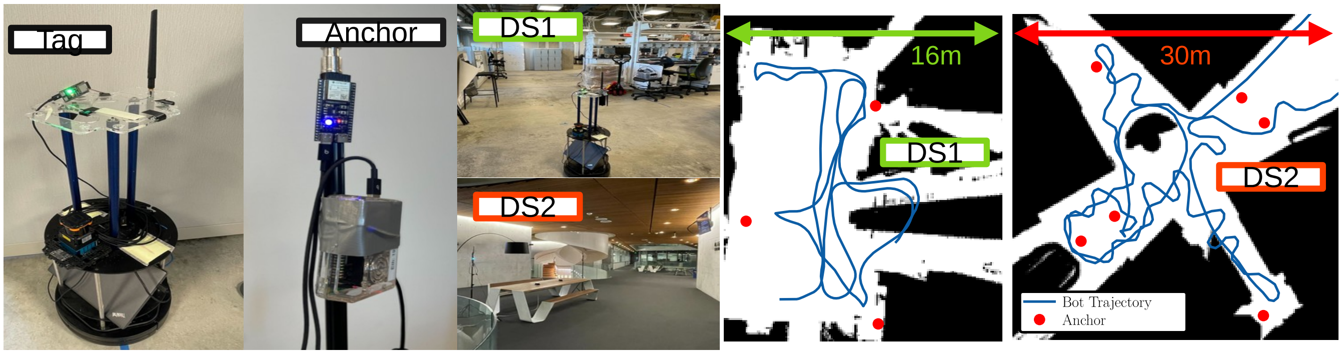


Fig. 7: Setup and Deployment of Wi-PRO: Left: Robot with ESP32 tag for ground-truth data collection, Wi-PRO anchor on tripod with RPi. Center: Environment photos of dataset 1 (DS1) and dataset 2 (DS2). Right: LiDAR maps of DS1 and DS2, robot trajectory in blue, anchor locations in red. DS1 spans roughly 12x20 meters, DS2 spans 30x40 meters, with up to 50 meter max range.

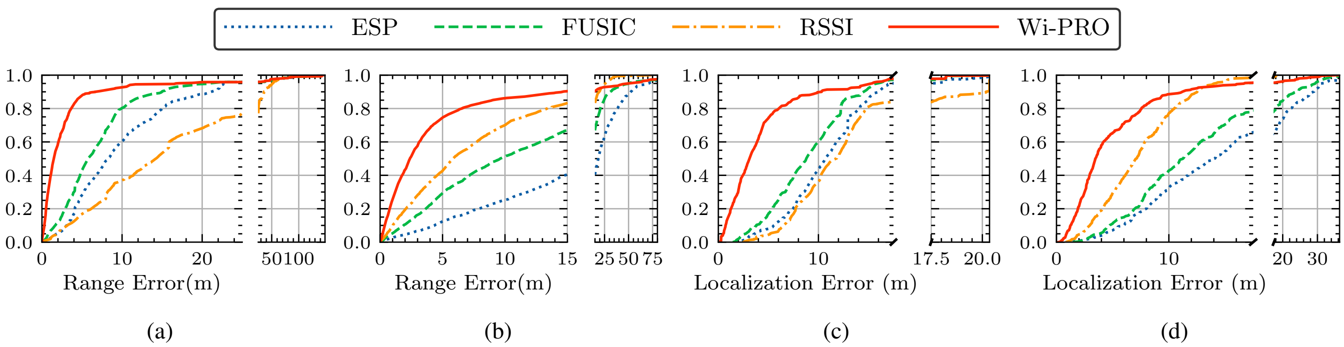


Fig. 8: (a,b) Range error CDF for (DS1,DS2) respectively. A split X-axis is used due to the long tail of range errors. (c,d) X-Y localization error CDF for (DS1,DS2).

up to 30 meters away from the nearest anchor. Figure 7 shows images of the tag and anchor design, and photos and LiDAR maps of each dataset. In both datasets we choose the leading edge threshold T as 25% of the strongest path's power and use Wi-Fi channel 6 at 40MHz bandwidth for all transmissions.

The goal is for the tag to localize itself using the estimated range to each of the anchors, whose location we assume is known. Localization is performed using a standard trilateration algorithm. The tag on the robot makes round-robin FTM initiations to the anchors, allowing it to obtain up-to-date range estimates. At every point in the robot's trajectory, we take the most recent FTM range measurement to each anchor, as well as the measured RSSI, and compute:

$$\mathcal{L}(x,y) = \sum_{i=1}^{N_{\text{anchor}}} 10^{\frac{R_i}{20}} \left(\left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right\| - c\tau_i \right)^2 \tag{16}$$

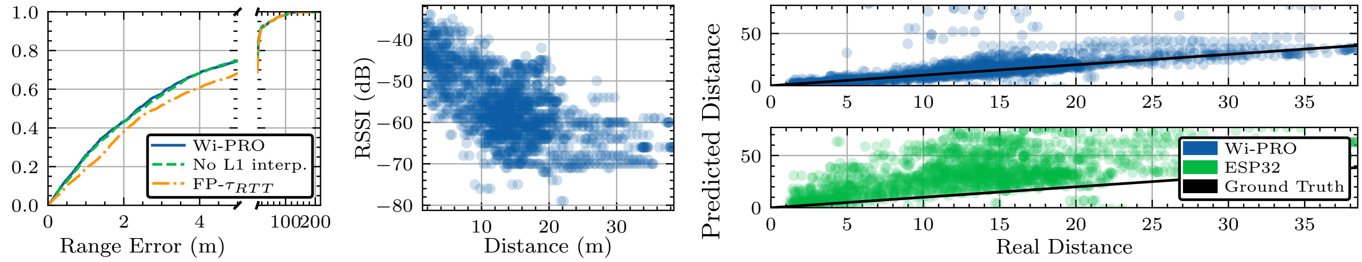
$$\hat{x}, \hat{y} = \arg\min_{x,y} \mathcal{L}(x,y) \tag{17}$$

In which x_i, y_i are the x and y positions of anchor i, R_i is the most recent RSSI and τ_i the most recent propagation delay from equation 15. We ignore anchors for which no FTM exchange has been made in the last 5 seconds.

A. *End-to-End Performance*

1) Ranging Accuracy: First we evaluate the ranging accuracy of Wi-PRO on both datasets. For all anchors deployed in each environment, we plot the CDF (Cumulative Distribution Function) of the absolute error between the ground truth from the robot and predicted range. Plots for DS1 and DS2 are visible in figures 8a, 8b respectively. Wi-PRO obtains a 1.51 meter median error on DS1 and 2.39 meter median error on DS2. 90^{th} percentile error is 6.65m on DS1 and 14.5m on DS2. We also benchmark against FUSIC [6]'s FTM correction algorithm, the ranges reported by the ESP32, and a simple RSSI baseline based on the ITU-R propagation model [18]. Wi-PRO obtains 3-4$\times$ better median range accuracy than the out-of-the-box ESP32 algorithm across the board. Note that non-Wi-PRO FTM algorithms outperform RSSI in a LoS scenario (DS1) but degrade significantly in a NLoS scenario (DS2,) validating that multipath is the primary cause of poor FTM performance. Table I reports the median and 90^{th} percentile for all algorithms.

2) Localization Accuracy: We also examine the x-y localization accuracy obtainable with the predicted range. We plot the CDFs of the error between the predicted x-y tag location and the robot's actual position. Figures 8c and 8d show these CDFs for dataset 1 and 2. Again, we compare



(a) (b) (c)

Fig. 9: (a) CDF of range error on DS2 with different parts of the algorithm disabled. See section IV-B1 for description of each legend item. (b) Range vs. RSSI for DS2. (c) Top: Predicted range vs. actual range Wi-PRO on DS2. Bottom: Equivalent plot for the raw ESP32. There is a significant bias visible in the ESP32's measurement, which is not a function of only range.

TABLE I: Ranging & Localization Error: Median (90^{th})

Algorithm	DS1 Range	DS2 Range	DS1 Loc.	DS2 Loc.
Wi-PRO	**1.51 (6.65)**	**2.39 (14.50)**	**3.17 (10.32)**	**3.52 (11.17)**
FUSIC	5.73 (14.98)	9.50 (30.11)	8.78 (14.52)	11.25 (23.89)
ESP32	8.23 (21.23)	18.52 (52.29)	10.68 (16.02)	14.02 (28.76)

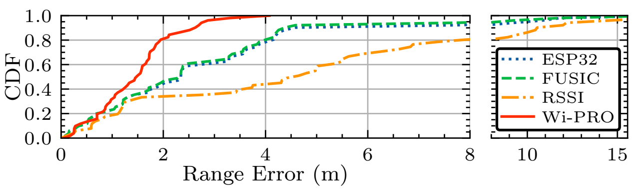


Fig. 10: Ranging error in an outdoor environment with little multipath.

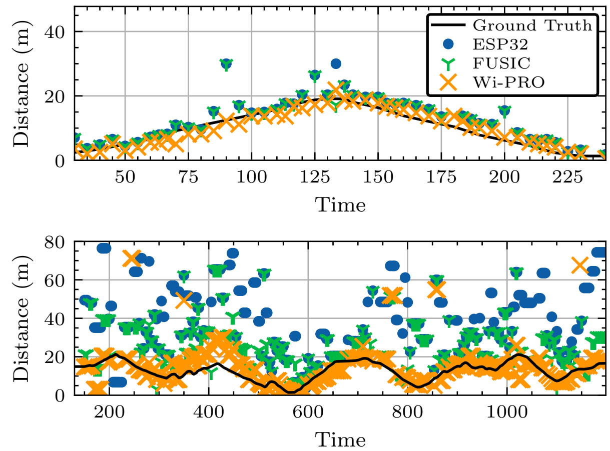


Fig. 11: Time-Series range predictions at one anchor for the ESP32's basic algorithm, FUSIC, and Wi-PRO. Top: Outdoor scenario with little multipath. Bottom: Dataset 2.

against FUSIC and the base ESP32 range, and show median and 90th percentile errors in table I. We obtain 3-4 meter median positioning error, a 2-3$\times$ improvement over the base ESP32 range. Note that all performance here is using single-shot range measurement to each anchor with no averaging or outlier rejection. A full implementation of a localization system would attain better performance by fusing with other sensors such as IMUs and taking into account multiple past measurements. Our focus in this work is on improving raw range measurements, not developing a full localization system.

B. Ablation Studies

We further perform a few ablation studies to understand the importance of various components of Wi-PRO and Wi-PRO's ranging estimation's robustness under various environmental conditions.

1) Algorithm Components: To understand which portions of Wi-PRO's algorithm contribute the most to improved localization accuracy, we test Wi-PRO on DS2 with different portions of the delay compensation disabled. Figure 9a shows CDFs for range error for a variety of permutations of our algorithm. Wi-PRO's full algorithm performs the best, the other variations are:

- *No L1 Interp.*: Null recovery from equation 8 is omitted, increasing the median error over Wi-PRO by 32cm.
- $FP - \tau_{RTT}$: The first-peak index instead of leading edge is used as the time correction, increasing median error by 47cm.

Table II gives median and 90^{th} percentile errors for the various algorithm permutations.

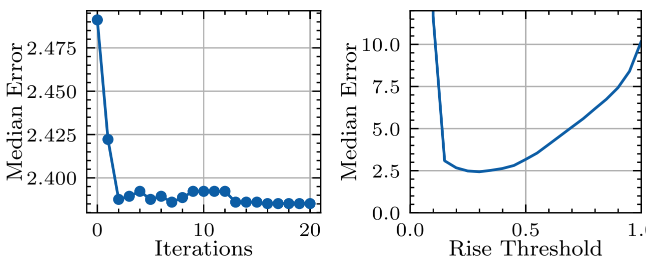
2) Outdoor Performance: We additionally collect one single-anchor dataset in an outdoor setting where there is little multipath. Figure 10 demonstrates range error CDFs, showing that Wi-PRO provides a small advantage over other implementations in this scenario. This is likely due to internal reflections inside the RF circuitry spreading the single transmission path into several close ones. To test this theory, we also run the 'First-Peak-Index' version of Wi-PRO, without leading edge detection. It performs somewhat in between Wi-PRO and prior methods, partially validating this theory. We plot time-series

TABLE II: Range Error: In meters for DS2

Algorithm	Wi-PRO	No L1 interp.	FP-τ_{RTT}
Median	**2.39**	2.49	2.87
90^{th}	**14.53**	14.44	16.11

errors in Fig. 11 for an indoor and outdoor scenario, to better visualize Wi-PRO's advantage over prior systems in indoor scenarios.

3) Range Bias: In figure 9c, we plot ground truth range on the X axis and predicted range on the Y axis to understand any correlation. The green ESP32 plot has a near-universal positive bias, indicating that multipath is indeed the primary source of ranging error. We also note that there is little functional correlation between the ESP's predicted range and its actual range. Figure 9b shows that there is also similarly weak correlation between the RSSI and the range, implying that data-driven methods alone cannot obtain accurate range estimation on our dataset. Using a log-linear fit, a typical model for indoor propagation path loss [19], this method only achieves a 3.84 meter median error. This performance is bolstered by the log-linear model being fit to the specific dataset collected, and this method cannot generalize to other transmitter locations, and is not robust to changes in path attenuations or transmitting tag.



(a) (b)

Fig. 12: (a): Median range error vs. number of l_1 minimization steps for DS2. (b): Median range error vs. threshold value T.

4) Hyperparameters: In figure 12a, plot the median ranging error on DS2 vs. the number of iteration steps in equation 8. We see that the problem is highly convex and converges after just 2 iterations. In figure 12b, we again plot median ranging error on DS2 but vary the rise-time selection threshold T. We find that values between 0.2-0.5 are acceptable.

V. Related Work

Several past works have attempted to improve the performance of FTM indoors. These works generally acknowledge that the chief reason for FTM's poor performance is due to multipath, and their solutions fall into two categories: signal-processing based solutions which use CSI, and statistical solutions which fit a bias model to the predicted range and RSSI.

Signal-Processing Methods: FUSIC [6] uses many FTM and CSI measurements to obtain many multipath profiles, averages them with MuSiC, and analyzes which peak is likely to correspond to the direct path based on their relative powers. It updates the FTM prediction when the first path's power falls below a threshold. FSI [7] uses a similar method, combined with an IMU, to estimate the tag position. However, these works fail to correctly model the source of timestamping error in the Wi-Fi hardware, as they only correct the range estimate in NLOS scenarios where the direct path is weak. Additionally, since they rely on MuSiC to average many packets to reduce noise in the CSI, they fail to generalize to mobile scenarios. We implement FUSIC's range correction algorithm as a benchmark, omitting the MuSiC averaging since we find that it results in unilaterally worse performance with a mobile tag.

Data-Driven Methods: [8] scans over a set of ranges, measuring the predicted range from FTM at each step, and then fits a monotonic curve to the data. This method requires hardware and location-specific calibration, and is not able to handle dynamic tag motion or changes in reflector position. [10], [9] train a neural network on measured range and RSSI, again requiring location-specific calibration that does not generalize. We do not benchmark against these systems, since we seek a general calibration-free solution.

Some works [20], [21], [22] have examined the ESP32's CSI measurement capabilities. These works are all focused on motion detection using variation in the CSI amplitude, and do not characterize the phase offsets that we observe. [23] also examines channel quality for timing indoors, but is focused only on carrier phase and time synchronization, instead of measuring accurate range. Outside of Wi-Fi, the UWB localization field has several works which attempt to correct range errors by NLOS path identification [24] [25]. Leading-edge detection is also sometimes used for identifying the LOS path in UWB [15] [16] to our knowledge, we are the first to explore this algorithm for a low-bandwidth system like Wi-Fi.

VI. Discussions & Future Work

We have presented Wi-PRO, a system that can remove the bias in FTM measurements indoors. This presents large potential for future lines of inquiry in the wireless localization field:

- **Developing robust localization systems**: While this work focuses solely on range estimation, there is a large body of literature on indoor wireless positioning which stands to benefit from increased ranging accuracy. Machine-learning methods like [26] or database-lookup methods like Skyhook [27] could be augmented with range or CIR estimates as features. For single-user self-localization scenarios, sensor fusion methods such as [5], [28], [29] could use range estimates to augment a smartphone or robot's internal sensors and obtain more accurate localization.
- **Indoor time distribution systems**: Parallel to delay estimation is clock offset estimation. While the estimated clock offset from FTM (equation 4) does not directly depend on the receiver identifying the direct path, dynamic multipath due to moving reflectors would cause variable delays between the uplink and downlink packet of the FTM exchange, causing a bias. Accurate direct-path recovery could lead to more accurate clock

synchronization systems, which in turn could enable applications like time-difference-of-arrival localization [30] or provide wireless frequency synchronization to align multiple distributed SDR receivers [31].

- **IoT Applications**: Many recent works [32], [33] consider range-based localization of a network of static devices with unknown initial positions. These works either only test their algorithms in simulation, or use high-cost UWB receivers. With a low-cost node design capable of measuring accurate range, many of these systems could be tested and implemented.

References

[1] F. Liu, J. Liu, Y. Yin, W. Wang, D. Hu, P. Chen, and Q. Niu, "Survey on wifi-based indoor positioning techniques," *IET Communications*, vol. 14, no. 9, pp. 1372–1383, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-com.2019.1059

[2] C. Yang and H.-r. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[3] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht, and D. Bharadia, "Deep learning based wireless localization for indoor navigation," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[4] R. Ayyalasomayajula, A. Arun, C. Wu, S. Rajagopalan, S. Ganesaraman, A. Seetharaman, I. K. Jain, and D. Bharadia, "{LocAP}: Autonomous millimeter accurate mapping of {WiFi} infrastructure," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 1115–1129.

[5] A. Arun, R. Ayyalasomayajula, W. Hunter, and D. Bharadia, "P2slam: Bearing based wifi slam for indoor robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3326–3333, 2022.

[6] K. Jiokeng, G. Jakllari, A. Tchana, and A.-L. Beylot, "When ftm discovered music: Accurate wifi-based ranging in the presence of multipath," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1857–1866.

[7] Y. Zhang, B. Lu, and W. Wang, "Fsi: A ftm calibration method using wi-fi physical layer information," in *Wireless Algorithms, Systems, and Applications*, L. Wang, M. Segal, J. Chen, and T. Qiu, Eds. Cham: Springer Nature Switzerland, 2022, pp. 365–374.

[8] M. Bullmann, T. Fetzer, M. Ebner, S. Kastner, F. Deinzer, and M. Grzegorzek, "Data driven sensor model for wi-fi fine timing measurement," in *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022, pp. 1–8.

[9] H.-W. Chan, A. I.-C. Lai, and R.-B. Wu, "Transfer learning of wi-fi ftm responder positioning with nlos identification," in *2021 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, 2021, pp. 23–26.

[10] V. Barral Vales, O. C. Fernández, T. Domínguez-Bolaño, C. J. Escudero, and J. A. García-Naya, "Fine time measurement for the internet of things: A practical approach using esp32," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 305–18 318, 2022.

[11] D. Vasisht, S. Kumar, and D. Katabi, "{Decimeter-Level} localization with a single {WiFi} access point," in *13th USENIX symposium on networked systems design and implementation (NSDI 16)*, 2016, pp. 165–178.

[12] A. Kulkarni and A. Lim, "Preliminary study on indoor localization using smartphone-based ieee 802.11 mc," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies*, 2019, pp. 43–44.

[13] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021. [Online]. Available: https://doi.org/10.21105/joss.02783

[14] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "The impact of channel bonding on 802.11 n network management," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, 2011, pp. 1–12.

[15] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for uwb localization," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1719–1728, 2012.

[16] Y. Xie, G. J. M. Janssen, S. Shakeri, and C. C. J. M. Tiberius, "UWB pulse detection and TOA estimation using GLRT," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 68, Sep. 2017. [Online]. Available: https://doi.org/10.1186/s13634-017-0500-1

[17] Espressif Systems, *ESP32-S3 Series Datasheet*, Espressif Systems (Shanghai) Co., Ltd., 2024, version 1.7. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32\-s3_datasheet_en.pdf

[18] ITU-R, "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 450 GHz," International Telecommunication Union, Geneva, Switzerland, Recommendation ITU-R P.1238-13, Sep. 2025. [Online]. Available: https://www.itu.int/rec/R-REC-P.1238/en

[19] D. J. Suroso, M. Arifin, and P. Cherntanomwong, "Distance-based indoor localization using empirical path loss model and rssi in wireless sensor networks," *Journal of Robotics and Control (JRC)*, vol. 1, no. 6, pp. 199–207, 2020.

[20] M. Atif, S. Muralidharan, H. Ko, and B. Yoo, "Wi-esp—a tool for csi-based device-free wi-fi sensing (dfws)," *Journal of Computational Design and Engineering*, vol. 7, no. 5, pp. 644–656, 05 2020. [Online]. Available: https://doi.org/10.1093/jcde/qwaa048

[21] T. Kitagishi, G. Hangli, T. Michikata, and N. Koshizuka, "Wi-monitor: Wi-fi channel state information-based crowd counting with lightweight and low-cost iot devices," in *Internet of Things*, A. González-Vidal, A. Mohamed Abdelgawad, E. Sabir, S. Ziegler, and L. Ladid, Eds. Cham: Springer International Publishing, 2022, pp. 135–148.

[22] J. Strohmayer and M. Kampel, "Wifi csi-based long-range through-wall human activity recognition with the esp32," in *Computer Vision Systems*, H. I. Christensen, P. Corke, R. Detry, J.-B. Weibel, and M. Vincze, Eds. Cham: Springer Nature Switzerland, 2023, pp. 41–50.

[23] T. Ouchi, T. Hatagi, S. Akasaka, S. Kameda, S. Yasuda, and N. Shiga, "Evaluation of time synchronization accuracy of wireless two-way interferometry (wi-wi): Consideration of distance dependence in nlos environment," in *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2024, pp. 109–111.

[24] I. Guvenc, C.-C. Chong, and F. Watanabe, "Nlos identification and mitigation for uwb localization systems," in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 1571–1576.

[25] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for uwb localization," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1719–1728, 2012.

[26] N. Singh, S. Choe, and R. Punmiya, "Machine learning based indoor localization using wi-fi rssi fingerprints: An overview," *IEEE Access*, vol. 9, pp. 127 150–127 174, 2021.

[27] P. A. Zandbergen, "Comparison of wifi positioning on two mobile devices," *Journal of Location Based Services*, vol. 6, no. 1, pp. 35–50, 2012. [Online]. Available: https://doi.org/10.1080/17489725.2011.630038

[28] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/1/715

[29] W. W. Ninad Jadhav, "Wsr: A wifi sensor for collaborative robotics," *ArXivorg*, 2021. [Online]. Available: https://par.nsf.gov/biblio/10323910

[30] R. Kaune, J. Hörst, and W. Koch, "Accuracy analysis for tdoa localization in sensor networks," in *14th International Conference on Information Fusion*, 2011, pp. 1–8.

[31] M. Krueckemeier, F. Schwartau, C. Monka-Ewe, and J. S. Technische, "Synchronization of multiple usrp sdrs for coherent receiver applications," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, 2019, pp. 11–16.

[32] P. N. Beuchat, H. Hesse, A. Domahidi, and J. Lygeros, "Enabling optimization-based localization for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5639–5650, 2019.

[33] L. T. Nguyen, J. Kim, S. Kim, and B. Shim, "Localization of iot networks via low-rank matrix completion," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5833–5847, 2019.