Time-of-flight Wireless Indoor Navigation System for Industrial Environment

Maurizio Rea
IMDEA Networks Institute
& University Carlos III of Madrid
Madrid, Spain
maurizio.rea@imdea.org

Héctor Cordobés
IMDEA Networks Institute
Madrid, Spain
hector.cordobes@imdea.org

Domenico Giustiniano
IMDEA Networks Institute
Madrid, Spain
domenico.giustiniano@imdea.org

ABSTRACT

We introduce TWINS, Time-of-flight based Wireless Indoor Navigation System, that estimates in near real-time the position of commercial off-the-shelf (COTS) devices equipped with WiFi. TWINS uses COTS WiFi Access Points (APs) and it does not require access to any inertial sensors on the mobile devices. We study the performance of TWINS in a harsh environment, strongly affected by the presence of metallic objects. The scenario can be considered representative of an industrial indoor environment with open spaces surrounded by metal obstacles. We experimentally show that our system can localize four mobile robots tracked together, with median error between 1.8 and 3.8 m.

1 INTRODUCTION

Systems based on Time-of-Flight (ToF) ranging are gaining momentum in WiFi networks as alternative to traditional SNR-based approaches [7]. These systems use two-way ranging, taking advantage of DATA/ACK traffic exchange. Yet, they come with the inherent difficulties to obtain accurate time measurements with Commercial Off-The-Shelf (COTS) hardware [5, 6]. Work in the literature has also studied the combination of ToF with inertial sensors [14], at the drawback of requiring the active intervention of the mobile device (e.g. installation of an app). Inertial sensors typically convert accelerometer readings into steps, which makes it ineffective in scenarios where the target is a mobile robot. Other approaches circumvent information noise in the time estimation of the mobile device position by using time-differential-of-arrival (TDOA), and making the transmitter and the receiver hop across different bands [20]. However, the above approach would break any communication as WiFi devices communicate in a single frequency band.

We present our positioning system, TOF-based Wireless Indoor Navigation System (TWINS). Our system uses ToF measurements to compute the ranges from APs to targets. Our system was first introduced in [16]. With respect to our preliminary work, we present and implement:

- a time scheduler for APs to speed-up the convergence time of the positioning algorithm (Section 2);
- a ranging estimation technique that estimates the first path, and solves stability problems of the distance estimate over time caused by over-estimating the number of paths (Section 3);
- a system architecture for near-real time positioning with novel weighting function designed to prioritize APs providing stable distance estimates (Section 4).

In Sections 2, 3 and 4 we present in details the main components of TWINS. We then conduct extensive and controlled tests of TWINS in an industrial-like scenario, the w.iLab.t testbed [1], and show its performance for static and mobile (robots) devices. Our evaluation shows that TWINS can track simultaneously 4 mobile robots, with median error between 1.8 and 3.8 m. TWINS greatly outperforms the best average error of state-of-the-art systems in the same environment, that was 6-7 m of average error for a single target device [9]. These types of environments are of particular interest with the advent of Industry 4.0 solutions to automate manufacturing technologies. TWINS operates in a single frequency band, and it could then be used simultaneously for communication and localization. For instance, the same network could be used for tracking and communication of employers.

2 ACCESS POINT ORCHESTRATION

Our system is called TWINS, Time-of-flight based Wireless Indoor Navigation System. It uses COTS APs with customized firmware operating in the core of the 802.11 Medium Access Control (MAC) state machine of a low-cost WiFi chipset. TWINS can estimate the position of any 802.11 standard-compatible devices. TWINS can also be used for communicating to WiFi devices, as it does not require any changes to the WiFi protocol.

The APs are equipped with Broadcom WiFi chipsets that run our customized version of the 802.11 openFWWF firmware [16]. We measure the ToF to the mobile device(s) with two-way ranging measurements using Probe Responses, an 802.11

WiNTECH'19, October 25, 2019, Los Cabos, Mexico
2019. ACM ISBN 978-1-4503-6931-2/19/10…$15.00
https://doi.org/10.1145/3349623.3355476

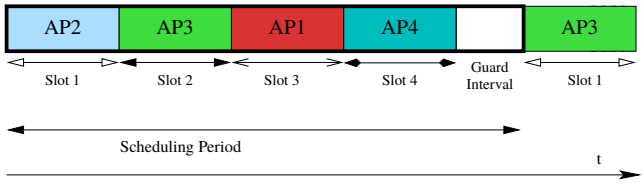


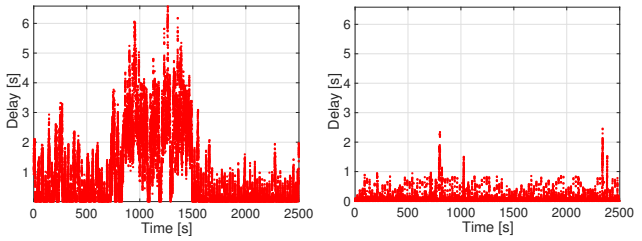
Figure 1: Measurement schedule and slot allocation

standard compliant solution. In particular, the target devices are associated to any of the ranging APs by means of a specific SSID common to the whole set of APs. Although the devices are connected to only one AP, they respond to the Probe Response sent from any AP in the network with an ACK after the SIFS time of the standard (the details are in Section 3). No modification to the clients is required for this. TOF measurements are passed to the open-source b43 driver[1] running in the AP, where they are pre-filtered. Data are then sent to the Central Location Unit (CLU) that orchestrates the network. The CLU computes the mobile position and connects to a database where the results are stored. The system is the result of several years of research and development.

2.1 Central Location Unit

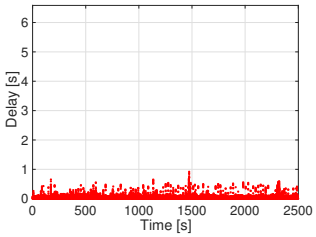
The positioning system is orchestrated by the CLU, which manages the APs and generates traffic to which target devices will respond. The CLU computes the ranges and positions, and makes data available to location-based applications through a database. As APs operate in the same radio frequency channel as the client, network collisions and hidden nodes among transmissions from different APs must be alleviated. For this reason, the measurements are scheduled by the CLU. As shown in the schematic view in Fig. 1, the CLU uses a time division scheduler to issue measurements rounds where only one AP at each time has the token to measure the ToF to target devices. As each AP implements the 802.11 protocol for transmitting the packets for ToF measurements, a final guard interval is allocated to drain the measurement queue of all APs. A report of the measurement round (series of ToF values per target) is sent by each AP at the end of the scheduling period through kernel socket. Using a time division scheduler, TWINS minimizes network collisions and hidden nodes. Together with the time division scheduler, we further randomize the AP slot assignment, within the scheduling period to compensate potential asymmetries due, for instance, to the target movement or hidden nodes effects caused by other networks.

[1]http://linuxwireless.sipsolutions.net/en/users/Drivers/b43/



(a) Period: 2ms, success rate: 91.7%.

(b) Period: 3ms, success rate: 98.6%.



(c) Period: 4ms, success rate: 99.1%.

Figure 2: Effect of different injection rates. Each figure shows, for an experiment in time (X-axis), the delay for the received packet in that instant (Y-axis).

2.2 Injection rate

The frequency to issue measuring messages is finite and is affected by different factors such as delay in communications (CLU towards AP, AP OS towards FW, collisions). To assess a good value for the injection frequency, we issue a series of tests, in which we observe the overall behavior at different injection rates. As seen in Fig. 2, smaller pauses between injections imply much larger delays since the measurement packet is injected (at OS level) until the report of TOF measurements is received. The drawback of this approach is that the CLU has less control on the time it needs to wait before receiving a sufficient number of measurements to perform range estimates. We also observe from Fig. 2 that, under such conditions, it is not possible to keep up with the injection rate. If the AP cannot send all packets in its assigned slot, it will start using the slot assigned to other APs. This, in turn, will increase the probability of collisions, and, especially of hidden nodes. The latter is an important issue as APs can be far away and may not see each other.

Despite we could consider the 4 ms scenario the safest of all, 3 ms injection period is more practical. In fact, Fig. 2(b) reports a success rate of Probe-ACKs of 98.6% against 99.1%, shown in Fig. 2(c). Furthermore, Fig. 3 shows the Empirical Cumulative Distribution Function (ECDF) of the delay for three different injection rates and, more specifically, it reports a 90%-ile delay of 97 ms and 90 ms for 3 ms and 4 ms injection

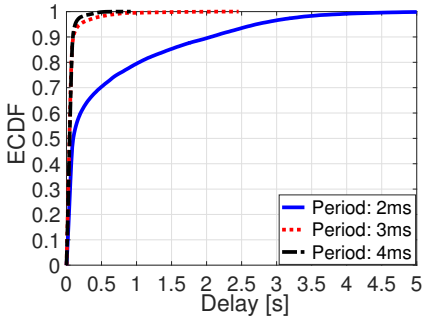


Figure 3: ECDF of the delay for different injection rates.

period, respectively. Hence, the difference between those two cases is not significant in terms of delay and success rate, but in in case of 3 ms injection period we are able to collect more samples in a fixed time window and this is an advantage, since TWINS aims to track users in real time.

For the implementation, TWINS sends $N = 20$ Probes in each measurement round [16], which results in a slot time of $20 \cdot 3 = 60$ ms. In the example of Fig. 1, the Scheduling Period is $60 \cdot 4 + 30 = 270$ ms, where 4 is the number of APs and 30 ms is the fixed final guard interval (11% of the scheduling period in this example). The final guard interval is allocated to drain the measurement queue of all APs.

Finally the experimental results from Fig. 2(b) show that 70% of the samples fall below the slot time of 60 ms, but the remaining 30% will surely occur after the allocated slot for the subsequent AP (but not the other APs). Once measurements are performed by each AP, the CLU collects these raw ToF measurements from the APs. After this set of measurements is processed, the CLU estimates the distance from each AP to each intended target as expressed in Section 3.

3 RANGING MEASUREMENTS

The ToF range is computed using regular 802.11 Probe Responses sent by the APs and acknowledged by the target device via 802.11 ACKs. We use Probe Responses rather than normal Data as we experimentally observe that the target device replies only to the Data of its associated AP, but not from other APs. In contrast, the target replies reliably to the short Probe Responses sent by any AP, which is a requirement of our positioning system. In this way, ToF ranges can be computed from multiple APs to estimate the mobile position. In addition, in order to guarantee the highest reliability of range traffic from APs at any distance from the target, Probes are sent at the basic rate of 1 Mb/s.

In order to minimize the duration of Probes, we limit the payload size of Probe Response to 30 Bytes (a small SSID and no extra extensions used). There are 12 bytes in fixed fields (timestamp, beacon interval and capabilities), and 18 tagged, of which 9 bytes are taken by the SSID, 6 by the B mode data rates and 3 by the DS parameters (channel used). Probes are generated in user space and sent to the virtual device driver operating in monitor mode through the pcap interface.

3.1 Distance estimation

For estimating the range to the target device on a given instant, we assume that, given N measurements, some of them follow the Line of Sight (LOS) path (or the shortest Non Line of Sight (NLOS) path in case the LOS path does not exist), and others have one or more NLOS paths. For each single path, the noise in the measurement is mainly due to the target device [5]. We consider a Gaussian distribution for the noise generated by the target replying with ACKs. Experimental observations of Gaussianity of the single path in controlled environments can be found in [16] with noise largely generated by the target device responding to the Probe Response.

We model the sum of all these multipath components as a *Gaussian Mixture Model (GMM)* [4]. A key aspect of the model is to identify the number of dominant paths (clusters) κ, which is up to 5 in typical indoor environments [11]. However, as each round contains 20 ToF measurements per AP, fewer clusters are expected. This is confirmed by our experimental observations, where setting a maximum number of clusters equal to 5 increases the probability that the GMM identifies clusters without physical meaning. Based on our experience, this, in turn, causes instability in the ranging estimates over time and, hence, in the positioning algorithm. Therefore, we limit the number of clusters to 3. We generate all the GMM models for each $\kappa \in \{1, \ldots, 3\}$, by using the iterative Expectation-Maximization (EM) algorithm initialized by a K-means++ run. We infer the optimal κ for the GMM statistical model selecting the model with the lowest Akaike Information Criterion (AIC) for the N measurements [10]. We finally consider the mean of each of the k paths. Our estimator rejects the paths with negative means, as they do not correspond to a physical propagation path, and uses the *path with the least positive mean* as ranging estimate.

No offline calibration is required by our ToF system, that automatically adapts to environmental changes, such as more or less people in the area, chairs and furnitures moved to a different location, variation of the radio interference level in the same channel. These features are considered essential for any positioning system [13].

3.2 GMM versus median

For the sake of the presentation, we compare two different implementations of GMM algorithm, one in C++ with libarmadillo [17], and the other one in Matlab, with the median metric for ranging estimation. Specifically, we use a data

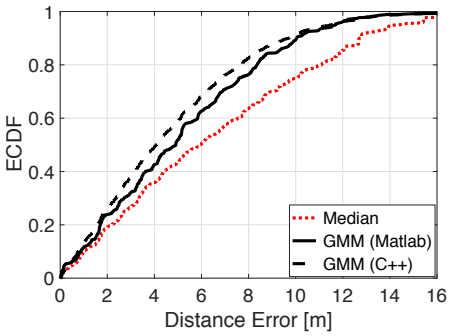


Figure 4: ECDF of the distance estimation error for GMM implementations compared to the simple median.

set from a testbed presented in [16], to analyse the distance estimation error, shown in Fig. 4. The plots confirm that both implementations of GMM algorithm achieve better results with respect to the median metric. Furthermore, C++ and Matlab versions have roughly similar estimation error, emphasising the robustness of GMM estimator across different implementations.

4 POSITION ESTIMATION

Using the estimated distances, a position-wise suboptimal but computationally efficient Linear Least Squares (LLS) multilateration algorithm is used to make an initial position estimate. Then the algorithm makes use the Weighted Non-Linear Least Squares (NLLS) technique to compute the position from this initial value.

We define a coordinate system on a two-dimensional map. To ease the notation, we consider only one target and a number of links equal to the number of APs in communication range, denoted as L. Let $\mathbf{s_i} = (x_i, y_i)$ be the position of the AP_i, $\mathbf{p} = (x, y)$ the position of the target device, and $r_i = \|\mathbf{s_i} - \mathbf{p}\| = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ the distance between the AP_i and the target device. Let $\hat{d}_i$ further denote the estimate of the distance from AP_i to the target. Our objective is to find the coordinates that solve the following weighted least square optimization problem:

$$\hat{\mathbf{p}} = (\hat{x}, \hat{y}) = \underset{x,y}{\operatorname{argmin}} \sum_{i=1}^{L} \frac{(\|\mathbf{s_i} - \mathbf{p}\| - \hat{d}_i)^2}{w_i} \quad (1)$$

The weight is computed as the distance with respect to the previously estimated position, that is:

$$w_i = \|\mathbf{s_i} - \hat{\mathbf{p}}[k-1]\| - \hat{d}_i[k] \quad (2)$$

where k indicates the iteration step. The idea of this approach is that a higher weight is an indication of instability in the link between the AP and the target device, likely meaning that the link is affected by multipath. This is especially important for industrial environments, where metals can completely block the line-of-sight paths and the estimated distance will be higher.



Figure 5: Picture of the testbed.

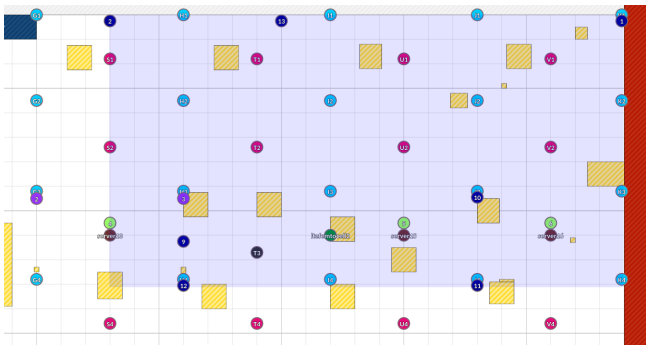


Figure 6: Map of the testbed. Yellow box corresponds to blockage.

5 EVALUATION

In this section we first introduce the experimental environment and then we analyze the performance in static and mobile scenarios. Finally we also show TWINS performance with multiple moving targets.

5.1 Testbed

We deploy and test TWINS in a testbed in Ghent, that is strongly affected by the presence of metallic objects. A picture of the testbed is shown in Fig. 5. The testbed can be considered representative of an industrial indoor environment with open spaces surrounded by metal obstacles. This scenario is a very harsh environment where radio communication has notorious difficulties. One of the key aspects is the fact that metallic objects create a strong blockage component and higher number of reflections that typical offices and houses.

The mobile measurements are done using as network elements a mobile node (Turtlebot II Robotic Platforms). On the map in Fig. 6, we depict the area of interest for the experiments (in blue), of about 11x22m. Yellow squares on the map indicate metallic blocks. Additional metallic objects in the

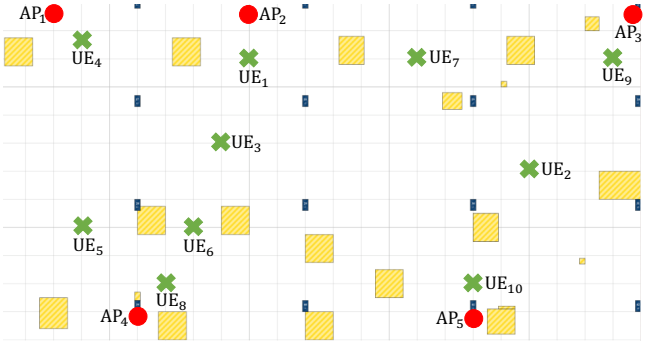


Figure 7: Scenario for static position estimation. Yellow box corresponds to blockage.

area are tubes presented in the area (cf. Fig. 5). We use five APs for positioning and all APs use Alixes boards. Mobile nodes (UEs) are all robots based on the Turtlebot II Robotic platforms.

Previous experimental studies in positioning systems in the same testbed have shown that all tested types of RF-based localization solutions degrade significantly in harsh environments because of the heavy presence of reflections, metal construction and fully shielded environment, in which the walls and ceilings are made of metal. In this industrial environment, the best known algorithms (based on signal strength measurements) have shown an average error distance of 6 - 7 m, while algorithms based on fingerprint and hybrid techniques have computed 11 m and 17 m of average error, respectively [9].

5.2 Static Node

In this subsection we analyze TWINS performance in a static scenario, shown in Fig. 7. We locate a specific robot in 10 randomly selected locations (marked with crosses) and we use five APs to test our localization system. It can be seen that the set of positions contains clear space as well as areas heavily surrounded by metal objects.

We study the position accuracy and summarize the results in Fig. 8. We show the ECDF of the positioning error for all the marked positions. The X-axis indicates the measured positioning error, and the Y-axis shows the cumulative probability (0.5 being the median). Two different results are shown: a raw estimate (in solid blue) and another one where the system applies knowledge of the area bounds (restricting the results to a given area, in dotted red). In the latter case, we limit the error on the extremes of the area by considering the closest position on the edge. As expected, the introduction of the area information improves the overall performance. We obtain a median and 80-percentile positioning error about 4 m and 5 m, respectively.

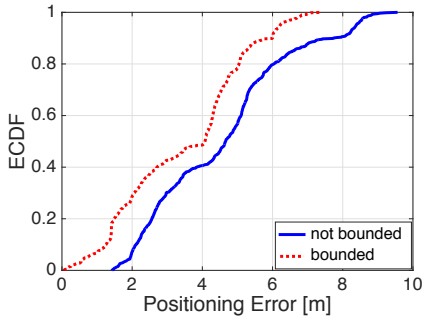


Figure 8: ECDF of position error in static conditions using 5 APs and 10 randomly selected locations.

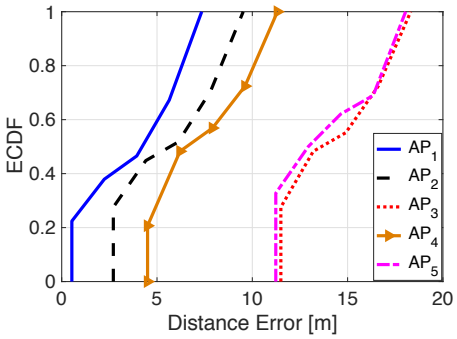


Figure 9: ECDF of distance error in static conditions for UE_4 only.

We use a subset of this data to analyze the ranging accuracy under different propagation conditions. We show in Fig. 9 the ECDF of the distance error of one specific location (UE_4) with respect to the five APs. From the map in Fig. 7, we can observe that three links are LOS (UE4 with AP1-AP2-AP4) and two are NLOS (UE4 with AP3-AP5). The results in Fig. 9 show that the blockage effects in the links AP_3-AP_5 to the target largely increases the median error of the distance estimate, without significantly changing the overall distribution. This effect is unique of industrial-like scenarios full of metal objects, as studied in this work. This property is instead not observed in other typical indoor environments as shopping malls and offices.

5.3 Mobile Node

We perform range and positioning tests for a mobile node using the testbed described in Sec. 5.1.

5.3.1 Distance. We start performing ranging measurements in mobile conditions. In the experiment, the robot moves toward the AP_1 at a speed of 0.16 m/s. The results are summarized in Fig. 10, where the x-axis shows the measurement round number for which the estimate is made, while the y-axis corresponds to the distance in meter. Each round is comprised of a small set of measurements (N=20) for which

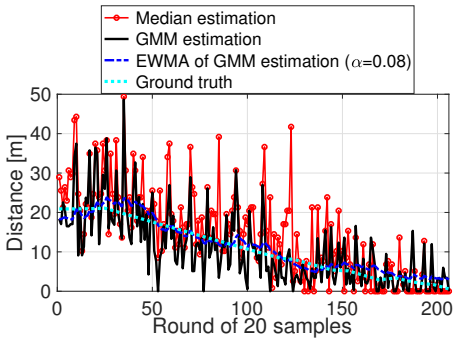


Figure 10: Ranging tests with mobile target.

we consider the target is at a semi-static position. The dotted line indicates the ground truth, and different statistics and filtered results are shown, namely median and our GMM-based estimation. We notice that regardless of the selection of a maximum of three clusters to run the GMM algorithm (cf. Section 3.1), the GMM can still under-estimate the distance more significantly with respect to the median estimator. As the estimate is measured on a relative small number of samples (20), a filter is necessary to improve the estimate. We then apply an Exponentially Weighted Moving Average (EWMA) filter of the distance:

$$\bar{d_k} = (1-\alpha)\bar{d}_{k-1} + \alpha\hat{d_k}, \quad s.t.\ \bar{d}_1 = \hat{d}_1 \tag{3}$$

where $\bar{d_k}$ is the current distance estimate, $\hat{d_k}$ is the last measured distance and α is the filter weight. An EWMA filter puts more or less weight on historical values according to the filter weight α, which depends on the mobility of the target. The Fig. 11 shows the Root Mean Squared Error (RMSE) of the GMM-based distance estimation versus α weight. At a speed of 0.16 m/s, α=0.08 is the best weight for our EWMA filter. We then can see in Fig. 10 that the EWMA metric estimates applied to the output of the GMM algorithm match very well to the ground truth. Even with the small time-delay caused by the EWMA filtering and the position mismatch (as we have a moving target, our estimate might be slightly delayed), the ranging error is limited, with the maximum error below 4 m. Hence the selection of a maximum of three clusters works well in practice with a low-pass filter. These values are very similar to those obtained previously by our system in other less challenging testbeds [16].

5.3.2 Positioning. We then make experimental tests for estimating the position of mobile robots. We run the experiment for four robots (9, 11, 13, 15) used as targets, along four different trajectories. Fig. 12 shows the ground truth (marked on blue continuous lines) and the position estimates (marked in red) for each trajectory.

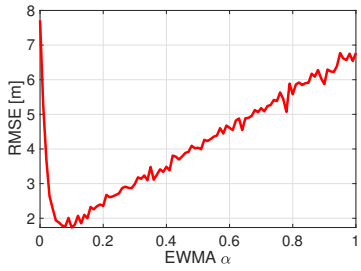
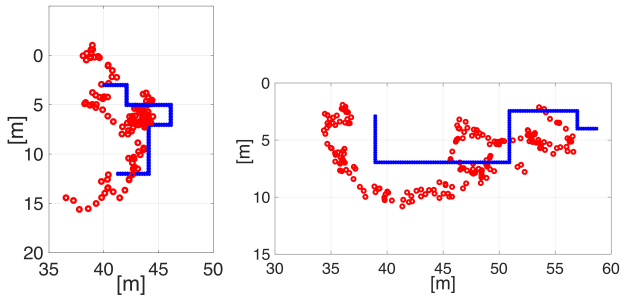


Figure 11: RMSE of GMM-based distance estimations versus EWMA α weight.



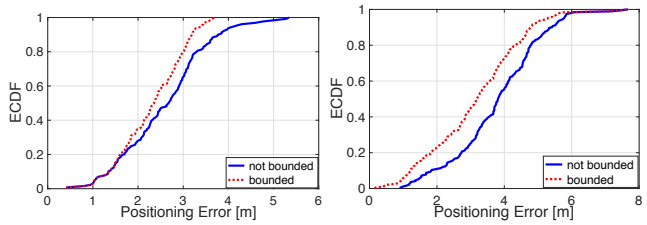
(a) Robot 9 trajectory. (b) Robot 11 trajectory.

(c) Robot 13 trajectory. (d) Robot 15 trajectory.

Figure 12: Real (blue) and estimated (red) trajectories in 4 different tests.

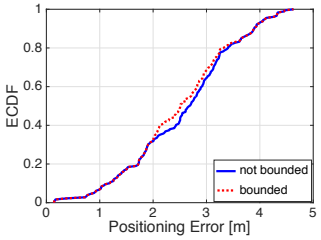
Fig. 13 shows the ECDF for each of the given paths in the test areas presented above. We evaluate the raw and bounded estimates on this test as well. It is worth noting that the positioning error is within the given error figures given above. Intuitively it might seem that in some cases the error is higher than expected, but we remark that the scale may be different for each figure.

The median error in the positioning estimate is below 4 m in the worst case, and in most cases below 3 m, for an area of 11 m x 22 m. We observe that the *accuracy tends to increase compared with the static case.* The reason is that moving devices experience different multipath constellations, and w_i as defined in eq. (2) tends to increase for links that are more affected by multipath with respect to static links. The result is that the positioning algorithm makes better decisions in the

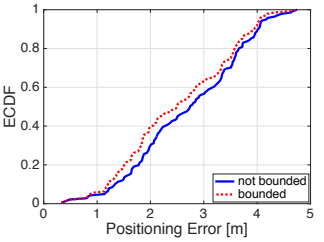


(a) Robot 9 trajectory.

(b) Robot 11 trajectory.



(c) Robot 13 trajectory.



(d) Robot 15 trajectory.

Figure 13: ECDFs of positioning error, with robots localized in separate tests.

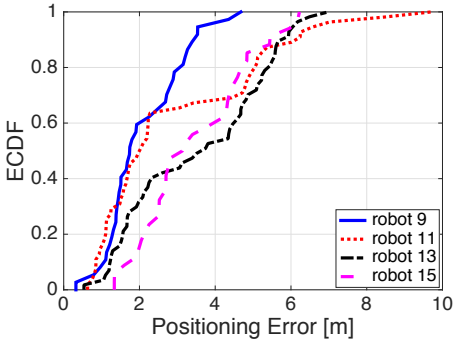


Figure 14: ECDF of positioning error, with TWINS localizing together all four robots.

weighting of each links and positioning computation. This effect compensates for the increased tracking error which arises when estimates are slightly lagging behind due to the measurement process delay, irrelevant when the device is at a static position.

5.4 Multiple Mobile Nodes

We finally run an experiment where we track all the robots simultaneously, on the given paths shown in Fig. 12. As main consequence of running the localization system with different targets at the same time, the latency of the system itself increases. Comparing the ECDF at 80% in Fig. 14 with Fig. 13, we can see a small degradation of the performances in terms of positioning error, which is to be expected as there are multiple targets.

6 RELATED WORK

ToF Echo techniques based on packet exchanges in WiFi networks have been proposed in [12, 15] and experimentally demonstrated in [3, 6, 8]. More recently, SAIL [14] is a ToF system using WiFi that has been designed for localization in multipath environments and achieve 80-percentile error of approximately 5 m. However, SAIL requires inputs from the inertial sensors in the smartphone. This is undesidered for the scalability of several applications. For applications where the target is a robot as studied in this work, the approach in [14] does not work, as the step counter is ineffective. ToneTrack [20] tries to overcome the problem of limited bandwidth, inherent to WiFi time-based localizations. It combines radio channels to form virtual larger bandwidths without increasing the radio sampling rate, taking benefit of frequency hopping, to increase the resolution of ToF profiles. The work SpotFi [11] uses APs equipped with 3 antennas and commodity WiFi chipsets. It jointly estimates the Angle of Arrival (AoA) and ToF pairs of each path using the channel state information. It then estimates the likelihood that these pairs correspond to the direct path between the AP and the target. However, they do not use ToF to estimate the range. In our previous work [16], we introduced a technique to estimate the range based on an online calibration model. However, the approach largely relies on the similarity between the multipath present in the AP to AP links and the ones between the AP and the target station. This may not hold in general, and especially in industrial-like scenario, where each link observes unique properties.

Although there are works that attempt to estimate the mobile position using a single AP [2, 19], they assume that the WiFi chipset in the AP can continuously change its frequency of operation and the client does the same, negatively affecting any other traffic in the network using commodity hardware and a single channel. Furthermore, these past works working with a single AP are negatively affected by strong blockage such as metal objects considered in this work. And they cannot rely on other APs as their assumption is that there are no other APs.

About orchestrations of Access Points, our proposed technique is designed to provide robust performance and low latency. There are other approaches presented in the literature, yet targeting different applications than ours. CENTAUR [18] introduced a semi-centralized data path for enterprise WiFi networks to address hidden and exposed nodes. As CENTAUR, our approach does not require any change on the client side. However, CENTAUR introduces a fixed backoff mechanism, which does not scale in hybrid deployments, where the deployed centralized network must coexist with other WiFi traffic. This could be, for instance, the case of

WiFi localization network deployed in a shopping mall, with WiFi channels shared with other WiFi APs.

7 CONCLUSION

We have proven that positioning mobile targets in industrial-like environments is possible using WiFi, and have demonstrated the capability to track in parallel 4 mobile robots with median positioning error in the range between 1.8 and 3.8 m, while the best performing from state-of-the-art systems in the same environment was 6-7 m of average error for a single target device.

ACKNOWLEDGEMENTS

We thank the team at the University of Ghent, and in particular Spilios Giannoulis, for their full support in the deployment of the testbed. We also thank Domenico Garlisi from the University of Palermo for his support in the firmware. This work has been funded in part by the European Commission in the framework of the H2020 project WiSHFUL (Grant agreement no. 645274), in part by the Madrid Regional Government through TAPIR-CM project S2018/TCS-4496 and in part by Ministerio de Ciencia, Innovación y Universidades (MICIU) grant RTI2018-094313-B-I00 (PinPoint5G+).

REFERENCES

[1] H2020 WiSHFUL project. http://www.wishful-project.eu.

[2] Chen, Z., Li, Z., Zhang, X., Zhu, G., Xu, Y., Xiong, J., and Wang, X. Awl: Turning spatial aliasing from foe to friend for accurate wifi localization. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies* (New York, NY, USA, 2017), CoNEXT '17, ACM, pp. 238–250.

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

[4] Fakhreddine, A., Giustiniano, D., and Lenders, V. Data fusion for hybrid and autonomous time-of-flight positioning. In *IPSN* (2018), ACM.

[5] Giustiniano, D., Bourchas, T., Bednarek, M., and Lenders, V. Deep inspection of the noise in wifi time-of-flight echo techniques. In *MSWiM '15* (2015), pp. 5–12.

[6] Giustiniano, D., and Mangold, S. CAESAR: Carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *CoNEXT '11* (2011), ACM, pp. 10:1–10:12.

[7] Goswami, A., Ortiz, L. E., and Das, S. R. Wigem: A learning-based approach for indoor localization. In *CoNEXT '11* (2011), ACM, pp. 3:1–3:12.

[8] Günther, A., and Hoene, C. Measuring round trip times to determine the distance between wlan nodes. NETWORKING'05, Springer-Verlag, pp. 768–779.

[9] Haute, T. V., Poorter, E. D., Moerman, I., Lemic, F., Handziski, V., Wolisz, A., Wirström, N., and Voigt, T. Comparability of rf-based indoor localisation solutions in heterogeneous environments: An experimental study. *Int. J. Ad Hoc Ubiquitous Comput. 23*, 1/2 (Jan. 2016), 92–114.

[10] Konishi, S., and Kitagawa, G. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.

[11] Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. Spotfi: Decimeter level localization using wifi. SIGCOMM '15, ACM, pp. 269–282.

[12] Li, X., Pahlavan, K., Latva-aho, M., and Ylianttila, M. Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems. In *IEEE Fall VTC 2000* (2000), vol. 6, pp. 3015–3020 vol.6.

[13] Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., and Sen, S. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *IPSN '15* (2015), ACM, pp. 178–189.

[14] Mariakakis, A. T., Sen, S., Lee, J., and Kim, K.-H. Sail: Single access point-based indoor localization. In *MobiSys '14* (2014), ACM, pp. 315–328.

[15] McCrady, D., Doyle, L., Forstrom, H., Dempsey, T., and Martorana, M. Mobile ranging using low-accuracy clocks. *Microwave Theory and Techniques, IEEE Transactions on 48*, 6 (2000), 951–958.

[16] Rea, M., Fakhreddine, A., Giustiniano, D., and Lenders, V. Filtering noisy 802.11 time-of-flight ranging measurements from commoditized wifi radios. *IEEE/ACM Transactions on Networking 25*, 4 (Aug 2017), 2514–2527.

[17] Sanderson, C. Armadillo c++ linear algebra library, June 2016.

[18] Shrivastava, V., Ahmed, N., Rayanchu, S., Banerjee, S., Keshav, S., Papagiannaki, K., and Mishra, A. Centaur: Realizing the full potential of centralized wlans through a hybrid data path. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2009), MobiCom '09, ACM, pp. 297–308.

[19] Vasisht, D., Kumar, S., and Katabi, D. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)* (Santa Clara, CA, 2016), USENIX Association, pp. 165–178.

[20] Xiong, J., Sundaresan, K., and Jamieson, K. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *MobiCom '15* (2015), ACM, pp. 537–549.