Is WiFi 802.11mc Fine Time Measurement Ready for Prime-Time Localization?

Shivang Aggarwal[1], Ramanujan K Sheshadri[2], Karthikeyan Sundaresan[3], Dimitrios Koutsonikolas[1]
[1]Institute for the Wireless Internet of Things, Northeastern University [2]NEC Laboratories America
[3]Georgia Institute of Technology

ABSTRACT

WiFi's fine time measurement (FTM) based ranging protocol has set the stage for mass adoption of location-aware applications and services in WiFi-pervading enterprise and consumer ecosystems. However, the lack of deployment of such commercial-scale localization solutions has motivated us to conduct a comprehensive experimental study that aims to verify *whether WiFi's FTM is indeed ready for prime-time localization.*

With *heterogeneity* in operation (devices, environments, and spectrum) being the fundamental essence of commercial deployments, our study focuses on FTM's ability to deliver useable localization under such practical conditions. Being a first of its kind, our study reveals several interesting insights for practical operation of FTM, with the most critical of them being its inability to eliminate substantial offsets in estimated ranges between heterogeneous devices and configurations that degrade performance significantly (up to 20 m error). Albeit a negative result for FTM's readiness, we also propose a simple but promising remedy – an over-the-top auto-calibration solution that allows every WiFi device, when it enters an enterprise environment, to self-calibrate its offsets on-demand, thereby salvaging FTM to render it useful (median error of 2 m) for localization.

CCS CONCEPTS

• **Networks** → *Wireless local area networks*; • **Human-centered computing** → *Ubiquitous and mobile devices.*

ACM Reference Format:
Shivang Aggarwal, Ramanujan K Sheshadri, Karthikeyan Sundaresan, Dimitrios Koutsonikolas. 2022. Is WiFi 802.11mc Fine Time Measurement Ready for Prime-Time Localization?. In *16th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterizatiion (WiNTECH'22), October 17, 2022, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3556564.3558234

1 INTRODUCTION

WiFi localization solutions received a significant boost with the introduction of the Fine Time Measurement (FTM) ranging protocol in the IEEE 802.11–2016 standard (a.k.a 802.11mc standard) [8]. The ability to orchestrate WiFi ranging (distance measurement between two devices through FTM) from user-space on commercial off-the-shelf devices has opened the door for third-party service providers to develop enterprise-grade solutions that leverage pre-deployed infrastructure (e.g., WiFi APs) and end-user devices like smartphones and tablets, to localize both humans and assets in large indoor spaces. Unlike other evolving RF technologies like Ultra Wide Band (UWB) that also offer ranging capability, WiFi devices are omnipresent, have a larger coverage area and better penetrability (lower operating frequency and higher power than UWBs), making WiFi FTM more convenient and practical for enterprise-grade indoor solutions. Further, most modern devices already implement WiFi FTM [1, 10]. Indeed, the WiFi consortium envisions a much larger scope and ecosystem for such location-aware WiFi services under the umbrella of *WiFi-aware* networking [16]. However, we have till date not seen compelling commercial solutions that leverage WiFi FTM for localization. Consequently, it behooves us to understand *whether WiFi FTM is ready for prime-time localization at an enterprise-scale.*

Previous work [23] has conducted experimental studies to understand WiFi FTM in indoor environments. Being preliminary in nature, it has largely focused on its ranging accuracy in limited experimental settings between *homogeneous* WiFi devices (with similar Intel WiFi NICs). To establish FTM's viability in delivering a practical localization solution, we need to understand the *ability of its ranging primitive to inter-operate seamlessly and accurately with heterogeneous devices*, lest there can be no "enterprise-grade" solution. To this end, we present one of the first comprehensive experimental studies that puts the spotlight on *heterogeneity* in multiple dimensions (device, spectrum, environment) in answering this key question.

Our extensive measurement study employs multiple off-the-shelf commercial devices (6 WiFi APs and 3 WiFi clients from different vendors) under different spectrum (ranging bandwidth and channel) and environmental (LoS vs. NLoS; cluttered vs. open indoor settings) conditions. Our key inferences from the study include:

- FTM ranging suffers from fixed (independent of the AP-client distance) range offsets that vary significantly for different AP-client pairs.
- The range offsets for a given AP-client pair can vary substantially across different ranging channels (up to 3.6 m) and channel bandwidths (up to 16.2 m).
- Ranging errors are largely amplified in NLoS conditions. Additionally, unlike in LoS conditions, wider channels in NLoS conditions do not always yield more accurate ranges.

While the presence of fixed offsets in estimated ranges was observed before [23], their reasons were unknown and offsets were addressed through a-priori calibration between a fixed set of devices. However, our extensive study across multiple dimensions has

WiNTECH'22, October 17, 2022, Sydney, NSW, Australia

ACM ISBN 978-1-4503-9527-4/22/10…$15.00
https://doi.org/10.1145/3556564.3558234

exposed the scale and impact of these offsets, while also shedding light on their potential causes. We find that to compensate for the impact of wireless multi-path and direct-path detection delays, device vendors employ proprietary compensation algorithms. Indeed, there is room in the standard for vendors to advertise for such compensation [8]. While a common vendor for the AP and client can have a synergistic compensation algorithm (as seen in our Google AP and Pixel phones with small offsets), this is impossible to address and scale across vendors, resulting in offsets that range from -12m to 14m, resulting in localization errors as large as 20 m. Indeed, this poses a fundamental issue in leveraging FTM for practical localization in enterprises, since these offsets cannot be pre-calibrated and accounted for. This overwhelmingly points to the conclusion that *WiFi FTM is not ready for prime-time enterprise-scale localization.*

Albeit a negative result, we believe the door for leveraging FTM is not completely closed. As a remedy, we propose a simple but effective over-the-top (OTT) auto-calibration solution that allows every WiFi device, when it enters an enterprise environment, to self-calibrate its offsets, rendering WiFi FTM useful for localization. Our approach follows the notion of *simultaneous localization and calibration*[1], whereby the mobility of the client device is used to measure multiple ranges to the AP at different points on the client's trajectory, which are then combined with its own mobility (informed by its inertial sensors) to jointly solve for its own location, while compensating for the unknown offsets. This solution can be easily deployed as an Android or iOS application on client devices, enabling seamless integration.

Our contributions in this work are two-fold:

- We conduct an extensive measurement study of WiFi FTM to assess its practical viability for enterprise-grade localization, with *heterogeneity* across various operational parameters as the focus.
- We propose WiLoc, an OTT auto-calibration solution to remedy the fundamental challenge in inaccurate range estimations, demonstrating that it is still possible to deliver useful localization (median accuracy of ~2m) with WiFi FTM, notwithstanding its large range offsets.

To the best of our knowledge, our work is the first to explore the shortcomings of WiFi FTM in broader enterprise settings, and discuss as well as remedy its implications towards a practical localization solution.

2 BACKGROUND

2.1 802.11mc WiFi Ranging

The IEEE 802.11mc FTM ranging protocol enables two WiFi devices with asynchronous clocks to cooperatively estimate the distance between them by measuring the Round Trip Time (RTT). To measure the RTT, a station (the initiator) and an AP (the responder) exchange bursts of messages with each other (Fig. 1). Both the STA and the AP record the Time-of-Departure (ToD), e.g., $T_1(k), T_3(k)$ in Fig. 1, and Time-of-Arrival (ToA), e.g., $T_2(k), T_4(k)$, for the sent and received FTM messages, respectively. Thereafter, the RTT (for

[1] akin to simultaneous localization and mapping

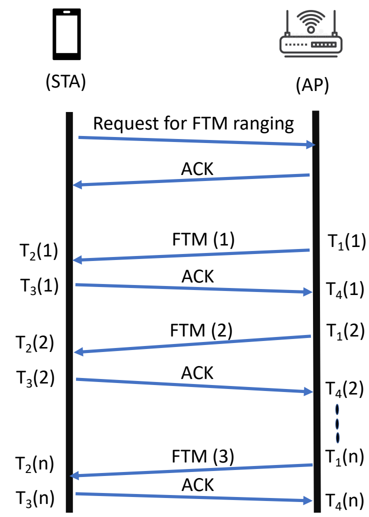


Figure 1: FTM Ranging Protocol Message Sequence

k bursts) is calculated as:

$$RTT = 1/k((\sum_{n=1}^{k} T_4(n) - \sum_{n=1}^{k} T_1(n)) - (\sum_{n=1}^{k} T_3(n) - \sum_{n=1}^{k} T_2(n)))$$

Once RTT is calculated, distance can be computed as $D = 1/2 * RTT * C$, where C is the speed of light.

2.2 Problem of Range-Offsets

The success of the FTM ranging protocol lies in the nodes' ability to accurately record the ToA and ToD. However, in practice, an implementation may capture the ToD earlier or later than when the signal actually arrives at the transmit antenna connector resulting in an offset [8]. Similarly, a significant amount of time can elapse from when the preamble arrives at the receive antenna (actual ToA) to when the node detects the frame, synchronizes with its logical structure, and computes the ToA. The standard requires devices to compensate for this delay in ToA computation by subtracting an offset from the computed ToA. However, the standard does not provide any guidance on how exactly this offset should be determined leading vendors to implement their own proprietary algorithms to correct the ToA/ToD estimations that often leads to over-or-under compensation resulting in either an inflated or a deflated range. The problem is further exacerbated when two different vendors' devices interact, with each device's (incorrect) ToA/ToD estimations (disproportionally) contributing to the eventual range offset.

The 802.11mc frame has specific fields for devices (both STAs and APs) to indicate to their ranging peer any potential error in ToA and ToD estimations (ToA Error, ToD Error fields – see Fig. 2). The standard provides this option to precisely capture a device's confidence in offset-prediction. However, in our measurement study we found that none of the commercial devices actually uses these options.

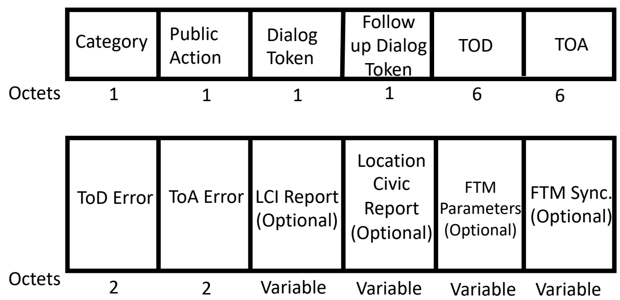


Figure 2: FTM frame format

Interestingly, we observe that the range offsets for a pair of devices is constant for a particular central frequency (channel) and channel bandwidth (e.g., 20/40/80 MHz) combination used during the FTM protocol (§3). We believe that the devices use constant offset values depending on the channel and bandwidth used (to account for the difference in processing overheads in wider channels) to compensate for the incorrect ToA/ToD estimations. To understand the scale of the range offsets problem, we conduct a detailed measurement study with multiple off-the-shelf devices.

3 WIFI FTM RANGING PERFORMANCE

Our measurement study utilizes 6 different WiFi APs – ASUS RT-ACRH13 [3], Linksys Velop [9], Google Nest's Mesh AP system (consisting of 3 APs in a mesh network, which we refer to as Google 1, Google 2 and Google 3) [6] and a Compulab Fitlet AP [4] – and 3 different STAs – Google Pixel 5 [7], Xiaomi Mi Note 10 [12], and a Compulab Fitlet acting as a STA. The WiFi chipset, OS, and the wireless configurations supported by each of these devices for FTM operation are shown in Table 1. The WiFi chipsets employed by each of these devices are different from each other and all devices support channel widths up to 80 MHz. Note that certain vendors (e.g., Google) do not allow manual channel and bandwidth selection. Hence, for these devices, we only use the channel and bandwidth configuration chosen by the device for FTM ranging.

Data Collection. To enable seamless FTM ranging on the Fitlet devices, we modified the Linux OS's *iw* utility using the patch provided in [5]. For the Android smartphones, we wrote an app that uses the WiFi Ranging Android API [2] to send FTM requests periodically (e.g., every 30 ms). We collected data in both LoS and NLoS scenarios, in diverse indoor environments inside an office building and a university campus: a large carpeted area, a lab with benches and equipment, a building lobby with furniture, and a large empty open area with tall glass windows.

3.1 LoS Conditions

We begin by studying the WiFi FTM ranging performance with static STAs in LoS scenarios. For each AP-STA pair, we place a static STA at various distances from 1 m to 50 m, and record the FTM ranges for a channel and bandwidth (channel 36 / 80 MHz) that is supported by all the AP-STA pairs. We physically measure the true distance between the client and the AP for Ground Truth (GT). We use the GT range to calculate the range offsets. Fig. 3 shows the measured ranges (mean and standard deviation across 100 readings) for different STAs connected to the same AP for 3 of our APs. We make two observations: (i) There is a fixed range offset, independent of the distance between the AP and the STA, confirming the finding from [23]. (ii) Range offsets vary significantly for different AP-client pairs, extending the results from [23], thus showing how extreme and disparate the offset problem is across heterogeneous devices.

Impact of Operating Frequency. We next study how range offsets vary based on the operating channel (for a fixed channel bandwidth). Fig. 4a shows the range offsets measured for the Pixel 5 client when ranging with all our APs across all possible (20 MHz bandwidth) channels (we show results for only one client due to space limitations). We observe that devices implement a frequency dependent offset (for ToA/ToD correction). However, the variance in these offsets for a given AP-client pair is not very large; the maximum offset difference across different channels, observed with a Xiaomi client and ASUS AP, is 3.6 m.

Impact of Channel Bandwidth. Next, we show how range offsets vary for different channel bandwidths (same central frequency), for a given AP-STA pair. Fig. 4 shows the range offsets across 20 MHz, 40 MHz, and 80 MHz channel bandwidths, for the Pixel 5 client when ranging with all our APs. The range offsets for the same pair of devices can vary by as much as 16.2 m (for ASUS AP-Fitlet Client) or as little as 1.2 m (for Fitlet AP-Pixel 5 Client) as shown in Fig. 4.

Accuracy of FTM ranges with offset correction. Lastly, we show the accuracy of the FTM ranges, with (manual) offset correction. For the data previously collected, we estimate the offset as the mean difference between the measured range and the GT range and use that as an offset value. Subtracting this offset from the measured range gives the range error. We observe that across device pairs, in the median case, the range errors are ~50 cm, ~30 cm, and ~10 cm, suggesting that FTM ranges in LoS are highly accurate and can definitely aid in accurate localization, provided that the disparate range offsets are resolved.

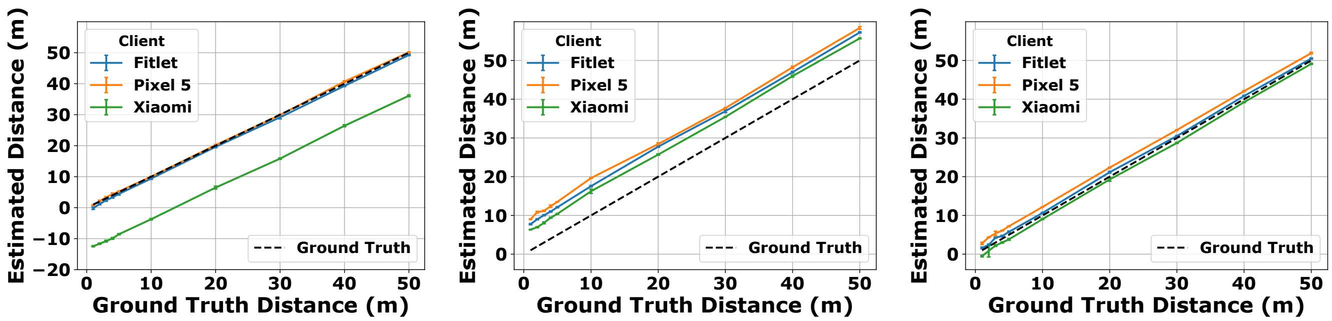
3.2 NLoS Conditions

We now study the FTM ranging accuracy in NLoS conditions. To determine the FTM ranging errors, we (manually) discount the range offset, which we calculated from the LoS measurements, from the measured range, and estimate the accuracy of the measured ranges. Fig. 5 plots the CDF of the ranging errors in NLoS for different AP-STA combinations. Unlike in LoS, the range errors (for a given channel and bandwidth) are amplified in NLoS conditions. The drop in channel quality, combined by the devices' inability to exactly identify the Channel Impulse Response peaks (a common problem for multipath signals) further degrades the ToA estimations leading to large range errors. The 80 MHz channel bandwidth, which had the most accurate range estimations in LoS, suffers the most in NLoS due to fact that a signal using wider bandwidth (e.g., 80 MHz) attenuates faster than the same signal spread over a narrower bandwidth (e.g., 20 MHz). More importantly, we conclude that *for localization solutions implementing FTM ranging, a wider channel does not always yield accurate ranges.*

LoS vs. NLoS FTM ranging – Impact on Localization: Given the high inaccuracy of NLoS range estimations, any localization

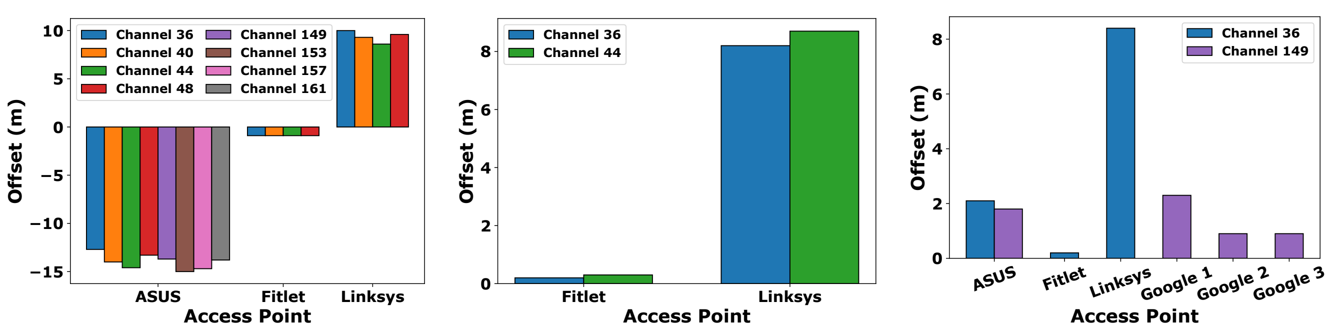
Table 1: Device Description.

Device	Role	Chipset	Operating System	Channel Widths Supported	5 GHz Channels Supported
ASUS RT-ACRH13	AP	Qualcomm IPQ4018	-	20/80 MHz	36-161
Linksys Velop WHW03	AP	Qualcomm QCA9986	-	20/40/80 MHz	36-48
Google Nest WiFi	AP	Qualcomm QCS405 (Router) Qualcomm QCS404 (Point)	-	80 MHz	149
Compulab Fitlet	AP/STA	Intel Wireless-AC 8260	Linux Mint 18	20/40/80 MHz	36-48
Google Pixel 5	STA	Qualcomm Snapdragon 765G	Android 11	20/40/80 MHz	36-161
Xiaomi Mi 10 Note	STA	Qualcomm Snapdragon 730G	Android 11	20/40/80 MHz	36-161



(a) Fitlet AP. (b) Linksys AP. (c) ASUS AP.

Figure 3: Ranging across distances with different AP-client pairs.



(a) Pixel 5 Client, 20 MHz Bandwidth. (b) Pixel 5 Client, 40 MHz Bandwidth. (c) Pixel 5 Client, 80 MHz Bandwidth.

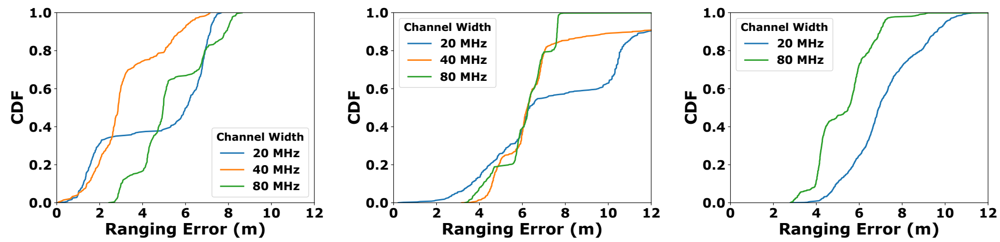
Figure 4: Offset changes with varying channel.

system that uses FTM ranges needs to be able to distinguish LoS ranges from NLoS ranges, so that it can filter out the NLoS ranges.

To make that classification, we employ a simple machine learning model, Random Forest, and using 3 inputs – last n ranges, last n RSSI values, and variance in last n ranges – we try to accurately classify whether the ranges are with an AP in LoS or NLoS. Table 2 shows the classification accuracy for different n values and time interval t between consecutive range and RSSI samples. Even using only 5 range samples obtained every 30 ms, the model can correctly classify LoS and NLoS ranges in 83.5% of the cases. Furthermore, assuming the user stays in LoS/NLoS for longer, taking a larger number of range/RSSI samples spaced further apart leads to even higher accuracy, up to 99.5% when considering the previous 20 range/RSSI values obtained every second.

Table 2: LoS-NLoS Classification Accuracy.

	t = 30 ms	t = 200 ms	t = 500 ms	t = 1000 ms
n = 5	83.5%	84.9%	86.4%	88.8%
n = 10	84.7%	86.7%	90.3%	94.6%
n = 20	85.6%	89.7%	95.5%	99.5%



(a) Fitlet AP, Pixel 5 Client. (b) Linksys AP, Xiaomi Client. (c) ASUS AP, Fitlet Client.

Figure 5: Ranging error in NLoS conditions.

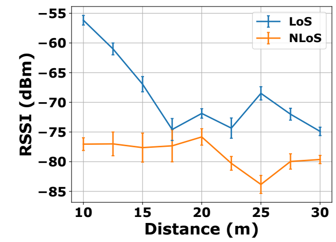


Figure 6: RSSI in LoS vs. NLoS.

3.3 Summary

To summarize, we find the range offsets depends on 1) the AP-STA pair, 2) the channel width, and 3) the channel (frequency). Given the plethora of different vendors for WiFi devices and the fact that the devices' spectrum configurations change frequently, it becomes practically impossible to have an offline solution to this problem. The only way to address this challenge is via an on-demand, real-time, online solution.

In addition, we see that the ranging accuracy deteriorates significantly as the user moves from LoS to NLoS conditions. Thus, it becomes imperative to design a system that can distinguish between LoS and NLoS ranges, so that it can filter out the NLoS ranges and use only the LoS ranges.

4 *WILOC* DESIGN

To address the range offset problem in a practical manner, we propose *WiLoc*, a novel OTT solution that can be deployed as an application on end-user devices like smartphones and enables these devices to self-calibrate their range offsets with multiple APs when they enter an enterprise environment, without any human input. At a high level, *WiLoc* follows the notion of simultaneous localization and calibration. It employs Euclidean geometry leveraging the user's mobility – computed using onboard inertial sensors – to measure the ranges at various points of its trajectory and combines both trajectory and range information to solve for the range offset.

WiLoc assumes an AP to be statically placed at (x_{AP}, y_{AP}), (as shown in Fig. 7), and the user to start at (x_C, y_C) and move continuously in a random direction with respect to the AP. At each time-interval t (e.g., t=1 s), the STA performs FTM ranging to measure a range R_p, where $R_p = r_p + \delta$ with δ being the range offset. To solve for δ, *WiLoc* employs two approaches – $WiLoc_{lin}$ and $WiLoc_{quad}$. While $WiLoc_{lin}$ uses a system of linear equations to solve for δ, $WiLoc_{quad}$ employs a quadratic solver. The linear approach is less complex, but results in a less accurate estimation of δ, while the quadratic approach is more complex, but offers more accurate estimation (§5).

4.1 $WiLoc_{lin}$: A Linear Solver

The first approach, $WiLoc_{lin}$, formulates the problem of solving for δ using a series of linear equations. $WiLoc_{lin}$ assumes (x_{AP}, y_{AP}) = (0,0) for the AP location. Given the STA's initial position (x_C, y_C),

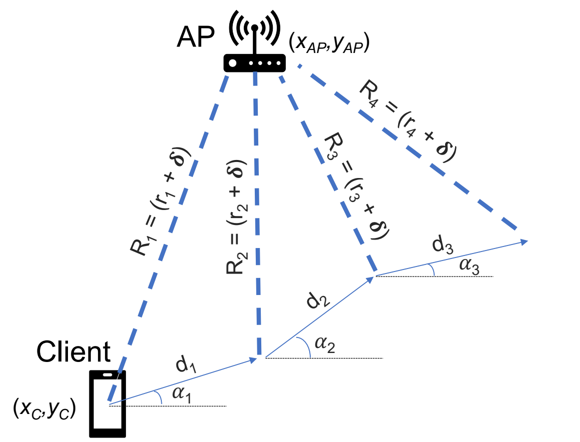


Figure 7: *WiLoc* Scenario.

we have:

$$x_C^2 + y_C^2 = (R_1 - \delta)^2 \tag{1}$$

As the user moves, at each time interval t, the STA reaches a new position p, resulting in a an additional equation:

$$\left(x_C + \sum_{i=1}^{T} d_i cos(\alpha_i)\right)^2 + \left(y_C + \sum_{i=1}^{T} d_i sin(\alpha_i)\right)^2 = (R_p - \delta)^2 \tag{2}$$

where, α_i is computed by measuring the heading (magnetometer) information, and d_i (displacement) is estimated via the STA's inertial sensors (accelerometer, magnetometer).

Now, for each new equation (2) obtained at every time interval t, we subtract equation (1) from it. This linearizes the system of equations with 3 unknown variables (x_C, y_C, and δ). Solving for the 3 unknowns requires at-least 3 equations (3 seconds if t = 1), with additional equations increasing the accuracy of the solution (δ).

4.2 $WiLoc_{quad}$: A Quadratic Solver

The second approach, $WiLoc_{quad}$, employs a quadratic solver.

$WiLoc_{quad}$ assumes (x_C, y_C) = (0,0) for the STA's initial position and given the AP is at (x_{AP}, y_{AP}), the Euclidean distance between the AP and STA is:

$$\sqrt{x_{AP}^2 + y_{AP}^2} = R_1 - \delta \tag{3}$$

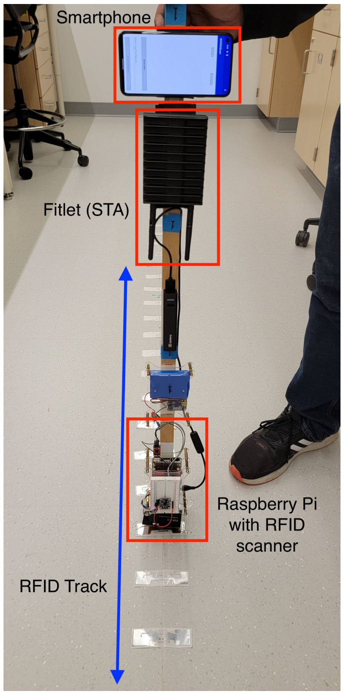


Figure 8: Experiment Setup.

At each time t, with the user at position p, the new Euclidean distance is:

$$\sqrt{\left(x_{AP} - \sum_{i=1}^{T} d_i cos(\alpha_i)\right)^2 + \left(y_{AP} - \sum_{i=1}^{T} d_i sin(\alpha_i)\right)^2} = R_p - \delta \quad (4)$$

For each new equation (4) we obtain at every time interval t, we subtract equation (3) from it (i.e., (4)-(3)). This cancels out δ leaving 2 unknowns, i.e., x_{AP} and y_{AP}. $WiLoc_{quad}$ requires only 2 equations to solve for the unknowns, with additional equations increasing the accuracy of the solution.

Once $WiLoc_{quad}$ solves for x_{AP} and y_{AP}, the range-offset δ is given by:

$$\delta = R_1 - \sqrt{x_{AP}^2 + y_{AP}^2} \quad (5)$$

where R_1 is the range when user (STA) was at (0,0).

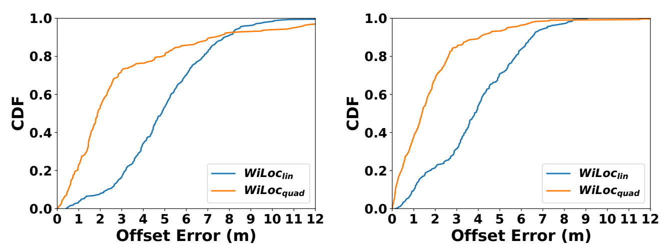
4.3 *WiLoc*: Localization

WiLoc schedules a STA to sequentially range with multiple APs (at least 3) at each time interval t, and solves for δ_{AP} to compute the true range from each AP. LoS APs are selected by the Random Forest classifier discussed in §3.2. Subsequently, *WiLoc* employs a simple least-square multilateration solver to localize the STA, with the offset-corrected ranges from LoS APs as input to the solver.

5 IMPLEMENTATION AND EVALUATION

We conduct a two-part evaluation of the proposed WiLoc solution. In the first part, we run a trace-based evaluation of *WiLoc*'s ability to correctly deduce the range offsets in practical indoor environments. Next, we employ *WiLoc*'s range resolution to localize (multilaterate) a user in real-time and evaluate the accuracy of the localization with respect to the Ground Truth (GT).

Ground Truth and Data Collection: To obtain the Ground Truth location during user mobility, we deploy pre-designed tracks (of different trajectories) of RFID tags 10 cm apart. Each RFID tag's location is pre-recorded manually with respect to the WiFi AP locations. We built a contraption (shown in Fig. 8), which has an RFID scanner at the bottom (connected to a Raspberry Pi) and the smartphone that implements *WiLoc* is attached to the stick. We time-sync the smartphone and the Raspberry Pi using a local NTP server, ensuring no time drift occurs between the two devices. A user walks along the tracks holding the stick in hand such that the RFID scanner is right above the RFID tag. As the RFID scanner moves on top of the tag, it records the RFID's unique EPC ID along with its scanned time. Meanwhile, the smartphone runs the FTM ranging protocol, sequentially ranging with each of the 6 APs, recording the measured range and time of measurement. In addition, the smartphone records its inertial data (for Fitlet STA, we use an external IMU) with a timestamp. Since both devices are time-synced, for each measured range (and inertial data), we get a corresponding RFID location. Furthermore, we interpolate the scanned RFID locations to get a real-time ground truth user location with errors < 10 cm.

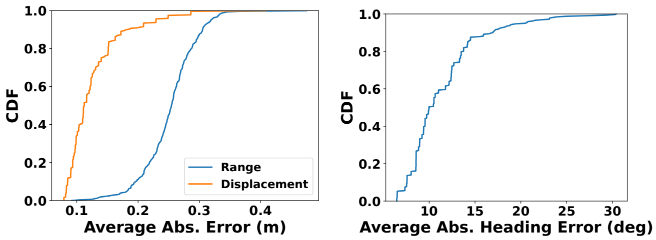


(a) Measured Inertial data. (b) GT Inertial data.

Figure 9: Offset Resolution Error.

To evaluate *WiLoc*, we collect data involving user mobility at diverse indoor locations with narrow corridors, large open spaces, and locations with a lot of clutter (NLoS). During each run, we place the 6 WiFi APs at random locations, while recording their locations (in the Euclidean space) with respect to the GT's starting point. We record data with the user walking at normal pace (1-2m/s). We restrict *WiLoc*'s range offset resolution evaluation to only LoS AP-STA data due to the highly erroneous NLoS ranges which need to be filtered (§3.2).

***WiLoc*: Range Offset Resolution:** We test the efficacy of both versions of *WiLoc* – $WiLoc_{lin}$ and $WiLoc_{quad}$ – in resolving the range offsets, using the data we collected. Due to space constraints, we show results of evaluations run with data collected for 80 MHz bandwidth only. Fig. 9a plots the CDF of the offset resolution error across all the environments. For $WiLoc_{lin}$, we observe that the offset resolution error is 4.45 m in the median case. On the other hand, $WiLoc_{quad}$, as expected, with a better least-square approximation has a median resolution error of 1.81 m.



(a) Range and Displacement Errors.

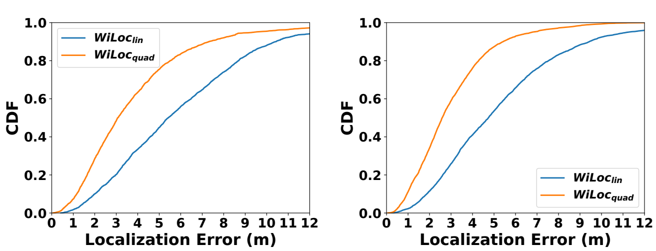
(b) Heading Error.

Figure 10: Errors observed in our evaluation dataset.

Impact of Data Errors: Data errors (range and inertial errors) impact the ability of *WiLoc* solvers to accurately determine the range offset. Fig. 10 plots the range, displacement, and heading errors that we observe in our dataset. The median range and displacement errors are ~25 cm and ~12 cm, respectively, and the median heading error (magnetometer) is ~10°.

While the range errors are inherent to the workings of the FTM ranging protocol and unavoidable, the errors in inertial estimates are mainly due to implementation and environment. Improving inertial estimations is out of the scope of this work and there already exist other works that propose ways to improve these estimations [17]. However, in order to analyze *WiLoc* performance in isolation, without the influence of incorrect inertial estimations, we run $WiLoc_{lin}$ and $WiLoc_{quad}$ using the measured ranges but with GT displacement and heading information. Fig. 9b shows that in this case, the range offset resolution error improves to 3.62 m and 1.22 m for $WiLoc_{lin}$ and $WiLoc_{quad}$ respectively.

Localization Accuracy: To understand the localization accuracy one can expect with the offset-free ranges (provided by *WiLoc*), we implement a least-squares multilateration solver to continuously localize the user. The CDF in Fig. 11 shows the localization errors when the measured inertial data are used for offset resolution. We observe the median localization error to be 5.42 m for ranges resolved by $WiLoc_{lin}$ and 2.91 m for ranges resolved by $WiLoc_{quad}$. On the other hand, if we assume a better inertial estimation in-play (i.e., assuming GT inertial data), the localization errors for $WiLoc_{lin}$ and $WiLoc_{quad}$ resolved ranges are shown in Fig. 11a In this case, we observe that the median localization error drops to 4.53 m for $WiLoc_{lin}$ and 2.28 m for $WiLoc_{quad}$ respectively.



(a) Measured Inertial data.

(b) GT Inertial data.

Figure 11: Localization Error.

6 DISCUSSION AND FUTURE WORK

Localization using WiFi FTM ranging can leverage existing ubiquitous WiFi infrastructure, and thus is more viable to be deployed at-scale compared to other localization systems that use additional hardware, such as ultra-wideband (UWB) anchors [21, 28, 29], Bluetooth Low Energy (BLE) beacons [19, 37], and video cameras [18, 22]. However, using the WiFi spectrum for localization comes at a cost, as APs cannot transmit data frames while responding to FTM ranging requests. Hence, it is imperative to find the right balance between using WiFi spectrum for localization and communication. One potential solution is to opportunistically schedule ranging on low-traffic channels. Another approach would be to leverage P2P (e.g., WiFi Aware [11, 16]) FTM ranging and use the WiFi APs selectively to reduce their communication overhead. Each of these approaches is promising but poses several challenges.

7 RELATED WORK

802.11mc FTM Ranging. The work in [23] conducted the first study of 802.11mc Fine Time Measurement and evaluated its performance in various scenarios with a limited set of devices. This study also observed the range offset problem and addressed it a-priori calibration between a fixed set of devices. In this work, we expand on this study by considering a much larger set of AP and client devices (including smartphones) and showing that this range offset problem exists at a wide scale and poses a fundamental issue in leveraging FTM for practical localization in enterprises.

Further, while previous works [15, 24, 31, 33] have proposed localization solutions based on WiFi FTM ranging, they largely ignore these range offsets or simply evaluate their solutions over a single set of devices, after manually removing the offsets. Such a manual calibration cannot work in practice, as different users may have different devices resulting in largely diverse offsets. In contrast to these works, we propose a simple yet effective OTT auto-calibration solution to the range offset problem that has the potential to enable useful localization in real-world enterprise environments.

WiFi-based Localization. There has been a large body of work on WiFi-based localization, e.g., [13, 14, 20, 25–27, 30, 32, 34–36]. Several works [13, 26, 27, 32, 35] leverage RSSI values, as they are easily available in APs and client devices. However, RSSI-based approaches require extensive fingerprinting of the environment before being deployed. The works in [25, 34] utilize CSI from a commodity 802.11n chipset to enable accurate indoor localization without the need for labor-intensive fingerprinting. However, none of today's WiFi chipsets makes that information readily available. On the other hand, the FTM protocol is standardized [8] and is already available in several commercial devices.

8 CONCLUSION

In this paper, we conducted an extensive experimental study, employing multiple commercial WiFi devices under different spectrum and environmental conditions, that aimed to verify *whether WiFi's FTM is indeed ready for prime-time localization.* The most critical finding of our study is the presence of substantial offsets in estimated ranges between heterogeneous devices and configurations that significantly degrade localization. Albeit a negative result for

FTM's readiness, we also proposed *WiLoc*– a simple but promising OTT auto-calibration solution that allows every WiFi device, when it enters an enterprise environment, to self-calibrate its offsets on-demand. Via *WiLoc*, we demonstrated that it is still possible to deliver useful localization (median accuracy of ≈2m) with WiFi FTM,despite its large range offsets.

REFERENCES

[1] [Online]. Android Developer Guide. https://developer.android.com/guide/topics/connectivity/wifi-rtt

[2] [Online]. Android WiFi RTT API. https://developer.android.com/guide/topics/connectivity/wifi-rtt

[3] [Online]. ASUS RT-ACRH13 Router. https://www.asus.com/supportonly/RT-ACRH13/HelpDesk/

[4] [Online]. Compulab fitlet2. https://fit-iot.com/web/products/fitlet2/

[5] [Online]. FTM patch for the Linux iw utility. https://goo.gl/TzJRGG

[6] [Online]. Google Nest WiFi. https://store.google.com/us/product/nest_wifi?hl=en-US

[7] [Online]. Google Pixel 5. https://www.gsmarena.com/google_pixel_5-10386.php

[8] [Online]. IEEE Standard for Information Technology - 802.11-2020. https://standards.ieee.org/ieee/802.11/5536/

[9] [Online]. Linksys Velop Intelligent Mesh WiFi System. https://www.linksys.com/us/whole-home-mesh-wifi/velop-intelligent-mesh-wifi-system-tri-band-ac2200/p/p-whw0301/

[10] [Online]. Wi-Fi access points support the IEEE 802.11mc FTM RTT. https://people.csail.mit.edu/bkph/ftmrtt_issues

[11] [Online]. Wi-Fi Aware Android API. https://developer.android.com/guide/topics/connectivity/wifi-aware

[12] [Online]. Xiaomi Mi Note 10. https://www.mi.com/global/mi-note-10/

[13] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. 2009. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proc. of ACM MobiCom*.

[14] P. Bahl and V.N. Padmanabhan. 2000. RADAR: an in-building RF-based user location and tracking system. In *Proc. of IEEE INFOCOM*.

[15] Leor Banin, Ofer Bar-Shalom, Nir Dvorecki, and Yuval Amizur. 2019. Scalable Wi-Fi Client Self-Positioning Using Cooperative FTM-Sensors. *IEEE Transactions on Instrumentation and Measurement* (2019).

[16] Daniel Camps-Mur, Eduard Garcia-Villegas, Elena Lopez-Aguilera, Paulo Loureiro, Paul Lambert, and Ali Raissinia. 2015. Enabling always on service discovery: Wifi neighbor awareness networking. *IEEE Wireless Communications* (2015).

[17] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. 2021. ITrackU: Tracking a Pen-like Instrument via UWB-IMU Fusion. In *Proc. of ACM MobiSys*.

[18] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. 2019. Selective Sensor Fusion for Neural Visual-Inertial Odometry. In *Proc. of IEEE/CVF CVPR*.

[19] Dongyao Chen, Kang G. Shin, Yurong Jiang, and Kyu-Han Kim. 2017. Locating and Tracking BLE Beacons with Smartphones. In *Proc. of ACM CoNEXT*.

[20] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. 2010. Indoor Localization without the Pain. In *Proc. of ACM MobiCom*.

[21] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. 2019. TrackIO: Tracking First Responders Inside-Out. In *Proc. of USENIX NSDI*.

[22] Zakieh S. Hashemifar, Charuvahan Adhivarahan, Anand Balakrishnan, and Karthik Dantu. 2019. Augmenting Visual SLAM with Wi-Fi Sensing for Indoor Applications. *Auton. Robots* (2019).

[23] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. 2018. Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform. In *Proc. of ACM MobiCom*.

[24] Mohamed Ibrahim, Ali Rostami, Bo Yu, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Fan Bai, and Richard Howard. 2020. Wi-Go: Accurate and Scalable Vehicle Positioning Using WiFi Fine Timing Measurement. In *Proc. of ACM MobiSys*.

[25] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. In *Proc. of ACM SIGCOMM*.

[26] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. 2012. Push the Limit of WiFi Based Localization for Smartphones. In *Proc. of ACM MobiCom*.

[27] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, and Venkata N. Padmanabhan. 2012. Centaur: Locating Devices in an Office Environment. In *Proc. of ACM MobiCom*.

[28] Daniel Neuhold, Christian Bettstetter, and Andreas F. Molisch. 2019. HiPR: High-Precision UWB Ranging for Sensor Networks. In *Proc. of ACM MSWiM*.

[29] Pat Pannuto, Benjamin Kempke, Li-Xuan Chuo, David Blaauw, and Prabal Dutta. 2018. Harmonium: Ultra Wideband Pulse Generation with Bandstitched Recovery for Fast, Accurate, and Robust Indoor Localization. *ACM Trans. Sen. Netw.* (2018).

[30] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-Effort Crowdsourcing for Indoor Localization. In *Proc. of ACM MobiCom*.

[31] Maurizio Rea, Domenico Giustiniano, and Joerg Widmer. 2020. Virtual Inertial Sensors with Fine Time Measurements. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*.

[32] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information. In *Proc. of ACM MobiSys*.

[33] Wenhua Shao, Haiyong Luo, Fang Zhao, Hui Tian, Shuo Yan, and Antonino Crivello. 2020. Accurate Indoor Positioning Using Temporal-Spatial Constraints Based on Wi-Fi Fine Time Measurements. *IEEE Internet of Things Journal* (2020).

[34] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point. In *Proc. of USENIX NSDI*.

[35] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No Need to War-Drive: Unsupervised Indoor Localization. In *Proc. of MobiSys 2012*.

[36] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *Proc. of USENIX NSDI*.

[37] Yuan Zhuang, Jun Yang, You Li, Longning Qi, and Naser El-Sheimy. 2016. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. *Sensors* (2016).