Evaluation of Self-Positioning Algorithms for Time-of-Flight based Localization

Aymen Fakhreddine
IMDEA Networks Institute
Madrid, Spain
University Carlos III of Madrid, Spain
aymen.fakhreddine@imdea.org

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

Vincent Lenders
Armasuisse
Thun, Switzerland
vincent.lenders@armasuisse.ch

***Abstract*—Self-localization systems based on the Time-of-Flight (ToF) of radio signals are highly susceptible to noise and their performance therefore heavily rely on the design and parametrization of robust algorithms. In this work, we study the noise sources of GPS and WiFi ToF ranging techniques and compare the performance of different self-positioning algorithms at a mobile node using those ranges. Our results show that the localization error varies greatly depending on the ranging technology, algorithm selection, and appropriate tuning of the algorithms. We characterize the localization error using real-world measurements and different parameter settings to provide guidance for the design of robust location estimators in realistic settings.**

I. Introduction

Positioning systems based on Time-of-Flight (ToF) have achieved large attention thanks to the widespread use of Global Positioning System (GPS). The underlying principle of ToF is that the distance between two devices is estimated measuring the time that the radio signal travels between two devices. More recently, there have been many attempts to apply ToF to WiFi networks, as an alternative to traditional methods based on Signal-to-Noise Ratio (SNR) fingerprints [1]–[6]. These works operate in the 802.11 infrastructure (WiFi Access Points, APs) with different requirements in terms of cost of deployment –from very cheap WiFi chipsets with modified firmware to Ettus software-defined radios–, and consider unmodified mobile devices. However, it is often the case that the WiFi infrastructure has been already deployed and it is not economically convenient to upgrade it to allow the APs to compute WiFi ranges.

In this work we consider a mobile device that aims to self-localize itself taking advantage of the available WiFi and GPS infrastructures. When WiFi (GPS) is selected, the mobile device measures the ToF ranges to commercial 802.11 APs (GPS satellites) and compute its position. This problem differs from the literature above, that instead considers that, for the case of WiFi, the ranging and the position are computed by the infrastructure. Although a mobile may compute its position selecting one between the embedded WiFi and GPS technologies, the mere application of a multilateration algorithm without a deep understanding of the underlying technologies and related noises largely limits the accuracy of a self-localizing mobile device. The key reason is that GPS and WiFi have been originally designed to address different fundamental problems: positioning for GPS and communication for WiFi. In order to understand, characterize and compare these technologies we implement the following simulators and testbeds:

- We build a GPS simulator that starts from raw data to perform all the GPS functionalities related to the satellites' position computation, and computes the position with the algorithm of our choice.
- For WiFi ToF positioning, we deploy a testbed equipped with a mobile client running a modified WiFi firmware to perform ToF measurements to Commercial-Off-The-Shelf (COTS) WiFi APs in range.

Using these tools, we investigate what are the key theoretical and practical differences in the noise sources of GPS and WiFi for self-positioning. Our contributions are listed in what follows:

- We investigate the errors inherent in the WiFi ranges and GPS pseudoranges measurements and provide a comparison between the two technologies for ranging.
- We study positioning algorithms and how they can be adapted to each technology by properly setting the right parameters.
- We implement techniques presented in the literature, discuss advantages/disadvantages, propose possible improvements, and characterize them with empirical studies.
- We carry out a performance evaluation in terms of positioning accuracy using experimental studies in both outdoor and (multipath-rich) indoor environments.

We show that a dynamic estimation of the standard deviation of the pseudoranges for the Extended Kalman Filter (EKF) is beneficial for the accuracy of GPS –a median accuracy of around 3 m in outdoor tests, better than the Least Squares (LS) that stagnates in a 6.95 m 2D error–. Unlike the GPS case, when the mobile client uses WiFi ToF positioning in indoor, the EKF and LS achieve relatively similar median accuracies of around 4 m and the dynamic estimation of the covariance matrix of the measurement noise does not bring the expected benefit. This indicates that the noise sources in WiFi ToF are less predictable than in GPS.

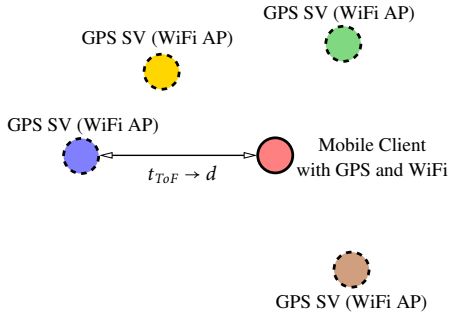

Fig. 1. High-level representation of a ToF-based self-positioning system with GPS and WiFi. A client estimates the ToF to the anchor (Satellite or Access Point) in range and computes its position.

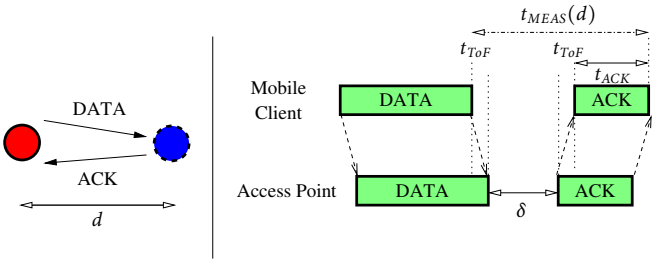

Fig. 2. ToF range measurements in WiFi.

II. ToF-based Ranging

In ToF-based positioning systems, a linear relation holds between the ToF (propagation time) t_{ToF} of radio signals and the distance d, $d = c \cdot t_{ToF}(d)$, where c indicates the speed of light. Multipath propagation makes the problem non-linear, due to the overestimate resulted from reflected paths. ToF ranging requires either synchronized clocks (as in GPS), or an echo technique (as in WiFi). The latter eliminates the need of clock synchronization. Using ToF measurements, the position can be computed by the client applying multilateration algorithms. A representation of the system with GPS and WiFi technologies is shown in Fig. 1. A client estimates the ToF to satellites (SVs) or APs in range and computes its position. In what follows, we review the background material of GPS and WiFi ToF ranging necessary to characterize and compare the noise sources of these two wireless technologies.

GPS ToF Pseudorange Computation. GPS signals are modulated with a Pseudo-Random Noise (PRN) code unique to each satellite (SV). Each satellite's signal is modulated additionally with a navigation message transmitted in 30 sec. The message consists of 5 frames (of six seconds each) and it includes the "ephemeris data" and the "clock corrections and satellite health", used to calculate the position of each satellite in orbit and compute the distance from each satellite to the client (called pseudorange pr). Each GPS satellite has an on-board atomic clock and includes the timestamp [time sent] of the signal it broadcasts. A GPS receiver also keeps track of time (with its clock) of the received signal [time received].

The receiver first performs a coarse estimation to compute t_{ToF} by comparing [time sent] and [time received] at packet level. The accuracy of this process is in the order of ms. A fine estimation is then obtained computing the sub-millisecond part of t_{ToF} at PRN code level, measuring the time shift t_{ToF} between the local copy and received PRN code. Finally, the pseudorange is estimated as $pr = c \cdot t_{ToF}(d)$.

WiFi ToF Range Computation. ToF measurements are subject to severe noise and have thus long been considered impracticable for WLAN localization. In the last years, the availability of open firmwares and drivers has made possible to investigate ToF ranging methods in WiFi networks using commodity hardware [3], [7], [8]. For WiFi ranging, we use the WiFi ToF echo technique used in recent works such as [8], that leverages the existing timing information of the WiFi 802.11 protocol. It relies on the DATA/ACK frames exchange in WiFi communication to infer the distance between the communicating devices. Fig. 2 shows how the ToF is extracted from the WiFi communication. As in the GPS case, in this work we consider that ToF ranging is *computed by the mobile client*, measuring the time t_{MEAS} elapsed from the end of DATA transmission to the end of ACK reception:

$$t_{MEAS}(d) = 2 \cdot t_{ToF}(d) + t_{off}, \tag{1}$$

$t_{ToF}(d)$ is the time of flight of the WiFi frame, and $t_{off} = t_{ACK} + \delta$ is the time offset in the measurement, where t_{ACK} is the duration of the ACK and δ is the interframe time between the reception of the DATA and the transmission of the ACK at the AP. Assuming that t_{off} is known (the case where this does not hold is presented in Sec. VI), for each DATA/ACK frames exchange, the mobile client performs a coarse approximation of the ToF as:

$$t_{ToF}(d) = \frac{t_{MEAS}(d) - t_{off}}{2}. \tag{2}$$

In order to achieve a fine estimation of the ToF, we use the filter presented in [8], that uses N consecutive samples $\{t_{ToF}\}_{1 \le n \le N}$ collected as above and estimates the distance based on an environmental training.

III. Error analysis of ranges

This section studies and compares the error sources in GPS and WiFi ToF ranges, and means for their correction.

GPS pseudoranges errors. We start with a brief description of the sources of errors in the GPS pseudoranges:

1) Satellite clock offset: The satellite's atomic clock experiences a bias and a drift that must be corrected by the receiver for each satellite it is receiving signal from individually. This correction is approximated by a second order polynomial:

$$C_{SVClock} = a_0 + a_1(t_{tr} - t_{oc}) + a_2(t_{tr} - t_{oc})^2, \tag{3}$$

where t_{tr} is the transmission time at the satellite that corresponds to the GPS satellite's time at the receiver t_{GPS} minus the travel time: $t_{tr} = t_{GPS} - pr/c$, and t_{oc} is a GPS reference time available in the ephemeris data. The polynomial

coefficients for clock correction a_0, a_1 and a_2 are broadcasted by the satellites in the ephemeris data and denote respectively the clock bias, the clock drift and the clock drift rate.

2) Satellite relativistic effect error: The relativistic effects influence the conversions of clocks proper times at the level of the satellite and on the earth's surface to GPS time and the expression of the positions in a turning geocentric system. The relativistic correction for Earth-Centered Earth-Fixed (ECEF) coordinates and a GPS satellite of eccentricity e and an orbit with semi-major axis A, eccentric anomaly E_k and group delay t_{gd}, is given by:

$$C_{Relativistic} = F \cdot e \cdot \sqrt{A} \cdot sin(E_k) - t_{gd}. \quad (4)$$

The inputs parameters are given in the ephemeris data.

3) Ionospheric error: The ionosphere is a dispersive medium ionized by the action of solar radiations, so as GPS signals pass through this atmospheric layer they suffer a delay proportional to the ion density that has a very large spatial and temporal variability, hence the difficulty of modeling the ionospheric time delay. For the implementation, we consider the analytical model developed by Klobuchar [9] to estimate the correction factor $C_{Ionospheric}$ related to this delay.

4) Earth-rotation during the travel-time: During the time the GPS signals travel to the receiver at the earth's surface, the earth rotates and consequently the ECEF coordinates sytem, thus the necessity of transforming the old satellites coordinates. Given a travel time t_{ToF} from a satellite with coordinates X_{SV} to the receiver, the rotated coordinates are:

$$\mathbf{X_{SV}^{Rot}} = \mathbf{M^{Rot}} \cdot \mathbf{X_{SV}}, \quad (5)$$

$$\mathbf{M^{Rot}} = \begin{pmatrix} \cos(\Omega_{t_{ToF}}) & \sin(\Omega_{t_{ToF}}) & 0 \\ -\sin(\Omega_{t_{ToF}}) & \cos(\Omega_{t_{ToF}}) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (6)$$

where $\Omega_{t_{ToF}} = \omega_{earth} \cdot t_{ToF}$ and $\omega_{earth} = 7.292115 \cdot 10^{-5} rad/s$ is the earth's rotation rate.

5) Multipath: Multipath may cause an enlargement of the travel time and thus an overestimation of the range.

Finally the measured pseudoranges from the receiver's antenna to the satellite SV is computed as follows:

$$cr_{SV} = pr_{SV} + c \cdot (C_{SVClock} - C_{Relativistic} - C_{Ionospheric}), \quad (7)$$

where cr_{SV} indicates the corrected pseudorange.

WiFi ToF ranging errors. Unlike GPS signals, WiFi signals are terrestrial so they are neither affected by the atmospheric perturbations (no ionospheric error) nor by the relativistic effects nor the earth rotation. Moreover the two-way ranging of WiFi-ToF localization eliminates the clock offset error. However new types of noise alter the accuracy of WiFi-ToF positioning:

1) SIFS Noise δ: The interframe time δ between the reception of the DATA frame and the transmission of the ACK frame is a time fixed by the 802.11 standard and called Short InterFrame Symbol (SIFS). Even though the SIFS time is set as a fixed value by the 802.11 standard, $\delta = 10\,\mu$s in 802.11b for instance, an error in the order of $1\,\mu$s is tolerated for communication. When it comes to localization, this tolerance becomes considerable, since an order of magnitude $1\,\mu$s corresponds to 300 meters of distance error.

The distribution of the overall offset $t_{off} = t_{ACK} + \delta$ is determined by δ. [8] showed the normality of this noise for the chipsets used in this work. However, this assumption may not hold in general [10]. The average value of t_{off} can be either pre-calibrated per chipset or estimated in real-time. Former works considered only pre-calibrated values. In Sec. VI, we study how to estimate t_{off} and the impact of this estimation on the WiFi ranging error.

2) Multipath: Multipath is the only common source of error in GPS and WiFi ToF ranges. However, the relative impact of multipath can be much higher in WiFi ranges with respect to GPS pseudoranges because:

- the WiFi range can be in the order of the multipath delay
- measurements of WiFi ranges do not have access to physical layer information. Instead, ToF is simply measured using WiFi frames that synchronize to the strongest (and sometimes delayed) signal component.

The multipath reflections depend on the location-specific characteristics of the environment, thus the need of the environment training algorithm (presented in Section II and described in more details in Section VI) to alleviate its impact.

3) Inaccuracy of the COTS WiFi clocks: Since WiFi was designed for communication but not for localization, WiFi chipsets are not equipped with highly precise clocks. This results in inaccuracies in the computation of t_{MEAS} [8] and therefore in the corresponding range and position.

IV. Positioning Algorithms

The mobile client's position is estimated applying the multilateration method that defines the position of the client at the intersection of the spheres centered in the SVs (APs) with each range equal to the GPS pseudorange (WiFi range) measured experimentally. When the localization system has a number of ranging measurements higher than the number of variables to estimate, the location problem can be posed as an optimization problem. The most used algorithms for multilateration are based on the Least Squares (LS) approach and the Extended Kalman filter (EKF) [11]. For these algorithms, we dissect the key parameters that must be tuned properly in order to achieve a good positioning accuracy.

Least Squares-based positioning. Given a set of estimated ranges $\{\hat{d}_i\}_{1 \le i \le M}$ with M the number of anchors used for positioning, the LS algorithm finds the coordinates $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ of the mobile client that satisfy the following

minimization problem:

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}^T = \underset{x,y,z}{\operatorname{argmin}} \sum_{i=1}^{M} (\|\mathbf{x_i} - \mathbf{p}\| - \hat{d}_i)^2, \quad (8)$$

where $\mathbf{x_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ is the position of the anchor i. We use the Newton-Raphson method for the computation.

For a fair comparison with the EKF presented in the following subsection, we use an Exponentially Weighted Moving Average (EWMA) smoothing of the position fixes estimated by the LS. The reason is that the EKF takes the previous position fix into account to estimate the next one, and then the smoothing property is inherent to this filter. When we refer to LS-positioning in the rest of the paper, we always mean the LS followed by the EWMA smoothing.

Extended Kalman filter-based positioning. We define the state vector to be estimated by the EKF as $\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & z & v_z & ck_B & ck_D \end{bmatrix}^T$ where $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ represents the cartesian coordinates of the receiver, v_x, v_y, and v_z those of the speed vector and finally ck_B and ck_D are respectively the receiver's clock bias and clock drift in meter unit (multiplying by c) to have an homogeneous state vector. The positioning problem is a discrete-time process with a state model:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}) + \mathbf{w_{k-1}}, \quad (9)$$

and a measurement model:

$$\mathbf{z_k} = g(\mathbf{x_k}) + \mathbf{v_k}, \quad (10)$$

Let t denote the positioning time interval. For GPS, $f_{GPS}(\mathbf{x}) = [x + v_x \cdot t, v_x, y + v_y \cdot t, v_y, z + v_z \cdot t, v_z, ck_B + ck_D \cdot t, ck_D]^T$, and for WiFi:

$$f_{WiFi}(\mathbf{x}) = [x + v_x \cdot t, v_x, y + v_y \cdot t, v_y, z + v_z \cdot t, v_z]^T,$$

while $g_{GPS}(\mathbf{x}) = \|\mathbf{p} - \mathbf{p_{SV}}\| + ck_B$ for the case of GPS positioning and $g_{WiFi}(\mathbf{x}) = \|\mathbf{p} - \mathbf{p_{AP}}\|$ for WiFi positioning ($\mathbf{p_{SV}}$ and $\mathbf{p_{AP}}$ refer to the coordinates of the position of the anchor considered for ranging measurements, a satellite SV or a WiFi Access Point AP). The usage of the EKF rather than the Kalman Filter (KF) results from the non-linearity of these measurement equations.

$\mathbf{w_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_k})$ and $\mathbf{v_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k})$ represent respectively the process noise and the measurement noise with autocovariance matrices $\mathbf{Q}$ and $\mathbf{R}$. In the next section, we provide an analysis about the measurement autocovariance matrix $\mathbf{R}$ and its impact on the EKF-positioning accuracy.

V. Covariance estimation for the EKF-positioning

Adequately characterizing the measurement noise is a decisive step to achieve acceptable performance with the EKF [12]. Given the differences between the two ToF localization technologies we are considering for our analysis, we give hereafter different methods that we use to estimate this measurement noise covariance.

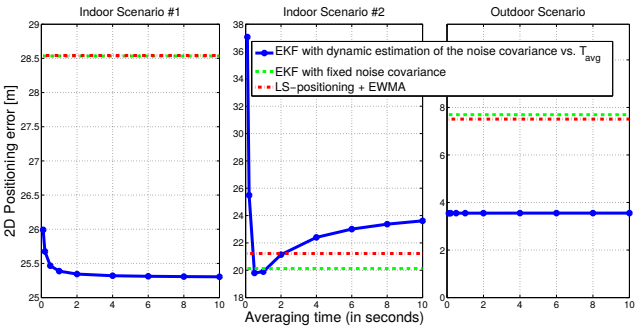

Fig. 3. The Root Mean Square Error (RMSE) of the positioning versus T_{avg} in different scenarios

Covariance estimation for EKF GPS-positioning. For GPS, we consider different ways to estimate $\mathbf{R}$. For space reasons, we only present the details of a method that uses a dynamic estimation of the noise covariance. In the analysis, we will compare it against a method that uses a fixed noise covariance.

1) The Early minus Late Delay Lock Loop model: This method takes into account the measurements at every GPS epoch and treats every single satellite separately. The model relies on the structure of the delay lock loop (DLL) used by most GPS receivers, including the one we are using in our experiments, that is based on the correlation of the received GPS signal with slightly early and late versions of that signal. Based on this functioning of the DLL, [13] derived the following formula to compute σ_{PR} of a given satellite $s_i \in \mathcal{SV}$, $\mathcal{SV} = \{s_{i1}, s_{i2}, \ldots, s_{im}\}$ being the set of visible satellites at the considered GPS epoch:

$$\sigma_{PR}(s_i) = c \cdot T_c \cdot \sqrt{\frac{d}{4 \cdot T_{avg} \cdot \{C/N_0\}(s_i)}}, \quad (11)$$

- d: the correlator spacing. It corresponds to the fixed time between the early and late correlator samples. We use $d = 0.1$, as implemented by many receiver manufacturers.
- T_c: the chip time set to $T_c = 1\mu$s as in normal GPS receivers.
- T_{avg}: the averaging time. We experimentally derive T_{avg} used in our chipset considering different scenarios (outdoor and indoor) and testing different values of T_{avg}. The results are reported in Fig. 3 and show that $T_{avg} = 1$s provides robust results.
- $\{C/N_0\}(s_i)$ (in dB-Hz): the carrier-to-noise-density ratio from satellite s_i.

The autocovariance matrix can then be written as:

$$\mathbf{R} = \mathrm{Diag}\big([\sigma^2_{PR}(s_{i1}), \sigma^2_{PR}(s_{i2}), \ldots, \sigma^2_{PR}(s_{im})]\big). \quad (12)$$

Covariance estimation for EKF WiFi-positioning. Also in WiFi, we consider two methods for the covariance estimation. In WiFi positioning, the anchors are in fixed locations which makes the estimation of $\mathbf{R}$ much easier than in the case of GPS where the satellites are permanently moving.

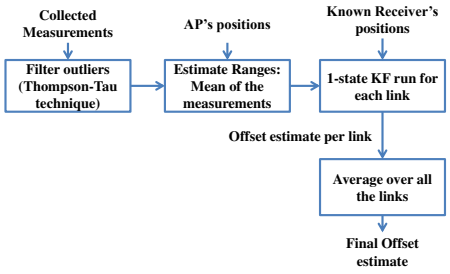

Fig. 4. Block scheme for the offset estimation

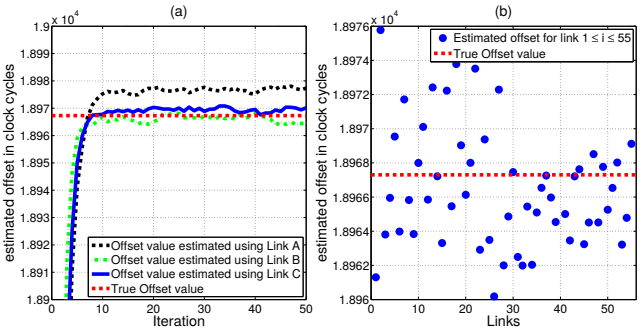

Fig. 5. Offset estimation for various links. The estimator converges quickly, after approximately 10 iterations (802.11 frames).

1) Estimation using a receiver-dependent fixed std-value: The standard deviation of the ToF measurements is lower bounded by the large noise added by δ. We experimentally measure that this noise is in the order of $[4.0, 4.1]$ WiFi clock cycles[1]. In practice, the multipath adds additional noise in the system. Consequently, we fix $\sigma_{WiFi} = 5.0$ clock cycles and have $\mathbf{R_{WiFi}} = \sigma^2_{WiFi} \cdot \mathbf{I_{n\times n}}$, with n being the number of WiFi APs used for ranging.

2) The Covariance Matching method for a fixed number of APs: [14] presented an adaptive estimation of $\mathbf{R}$ using the covariance matching principle. It makes the residuals $\hat{v}_k = \mathbf{z_k} - g(\hat{\mathbf{x}}_\mathbf{k})$ that are the differences between the observation $\mathbf{z_k}$ and its corrected estimated value $g(\hat{\mathbf{x}}_\mathbf{k})$ consistent with their theoretical values ($\hat{\mathbf{x}}_\mathbf{k}$ is the corrected state vector). $\mathbf{R_k}$ is then computed based on the estimated variance-covariance matrix of the residuals sequences $\mathbf{C}^{\hat{v}}_\mathbf{k}$ over a moving window of m time instants.

$$\mathbf{C}^{\hat{v}}_\mathbf{k} = \frac{1}{m}\sum_{i=1}^{m} \hat{v}_{k-i} \cdot \hat{v}^T_{k-i} \tag{13}$$

We only apply this method to WiFi since it requires a stable set of anchors for a sufficient number of measurements, something we do not observe in GPS environments that are affected by the movement of the satellites and consequent variation of the ranging quality.

VI. WiFi ToF measurement-offset t_{off} estimation

The standard approach to perform ToF in WiFi is to consider that t_{off} is calibrated offline using controlled measurements, for instance using cables connected to the WiFi transceiver [3], [8]. However, t_{off} depends on the chipset's manufacturer and the transmission rate. This presents the drawback of calibrating each WiFi chipset. This is unfeasible considering the application scenario of a mobile that aims to measure the ToF to any AP in range to position itself. We then introduce a methodology to estimate t_{off} in an iterative way. Using the Kalman Filter (KF)[2], and following the measurement model of GPS that we presented in Sec. IV, we observe that there is an analogy between the WiFi ToF measurement offset t_{off} and the clock bias at the GPS receiver side ck_B, since both of them add an offset to the ToF. From eq. (1), we can then express the measurement model as:

$$g(\mathbf{x}) = 2 \cdot \|\mathbf{p} - \mathbf{p_{AP}}\| + d_{off}, \tag{14}$$

where $d_{off} = c \cdot t_{off}$ is equivalent to the ck_B in GPS estimated iteratively in the state vector. The block scheme for the offset estimation is depicted in Fig. 4.

For the estimation, we collect $\{t_{MEAS}(d_n)\}_{1\le n\le N}$ sequences for a given set of positions of the mobile device in the indoor testbed (known beforehand), and for a fixed position of the APs in range. This procedure is performed once, as it is also needed for the environmental training of WiFi ranging used in this work (cf. Section II).

Since the coordinates (x, y, z) of the mobile client are known, the only remaining unknown is the offset d_{off}. We use the mean of the sequences $\{t_{MEAS}(d_m)\}_{1\le m\le M}$ after that the outliers are filtered using the Thompson-Tau technique to estimate the WiFi ranges that will be the inputs of the KF later on to estimate d_{off}. For the implementation, we use $M = 30$ samples to reliably perform this outliers filtering. As mentioned before, we use a 1-state KF to estimate the d_{off}, which means that only one equation is needed. We then take the average value of all these values for the set of links with the same WiFi chipset in the AP[3] as a final t_{off} estimate. In Fig. 5(a) we illustrate the offset estimation for 3 different links to show that the convergence is very fast, considering as initial state the nominal value of t_{off}, given by $\delta = 10\,\mu s$ and t_{ACK} derived based on the transmission rate [15]. The final estimated offset value for a given link is the average over all the iterations after convergence. Additionally, Fig. 5(b) shows

[1]Considering that the WiFi clock in our mobile client runs at 88 MHz, the clock cycle is equal to $\frac{10^{-6}}{88} \approx 1.13 \cdot 10^{-8}$ sec.

[2]The eq. (14) is linear because the positions of the mobile client are known. Thefore, we use the KF instead of the EKF.

[3]In this work, the APs have the same WiFi chipset. The algorithm works as well for APs with different WiFi chipsets.

Rate [Mb/s] (802.11b)	1	11
True Offset [clock cycles]	27854.0	18967.3
Estimated Offset [clock cycles]	27855.6	18966.6

TABLE I
True and estimated offset for the Broadcom AirForce54G 4318 wireless card in the APs.

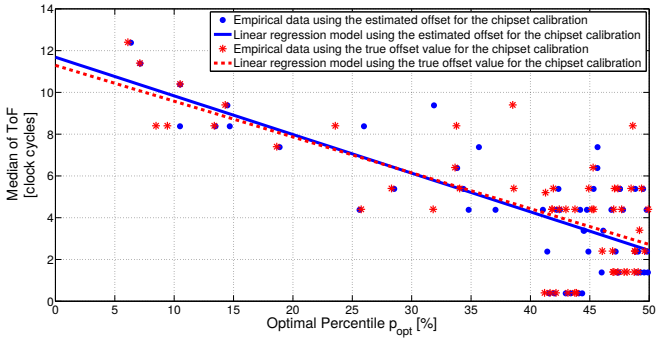

Fig. 6. Environment training: the linear regression model using the estimated offset does not differ considerably from the one using the true offset.

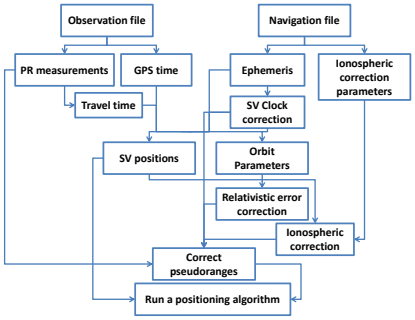

Fig. 7. Flowchart of our GPS simulator based on real traces

how the estimated offsets per links are spread around the true offset value, thus the use of the averaging to find the final offset estimate.

Robustness of the offset estimation for different transmission rates. The previous analysis was carried out using the ToF measurements with ACK transmitted at 11 Mb/s transmission rate. We perform the same algorithm using the collected samples with ACK transmitted at 1 Mb/s transmission rate to show that this method gives accurate estimation of t_{off} for different transmission rates. Table I summarizes the results. We observe that the estimated offset is remarkably close to the true offset, measured in a controlled test using coaxial cables between the WiFi chipsets.

Impact on the environmental training model. As expressed in Section II, we use the environmental training first presented in [8] to achieve a fine estimation of the distance. In this section, we study how robust is the environmental training using an offset t_{off} that is estimated as in Fig. 4. From [8], let $t_{ToF}(p)$ denote the ToF using the p-percentile of $\{t_{ToF}\}_{1\leq n\leq N}$. In addition, let p_{opt} denote the optimal percentile value that gives the closest approximation $t_{ToF}(p_{opt})$ to the ground-truth ToF $t_{ToF,gt}$ and $\hat{p}_{opt}$ its estimated value. p_{opt} can be expressed as:

$$p_{opt} = \underset{0\leq p\leq 0.5}{\operatorname{argmin}} \; |t_{ToF,gt} - t_{ToF}(p)|. \qquad (15)$$

p_{opt} is a value equal or below 50% to counterbalance the positive biased values due to the multipath in NLOS (Non Line Of Sight) links. $\hat{p}_{opt}$ is estimated based on a linear model derived from the empirical distribution of the median of ToF $t_{ToF}(p = 0.5)$ versus p_{opt}.

Since the environment training above is defining a linear regression model linking the optimal percentile and the median of sequences of t_{ToF}, this environment training depends in fact on the chipset calibration because it is what allows us to obtain t_{ToF} starting from t_{MEAS} (see Eq. (2)). Fig. 6 shows that the new environment training model we get using the estimated offset value is close to the one we get using the true offset, which shows the robustness of our approach to calibrate the chipset offset. In Table II we give the numerical values of these coefficients.

regression coefficients using true offset	−0.17	11.29
regression coefficients using estimated offset	−0.18	11.68

TABLE II
COEFFICIENTS OF THE LINEAR REGRESSION MODELS USED FOR THE ENVIRONMENT CALIBRATION FOR ESTIMATED AND TRUE OFFSET VALUES

VII. THE GPS SIMULATOR BASED ON REAL TRACES

We collect real traces using the Evaluation Kit with Precision Timing manufactured by U-blox and equipped with an active GPS antenna of type u-blox ANN-MS, connected by a 5 meters coaxial cable. This device runs the U-center software for configuring the GPS receiver's settings, and testing, visualizing and analyzing the collected data. We choose this device because it presents the advantage of storing the raw data that we need to evaluate the positioning algorithms and deeply analyze the sources of error.

We connect the U-blox device to a laptop and collect traces in three different scenarios: indoor scenario #1, indoor scenario #2 and outdoor. We first convert the proprietary file format of U-blox to Receiver Independent Exchange Format (RINEX), a raw navigation system data format that does not depend on the receivers manufacturer or its special features. The data logs are classified in different files; the ones of interest for this study are the observation and navigation files (.obs and .nav).

1) Observation file: This file contains the pseudoranges, the code phase, the Doppler shift and the signal strength measurements from every single visible satellite mapped by their pseudorandom noise numbers (PRN) at every GPS epoch when the measurements were carried out.

2) Navigation file: Additionally to the ephemeris data broadcasted by the satellites that includes the SV clock bias, drift and drift rate, the Keplerian parameters, perturbation parameters and additional information proper to each SV, this file contains the ionospheric correction parameters and the terms of polynomial that are necessary to compute the satellite clock offset. To convert the data logs from .ubx to RINEX we use RTKLIB, an open source program package for GPS positioning.

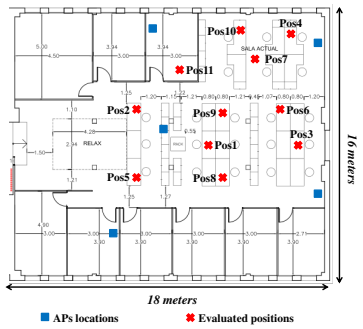

Fig. 8. WiFi Testbed with the APs locations and the evaluated positions

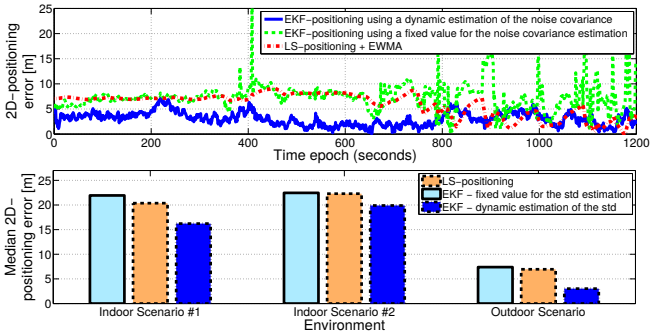

Fig. 9. On the top of the figure we provide a comparison of the accuracy of different GPS positioning techniques in outdoor scenario, on the bottom we show the median accuracy of GPS positioning techniques in different scenarios.

The simulator implementation. We implement a GPS simulator in Matlab that starts from the raw data logs in a RINEX format to perform all the functions related to the satellites' positions computation and the pseudoranges corrections introduced in Section III. The flowchart of the simulator is depicted in Fig. 7. The simulator allows us to run the positioning algorithm we want to evaluate for the receiver's position calculation at every epoch.

VIII. The WiFi Localization Testbed

The WiFi testbed is equipped with 5 APs and a mobile client that we place in several positions. The map of the testbed is shown in Fig. 8, where we consider 11 positions in an office environment with a rich-multipath experience. The APs are COTS embedded machine (Soekris net5501) with a 500 MHz AMD Geode LX single chip processor equipped with Broadcom AirForce54G 4318 mini PCI type III chipsets. The mobile client is a laptop operating Ubuntu 12.04 with a kernel version 3.0.0-12, it is equipped with the same chipset as in the APs that runs our customized version of the 802.11 openFWWF firmware and b43 driver. This mobile client measures the ToF to the APs in range using regular 802.11 DATA frame as presented in Section II.

Implementation. For the implementation, $t_{MEAS}(d)$ is measured by monitoring two events, the end of the DATA transmission and the end of the ACK reception. The timing of the WiFi echo technique is defined by the General Purpose Timer (GPT), that operates in the wireless chipset at the resolution of the internal clock (88 MHz in our chipset). The GPT is launched in the MAC state machine of the firmware just after the 802.11 processor sets up the COND_TX_DONE condition register (a frame has been sent) which indicates that the timer starts to count clock cycles. The firmware does other operations as required by the 802.11 protocol. Once the ACK frame has been received (or the ACK timeout has elapsed), the COND_RX_COMPLETE register gets updated and the timer gets stopped. The value is saved as the number of clock cycles elapsed since the end of the DATA transmission. Because of the limited space available in the 802.11 chipset for storing, we copy the new measurement $t_{MEAS}(d)$ into the shared memory in hexadecimal format after that it has been obtained, and we then retrieve it using the b43 driver. In the driver, we gather additional data about the incoming ACK such as the data rate, MAC addresses, etc, and store them in a buffer. Once the buffer is full or a timeout elapsed, the data is transferred to the user space with UDP sockets.

IX. Performance Evaluation

In this section we present the results of the experimental studies we carried out.

Experimental study: GPS ToF localization. The top of Fig. 9 shows the 2D-accuracy over time of GPS localization. We observe that the EKF with a dynamic estimation of the measurement noise autocovariance matrix achieves the best performance with a median positioning error of 3.02 m. Using the aforementioned version of the EKF, we gain more than 4 m with respect to the EKF using the same value of the variance of the pseudoranges for all the satellites, that has a median error of 7.39 m. The reason is that a dynamic estimation of the measurement noise is a more accurate model, that considers at the same time the properties of the receiver and the quality of every single satellite where the GPS signals are coming from. The LS-positioning algorithm has a median error of 6.95 m, and it slightly outperforms the EKF-positioning using a fixed value for the noise covariance estimation. Yet the EKF-version with the dynamic estimation of the noise is significantly better. Indeed, when it is well tuned, the EKF performs better than the LS in estimating a discrete-time dynamic process, which is the case here: although we are considering a static GPS receiver, the motion of the satellites makes the whole system dynamic. Fig. 3 supports these results while considering different scenarios, for instance an indoor environment with a very reduced SV visibility (Indoor Scenario #1), an indoor environment with acceptable SV visibility (Indoor Scenario #2), and an outdoor environment. Finally, on the bottom of Fig. 9, we compare the median accuracy of GPS positioning techniques in different outdoor and indoor scenarios that confirm that we encounter high errors in the indoor scenarios.

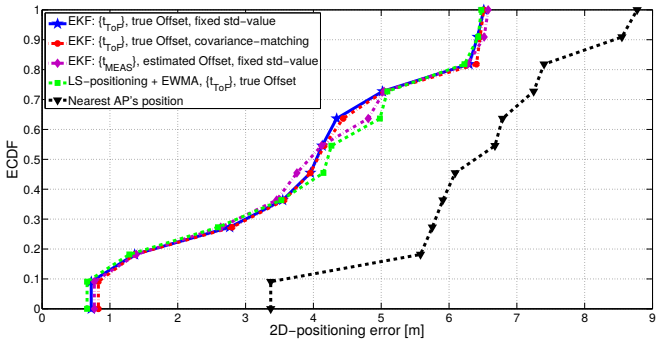

Fig. 10. Comparison of the accuracy of different WiFi positioning techniques for different positioning algorithms.

Experimental study: WiFi ToF localization. In Fig. 10, we compare the accuracies of the EKF and LS based positioning taking as a baseline the positioning approach that takes the nearest AP's position in terms of shortest ToF as the mobile station's position. We evaluate the EKF under different schemes: i) EKF where the inputs are sequences of t_{ToF} obtained after a calibration using the true offset value and ii) under different noise autocovariances matrices, one using a fixed standard deviation (std) value of the noise and the other one using the covariance matching method explained in Section V, and iii) EKF having as inputs the sequences of t_{MEAS} calibrated using the estimated offset as described in Section VI with a fixed autocovariance matrix. For the LS, we only use calibrated measurements using the true offset. Overall, the EKF and the LS have similar performance (median error of 4 meters for the EKF and 4.26 meters for the LS). This is unlike the GPS localization, where the EKF clearly outperforms the LS. The reason for this different result with respect to GPS stands in the ranging computation, where an adaptive filter is needed in WiFi to correct the impact of large indoor multipath (cf. Section II). Finally, LS and EKF both perform better than the "Nearest AP's position" approach with a gain of the median error of more than 2 m.

Using the covariance matching method for the EKF instead of a fixed autocovariance matrix did not improve the positioning accuracy (both schemes achieve practically the same performance). In fact, we are evaluating a static scenario, wherein the variance of the measurement noise does not considerably change. In this setting, the autocovariance matrix is sufficiently well characterized by the standard deviation value presented in Sec. V. However, in dynamic scenarios, the number of anchors will change frequently, and we foresee that this will cause problems of convergence of the covariance matching method. Finally, comparing the positioning accuracy between using the true offset versus the estimated one for the chipset calibration, we observe in Fig. 10 that using the estimated one we do not have a noticeable loss of performance in terms of positioning accuracy which confirms the robustness of the KF algorithm presented in Section VI.

X. Conclusion

In this paper, we studied the problem of a mobile client aiming to position itself using ToF-based localization technologies. We studied and compared the sources of errors affecting ToF ranging measurements, designed and implemented testbeds and simulators, and presented an extensive performance evaluation of LS and EKF positioning algorithms. We showed how crucial is the adequate tuning of the measurement noise autocovariance matrix to obtain good performances with the EKF, and the difference in the performance of EKF and LS algorithms for ToF GPS and WiFi self-positioning, as a result of the heterogeneous sources of noise in the two technologies.

Acknowledgment

This work has been funded in part by the European Commission in the framework of the H2020-ICT-2014-2 project Flex5Gware (Grant agreement no. 671563) and in part by Ministerio de Economía y Competitividad grant TEC2014-55713-R and partially by the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

References

[1] A. Günther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," ser. NETWORKING'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 768–779.

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

[3] D. Giustiniano and S. Mangold, "CAESAR: Carrier sense-based ranging in off-the-shelf 802.11 wireless LAN," in *CoNEXT '11*. New York, NY, USA: ACM, 2011, pp. 10:1–10:12.

[4] P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, and I. Tinnirello, "WMPS: A positioning system for localizing legacy 802.11 devices," in *Transactions on Smart Processing and Computing,*, October 2012.

[5] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 315–328.

[6] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *MobiCom '15*. New York, NY, USA: ACM, 2015, pp. 537–549.

[7] M. Ciurana, D. Giustiniano, A. Neira, F. Barcelo-Arroyo, and I. Martin-Escalona, "Performance stability of software ToA-based ranging in WLAN," in *IPIN*, 2010, pp. 1–8.

[8] A. Marcaletti, M. Rea, D. Giustiniano, V. Lenders, and A. Fakhreddine, "Filtering noisy 802.11 time-of-flight ranging measurements," in *CoNEXT '14*. New York, NY, USA: ACM, 2014, pp. 13–20.

[9] J. Klobuchar *et al.*, "Ionospheric time-delay algorithm for single-frequency gps users," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 3, pp. 325–331, 1987.

[10] D. Giustiniano, T. Bourchas, M. Bednarek, and V. Lenders, "Deep inspection of the noise in wifi time-of-flight echo techniques," in *MSWiM*. ACM, 2015, pp. 5–12.

[11] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, no. 3.

[12] J. Durbin and S. Koopman, *Time series analysis by state space models*. Oxford University Press, Oxford, 2001.

[13] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[14] A. Almagbile, J. Wang, and W. Ding, "Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33–40, 2010.

[15] I. Tinnirello, S. Choi, and Y. Kim, "Revisit of RTS/CTS exchange in high-speed IEEE 802.11 networks," in *WoWMoM*, June 2005, pp. 240–248.