Off-the-shelf Wi-Fi Indoor Smartphone Localization

Hongyu Jin
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
hongyuj@kth.se

Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

***Abstract*—Recently released Wi-Fi adapters, such as Intel AX200 802.11ax NIC, support both Channel State Information (CSI) measurement and Fine Time Measurement (FTM). Angle of Arrival (AoA) estimation with CSI, using MUltiple SIgnal Classification (MUSIC), and FTM are both promising localization methods. But each suffers from practical constraints pertinent to the specific hardware and firmware used. The result can be rather inaccurate localization if AoA or FTM alone were used. We identify the issues/challenges specific to AX200, and as a remedy, we propose a localization approach that combines both CSI-based AoA and FTM. Our approach does not require any modification of the localization target device. This makes the solution readily available for localizing smartphones or any Wi-Fi devices with FTM functionality. Our experimental evaluation shows that our approach achieves a successful localization ratio of 80%, with localization error less than $1\ m$; and less than $0.5\ m$ for 66% of the experiments.**

***Index Terms*—CSI calibration, FTM, MUSIC, Android**

I. Introduction

Indoor localization of mobile devices has been actively studied over the past decades. Universally available Wi-Fi equipment and infrastructure render Wi-Fi indoor localization an attractive, highly available option. Localization based on Channel State Information (CSI), measured from Wi-Fi frames, is a popular approach with high accuracy and usability. Prior work leverages mainly Intel 5300 NIC for CSI measurement [1]–[3]. However, CSI measurements with Intel 5300 are only available for the 802.11n version of the standard, with no support for the latest 802.11ac or 802.11ax. Intel has recently added CSI measurement capability to the latest Intel 9260 and AX200 NICs. Their CSI-based localization capabilities were recently investigated [4], [5].

At the same time, Intel 8260, 9260 and AX200 support the latest Wi-Fi based distance measurement technique, Fine Time Measurement (FTM), which leverages Round-trip Time (RTT) measurements for Wi-Fi signals. However, constant distance offsets might exist in the FTM results [6], which lead to constantly shifted distance values. Although offset compensation could solve the issue, achieving this assumes known FTM initiators; while the offset calibration requires the involvement of both the transmitting (TX) and receiving (RX) devices.

In this paper, we investigate how to overcome the limitations of each of the CSI- and FTM-based localization approaches, working with the latest Intel AX200 that supports both CSI and FTM. We calibrate the CSI phase offset with an approach similar to that used in [1], [5], [7], [8], and report observations specific to the Intel AX200. We propose a localization approach that leverages both Angle of Arrival (AoA) estimation using MUltiple SIgnal Classification (MUSIC) [1], [9] and distance measurements with FTM. Our approach works with two or more static localization reference points (devices) that support both CSI and FTM, and it localizes targets based on triangulation. AoAs of signals transmitted from a localization target are estimated by the reference points, and the signal RTTs are measured at the target. Unlike several indoor localization systems that customize the localization target devices [1]–[3], [7], our approach works for off-the-shelf localization targets, smartphones or any Wi-Fi device, with FTM functionality, without any additional hardware or firmware modification.

II. Problem Statement

Different Phase Locked Loop (PLL) initial phases introduce phase offsets among the RF chains, making offset correction necessary before the CSI phases can be used by localization applications [5], [8]. In addition to the offset by PLLs, we observed a (randomly added) additional π offset for AX200, similarly to that observed for Intel 5300 [8], which leads to two offset variants. However, unlike the Intel 5300 observations, whose offset locks on to one of the variants once the device boots (and it changes randomly after the system reboots), we observed both variants without any system boot or driver reload; which makes it even harder to figure out the correct phase offset.

CSI phases measured at different antennas reflect the difference of Times of Flight (ToFs) of a transmitted signal. This can be exploited by the MUSIC algorithm to estimate signal AoAs [1], [7]. Let the relative distance between the two RX antennas be d. Then, the phase shift/rotation introduced at the second antenna, relative to the first antenna, is $\varphi = -2 \times \pi \times d \times sin(\theta) \times f/c$, where θ is the AoA of the signal, f is the signal frequency, and c is the speed of light. However, due to the observed signal phase wrapped to a range of $[-\pi, \pi]$ (i.e., a phase value increasing beyond π is wrapped to $-\pi$, and vice versa), φ calculated based on the observed CSI phases might not be the true phase difference. The phase of a signal that traveled a longer distance, thus corresponding to a longer ToF, could result in lower phase values due to the

wrapped signal phases. An RX antenna array with more than two antennas (e.g., an Intel 5300 NIC that has three antennas) could eliminate such ambiguity by considering phase shifts between different pairs of antennas [1], [7]. However, given the limitation of two antennas available on Intel AX200 NICs, estimating the true AoA based on a single device is challenging[1].

FTM is another recent localization technique (standardized with IEEE 802.11mc), supported by the latest Android devices [10] and Intel Wi-Fi NICs [6]. It estimates distances by calculating RTTs between the TX and RX devices. Offset correction is necessary before the obtained RTT values can be used for distance estimation [6], given the observed constant offset. However, unlike the phase offset that can be compensated with a one-time calibration of the RX, the RTT offsets depend on both the TX and the RX [6]; thus, offset correction is unrealistic because the FTM initiators can be unknown devices.

In this paper, we address these challenges by combining two localization approaches, CSI MUSIC AoA estimation and FTM RTT distance estimation. Unlike other works that rely on CSI measurements on the localization target [1], [2], our approach works with off-the-shelf Android smartphones that support FTM. This a significant advantage for easy deployment and fast, low-cost adoption of a solution that is effective and efficient.

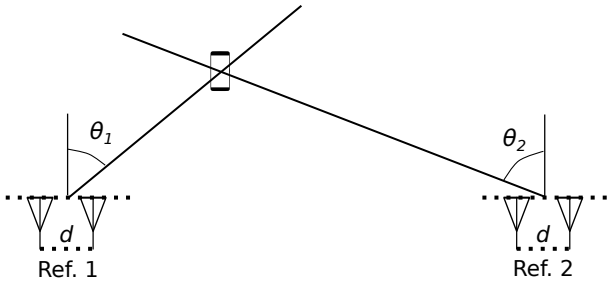


Fig. 1: Localization based on two reference points.

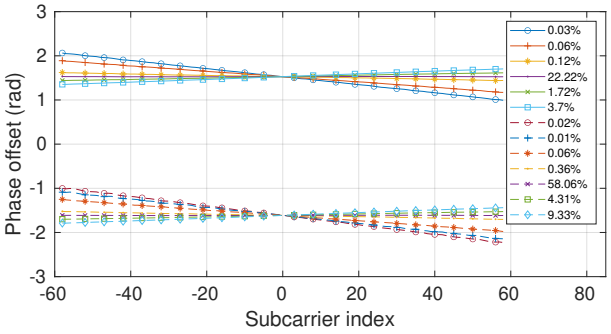


Fig. 2: Variants of phase offsets with distinct slopes. Each line represents average phase offset for the variant. The legend shows the percentages of the packets that fall into each variant.

III. Our Approach

We measure CSI phase offsets on an RX device and feed corrected CSI phases to the MUSIC algorithm for AoA estimation. We localize the target based on triangulation with AoAs measured at multiple reference points. The AoA ambiguity is then eliminated by comparing with non-calibrated FTM results. Fig. 1 shows an example setup for our approach with two reference points. In this paper, we use the center frequency of 5.18 GHz with 40 MHz bandwidth and CSI for 114 subcarriers (with indices -58:-2 and 2:58) is reported.

A. Calibrating CSI Phases

We first calibrate the phase offset between the two antennas for each reference point (i.e., CSI collector), with the technique used in [1], [5], [7]. Intuitively, when the antennas are connected to a signal splitter, the CSI phases measured at the two antennas should be, ideally, identical. However, due to imperfect circuits and phase offsets introduced by external cables, the measured CSI phases have offsets, between the antennas across the subcarriers, which need to be compensated. As shown in Fig. 2, we observed 13 variants of phase offset within the 10^4 CSI matrices, divided into two batches. The difference between the positive and the negative offset variants is exactly π. We observed an average offset of around 1.5 (or around $\pi/2$)[2]. Each variant is an average offset for a batch of packets with similar offset slope over the subcarriers. The different slopes indicate offset variations between the RF chains across packets. Not knowing which of the two batches is correct, we simply compensate a π offset for one of the batches. Then, when we apply the MUSIC algorithm to the CSI phases, we simply run the algorithm twice, by adding, in addition to the phase offset, a zero and a π offset respectively. Instead of averaging over all packets, we consider the offset variant that has the highest packet ratio. As the two variants with 22.22% and 58.06% have the same slope value, we consider the offset calculated from the aggregate 80.28% of packets as the compensated offset between the antennas. We also compensated a $+90^\circ$ shift for the subcarriers with positive indices in the 40 MHz band [11].

B. AoA Estimation with MUSIC

Before we apply the MUSIC algorithm, we remove slopes (i.e., phase rotations over time introduced by ToF and Packet Detection Delay (PDD)) of the CSI phases, similarly to SpotFi [1]. Instead of obtaining the best linear fit of the unwrapped CSI phases [1], we remove the slope gradually (25 ns for each step), until it is close to zero. This helped us observe an additional time offset of around 2800 ns for the second spatial stream (for all packets), given significantly different CSI phases for the two spatial streams. Fig. 3 shows CSI

[1]For example, given a true AoA of 36°, a center frequency of 5.18 GHz, and an RX antenna distance of 10 cm in our experiments, the MUSIC algorithm returned -34°, -2°, and 37° as the candidate AoAs. The corresponding φ values are around 6.07, -0.38, and -6.52 radians, with a roughly 2π spacing.

[2]Similar offsets observed after exchanging the cables for two RX antennas, indicating the offset introduced by external cables is close to zero.

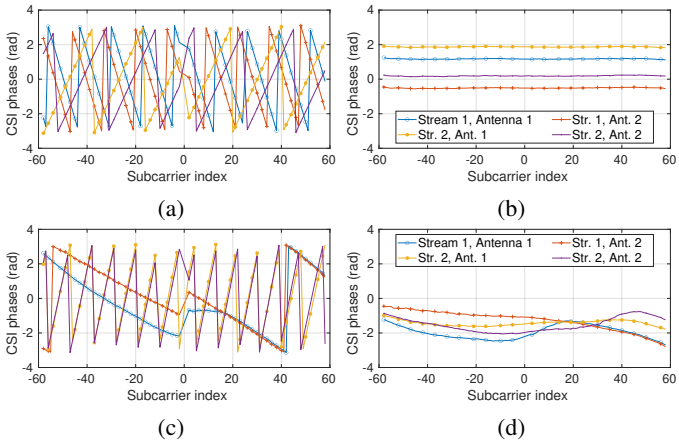


Fig. 3: CSI phases for two packets (before offset compensation) reported for two spatial streams, before ((a), (c)) and after ((b), (d)) the ToF and PDD removal (around 250 ns, 3050 ns, 50 ns, and 2850 ns respectively); with cables and splitters ((a), (b)) and antennas ((c), (d)).

phases before and after the removal of ToF and PDD. After the ToF and PDD removal, the CSI phase lines are almost flat.

Unlike Intel 5300, which returns CSI (available only for 802.11n) for 30 out of 56 subcarriers (20 MHz) or 114 subcarriers (40 MHz), Intel AX200 returns CSI for all available subcarriers for all available standards (802.11 a/b/g/n/ac/ax; please see Table 9-58 and 9-76 in [11] for the reported CSI subcarrier indices). We leverage all available subcarriers for better localization accuracy. We apply the MUSIC algorithm on a smoothed CSI matrix [1], constructed based on subcarriers with identical frequency spacing. In SpotFi [1], 30 subcarriers between the subcarriers -58 and 58 (for the 40 MHz mode) were used, with subcarrier (index) spacing of 4. Intel AX200 reports 114 subcarriers (-58:-2, and 2:58), while six pilot subcarriers (i.e., -53, -25, -11, 11, 25, and 53) are not properly reported. Therefore, we interpolate the CSI phases at the 9 missing subcarriers based on the available 108 subcarriers, and use the 117 subcarriers (between -58 and 58 with an identical index spacing of 1) for the MUSIC algorithm.

C. Localization Algorithm

Our algorithm exploits the fact that, although offsets exist in the FTM results, calculating the difference between the FTM results with two (or more) FTM responders (equipped with identical NIC models) would cancel out the offset and reflect the true distance difference. Let d_1 and d_2 denote true distances to two FTM responders, and $\hat{d}_1 = d_1 + \delta$ and $\hat{d}_2 = d_2 + \delta$ denote measured distances, where δ is the offset in FTM; then, we can derive $d_1 - d_2 = \hat{d}_1 - \hat{d}_2$.

Algorithm 1 shows our localization algorithm pseudocode. We first estimate AoAs at the reference points using the MUSIC algorithm. There will be multiple candidate AoAs for a given localization target at each reference point. Then, we estimate possible coordinates of the target based on the candidate AoAs using triangulation. For each candidate position,

Algorithm 1: Localization using CSI and FTM

Input: csi_matrix_set: The measured CSI matrix for a set of packets by two reference devices. d_{ftm_1}, d_{ftm_2}: Distances measured, using FTM, to two reference devices.

Output: $coord$: The resultant coordinate of the localization target.

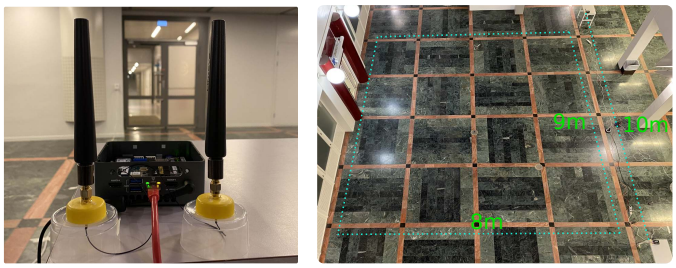
1 Estimate possible AoAs with csi_matrix_set using MUSIC algorithm;
2 Estimate possible coordinates, $coord\_set$, of the target using triangulation with AoA estimations;
3 $ftm\_diff = d_{ftm_1} - d_{ftm_2}$;
4 $coord = null$;
5 **foreach** $coord\_candidate$ *in* $coord\_set$ **do**
6 Calculate d'_1 and d'_2 to two reference points from $coord\_candidate$;
7 Calculate the similarity between the $coord\_candidate$ and the FTM results, $delta' = |d'_1 - d'_2 - ftm\_diff|$;
  /* The candidate with lower $delta'$ value is ranked higher. */
8 **return** *Ranked* $coord\_set$ *based on* $delta'$;

we calculate the difference between the distances to the two reference points, and compare it to ftm_diff, that is, the difference between the FTM-based distance measurements to the two reference points. We rank the candidates based on the similarity between the two difference values, and use the top-ranked candidate as the localization result.

IV. Experimental Evaluation

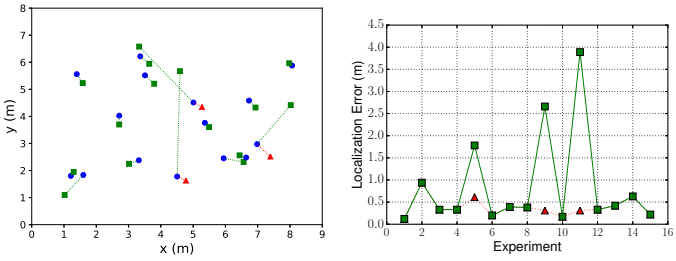
We use two Intel AX200 NICs as CSI collectors and FTM responders, i.e., reference points for localization, each with two external antennas. For CSI measurements and FTM, we installed the Intel backport driver release/core59 to Linux kernel 5.8.0 on Ubuntu 20.04 and the firmware version 55[3]. We use a Google Pixel 4 XL that supports FTM [10] as the localization target. Each Intel AX200 measures CSI in monitor mode, while acting as FTM responder in Access Point (AP) mode. Although Intel AX200 can measure CSI in AP mode, it requires the smartphone to connect to the AP (Intel AX200). As a result, a smartphone would have to switch across the reference point APs and the AP for Internet access. To avoid disrupting connectivity and user experience, the reference points switch between monitor mode and AP mode periodically, while the smartphone remains connected to the AP providing Internet access. This does not affect user network access while performing the user device localization. The smartphone is connected to a 5 GHz AP operating at channel 36 (5.18 GHz) and 40 MHz bandwidth, emulating a daily situation; given 5 GHz APs are becoming more common in indoor environment nowadays. The reference points monitor

[3]https://wireless.wiki.kernel.org/en/users/drivers/iwlwifi/core_release



(a) (b)

Fig. 4: (a) Device at the reference point, with 10 cm distance between the antennas. (b) A virtually fenced 9 $m \times 8$ m area for localization. 10 m between the reference points.



(a) (b)

Fig. 5: (a) 15 circle markers indicate true positions, and square markers indicate the top ranked candidates in the localization results (position estimates returned by the algorithm), and triangle markers indicate the 3 experiments, where the second ranked candidates are closer to the true positions. (b) The localization error values for the 15 localization experiments in (a).

at the same frequency. Due to higher attenuation in the 5 GHz band, compared to the 2.4 GHz one, localization in 5 GHz is more challenging. We use 2.4 GHz for FTM, because FTM operates only in 2.4 GHz with Intel NICs [6]. Note that the smartphone is able to initiate FTM requests in any frequency, regardless of the frequency of the AP it connected to.

We implemented a smartphone app that sends out ICMP packets (with arbitrary destination IPs) and FTM requests at a high frequency. The ICMP packets are monitored by the reference points for CSI measurements. We use CSI for 20-40 packets (not all ICMP packets are successfully monitored due to packet loss) and 50 FTM responses. Collecting more CSI and FTM responses would increase the localization accuracy, but it would also result in an undesirable longer delay. The CSI and FTM responses are all uploaded to a server that runs the localization algorithm.

Fig. 4 shows the experimental setup. Fig. 4a is the device used as the reference point, and Fig. 4b shows a virtually fenced 9 $m \times 8$ m area where localization was performed. The distance between the reference points is 10 m, and the distance between the two antennas is 10 cm.

As shown in Fig. 5, we placed the smartphone at 15 randomly chosen positions and localized it. The localization error for 12 (i.e., 80%) of them was less than 1 m, and for 10 (i.e., 66%) of them was less than 0.5 m. For 3 experiments, the second-ranked candidate localization result (as output by Algorithm 1) was the closest to the true position. For 12 of the experiments, our localization algorithm performed better than FTM after offset compensation (i.e, localization errors less than 2 m in similar 2D Line-of-Sight (LoS) static localization), based on trilateration with three or more reference points [12]. Moreover, localization would have been impossible with only AoA estimations, because the algorithm returns multiple candidates, with localization error for each candidate from 0.1 m to 5 m in our experiments. Our algorithm and experimental results are based on only two reference points; we can expect significant localization performance improvements with more reference points.

V. Conclusion

We investigated AoA estimation and FTM using Intel AX200, and proposed a localization algorithm that combines both methods. The experimental results show localization errors less than 1 m for 80% of the experiments. An immediate extension is to introduce more than two reference points. Another extension, having worked with the 40 MHz bandwidth at 5 GHz, is to explore CSI measurements with all modes (802.11a/g/n/ac/ax) and all bandwidths (20/40/80/160 MHz) in both 2.4 GHz and 5 GHz bands. Finally, with results in LoS conditions, we will experiment in more challenging indoor propagation environments (obstacles and non-LoS).

Acknowledgment

This work was supported in part by the KAW Fellow Trustworthy IoT project and the Security Link strategic research center. The authors would like to thank Dr. Revathy Narayanan for her input in the early stages of this work.

References

[1] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *SIGCOMM*, 2015.
[2] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *NSDI*, 2016.
[3] 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.
[4] J. Choi, "Sensor-aided learning for wi-fi positioning with beacon channel state information," in *arXiv:2007.06204*, 2020.
[5] A. Zubow, P. Gawłowicz, and F. Dressler, "On phase offsets of 802.11 ac commodity wifi," in *IEEE/IFIP WONS*, 2021.
[6] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai, "Verification: Accuracy evaluation of wifi fine time measurements on an open platform," in *MobiCom*, 2018.
[7] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *NSDI*, 2013.
[8] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "Calibrating phase offsets for commodity wifi," *IEEE Systems Journal*, vol. 14, no. 1, 2019.
[9] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
[10] "Wi-fi location: ranging with rtt." [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifi-rtt
[11] "IEEE Std 802.11-2020 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2021.
[12] C. Ma, B. Wu, S. Poslad, and D. R. Selviah, "Wi-fi rtt ranging performance characterization and positioning system design," *IEEE Transactions on Mobile Computing*, 2020.