Multiple Transmitter Localization under Time-Skewed Observations

Mohammad Ghaderibaneh, Mallesham Dasari, Himanshu Gupta
Department of Computer Science, Stony Brook University

***Abstract*—Radio spectrum is a limited natural resource under a significant demand and thus, must be effectively monitored and protected from unauthorized access. Recently, there has been a significant interest in the use of inexpensive commodity-grade spectrum sensors for large-scale RF spectrum monitoring. These sensors being inexpensive can be deployed at much higher density, and thus, can provide much more accurate spectrum occupancy maps or intruder detection schemes. However, these sensors being inexpensive also have limited computing resources, and being independent and distributed can suffer from clock skew (i.e., their clocks may not be sufficiently synchronized).**

In this paper, we are interested in the problem of detection and localization of multiple intruders present simultaneously, in the above context of distributed sensors with limited resources and clock skew. The key challenge in addressing the intruder localization problem using sensors with clock skew is that it is very difficult to even derive an observation vector over sensors, for any (absolute) instant. In this work, we propose `Group-Based` Algorithm, a skew-aware multiple intruders localization method that essentially works by extracting observations across sensors for certain small sets of transmitters. Our results show that `Group-Based` Algorithm yields significant improvement of accuracy over relatively simpler approaches.

I. Introduction

Radio spectrum is a limited resource and is in extreme demand because of exponential growth in wireless applications. Shared spectrum can be vulnerable to unauthorized transmissions and thus, must be protected against unauthorized users (intruders). This issue has recently been exacerbated by the increasing affordability of software-defined radio (SDR) technologies making RF transmissions of arbitrary waveforms in arbitrary spectrum bands more practical than ever before. Popular among such attacks include misguided road navigation using GPS spoofing [1], crashing aircraft instrument landing systems [2], etc. Detecting and localizing such spectrum offenders is a challenging problem that has been recently tackled in multiple dimensions [3].

One way to protect spectrum is via large-scale spectrum monitoring [4]. One issue in such efforts is that lab-grade spectrum sensors are large and expensive both to procure and operate. This issue has recently been addressed by promoting the use of small and inexpensive spectrum sensors that can potentially be crowdsourced [5]. In such crowdsourced architectures formed of large number of spectrum sensors, a typical architecture involves a centralized entity (aka spectrum manager, or a fusion center) collects spectrum data from the deployed, and processes the data received for a collective decision on detecting and localizing the intruders accurately [6].

However, one concern in the context is that these inexpensive sensors can suffer from clock drift/skew, resulting in skewed-observations at the fusion center. This problem is exacerbated by the fact that these sensors are heterogeneous with varying processing speeds and communication capabilities [7]—these can introduce further time-offset in the observations of these sensors. Despite the accuracy of well-known clock synchronization techniques [8], [9], the sensors can have a clock drift of as much as a few tens of μsecs. Overall, we have observed the offset between observation timestamps to be as high as 100s of microseconds. As a result, the signals from neighboring sensors are misaligned and can lead to poor localization accuracy. A recent work has already shown such signal alignment problem and tackled in a different context [10], [11].

Our goal in this work is to address the multiple intruder localization problem in the above context, design schemes that are able to circumvent the localization inaccuracy introduced due to the time-offset observations from distributed sensors, and evaluate their performance for varying parameters. In our setup, an intruder can generate tones/pulses of few 10s of μsecs and/or change its power every 10s of μsecs. Localizing the intruders transmitting such intermittent pulses of duration that are of the same order as the time-offset among the sensors, poses a significant challenge while fusing the observations from the distributed sensors to localize intruders.

In this paper, we present `Group-Based` Algorithm, a scheme that attempts to localize intruders in the above setup and achieve accurate localization of multiple intruders simultaneously. `Group-Based` Algorithm relies on the fact that a some distributed sensors will likely receive power from the *same* set of transmitters. Thus, the `Group-Based` Algorithm works by dividing sensors into groups such that each group is receiving signal from the *same set* of intruders[1]. Using these groups, `Group-Based` Algorithm extracts sufficient observation sequences to localize minimal-sized sets of transmitters independently.

We evaluate the `Group-Based` Algorithm using a small scale real data and large scale simulations. For real experiments, we create an indoor lab test-bed that consists of 4 sensors and 2 transmitters, with USRP [12] SDRs as transmitters and RTL-SDRs [13] as sensors. The SDRs are interfaced with Odroid boards [14] to provide compute and network capability. Our experiments on the data from real test-bed shows that

[1]We use the words intruders and transmitters, interchangeably.

these sensors have an average offset of 100 μs (§IV-A). Using these real offset measurements, we evaluate `Group-Based` Algorithm under diverse conditions such as different skew ranges, number of intruders and sensors. We find that our `Group-Based` Algorithm can achieve 23% less false alarm rate compared a `Naive` algorithm under real data with two intruders (§IV-A). With extensive simulations, we also show that our `Group-Based` Algorithm can perform 2-4$\times$ better than the `Naive` algorithm under various settings (§IV-B).

II. Problem Formulation and Related Work

Localization Using Skewed Observations (LUSO) Problem. Consider an shared spectrum region, where we are interested in localizing any intruders/transmitters present. The intruders may transmit an arbitrary signal with varying power, but we assume that each power level is maintained for a few 10s of microseconds.[2] To localize such intruders, we have a set of spectrum sensors that have been deployed a priori over the region. Each sensor i records IQ samples over *sensing windows* of say 10 μsecs, and report the received power in the channel of interest (we assume a single channel of interest, for simplicity). We assume that each sensor makes observations at the same frequency, and denote the j^{th} observation by the i^{th} sensor by o_{ij}. Due to the skew across sensors, note that o_{ij}'s for a particular j may not correspond to observations at the same absolute instant. We do not make any assumptions about the channel propagation model, i.e., we assume the path loss between a pair of points to be a Gaussian distribution of arbitrary mean and standard deviation. The sensor observations are periodically sent to a *fusion center*, which collects all the observations over a period, and processes them in some manner to determine locations of the intruders with high accuracy. Since the number of intruders is not known, the localization result may yield many misses and/or false alarms. The focus of our work is on circumventing the challenge of *skewed* observations and not the well-studied multiple-intruder localization problem, and thus, we use the state-of-the-art multiple intruder localization scheme SPLOT [3] as an available procedure.

At a high-level, any localization scheme effectively uses a *single* observation vector $(o_1, o_2, \ldots, o_n)$, where o_i is an observation of i^{th} sensor, to localize intruders. The key point here is that the observations o_i are of the same absolute instant or the intruders are continuously transmitting. In our context, due to the skewed observations and the intruders transmitting short duration pulses, a direct application of a standard localization scheme is not effective.

Related Work. Localization of a transmitter or an intruder in a field using sensor observations has been widely studied, with the majority of works having focused on localization of a single transmitter [15]. To localize multiple intruders, the main challenge comes from the need to "separate" powers at the sensors, and in absence of blind separation methods, there have been few works that have directly addressed the multiple intruder localization problem. Recently, [3], [16] have presented interesting techniques for the problem. We have used the best of these approaches, i.e., SPLOT [3], as a procedure within our developed techniques. The focus of this work is on addressing the challenge that arises due to time-skewed observations from the sensors; to the best of our knowledge, ours is the first work to address this issue, especially in the context of multiple intruder localization.

[2]This assumption is due to the "sensing window" being of size 10 microseconds as defined later, and is needed to be able to "perfectly align" similar signals from different spectrum sensors, as discussed towards the end of §III.

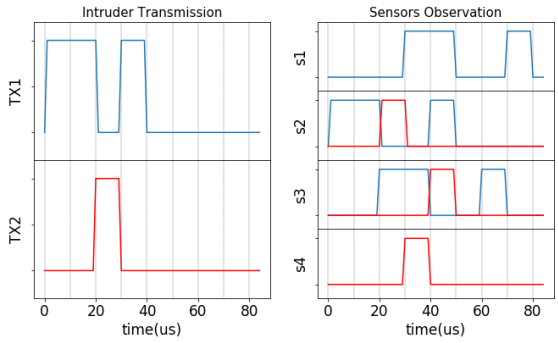

Fig. 1: Transmitted pulses from two intruders, and received signal at four sensors (with skewed clocks).

`Naive` **Approach.** To illustrate the problem challenges, let us describe a simple approach to solve the above problem. Given the observations o_{ij} for $1 \leq i \leq n$ and $1 \leq j \leq m$, where n is the number of sensor and m is the number of time instants, the `Naive` algorithm essentially consists of two steps:

1) For each j, Solve the multiple-intruder localization problem using the observation vector $O_j = (o_{1j}, o_{2j}, \ldots, o_{nj})$. Let the *set* of intruder locations returned be L_j.
2) Cluster the set of locations in $\bigcup_j L_j$. We use k-means method, which requires a known number of clusters. In our evaluations, we vary the range of number of clusters that is given to the algorithm.

LUSO Example. Consider an area with four sensors and two intruders, $TX1$ and $TX2$. Figure 1 shows the pulses transmitted by the intruders in absolute time. In particular, $TX1$ sends two pulses (in blue) of length 20μs and 10μs at absolute times 0 and 20 μs, and $TX2$ send a single pulse (in red) of length 10μs at 20μs. The figure also shows the pulses received by the sensors, based on their local timestamps. Without any information about the skew, the `Naive` algorithm would create 8 observation vectors of 4 observations each. Note that for an observation vector to yield any localization output, at least 3 values in the vector must be non-zero value (as at least three sensors are required to localize an intruder). Here, only two observation vectors (at 30-40μs and 40-50μs) consists of non-zero values from three sensors—and that formed of observations from different absolute times. Thus, this example shows that the localization based on observation vectors as done by `Naive` is likely to be very inaccurate.

III. `Group-Based` Algorithm

In this section, we describe our proposed algorithm, referred to as `Group-Based` Algorithm, for solving the `LUSO` problem effectively. We start with giving a high-level idea and intuition behind our proposed technique.

As in the previous section, let n be the number of sensors, and let o_{ij} denote the j^{th} observation of the i^{th} sensor. We assume the sensor locations are known. In the previous section, we presented the simple `Naive` approach. At a high-level, the idea of the `Naive` approach is to localize based on observation vectors for each time instant. Such a scheme does reasonable well *if* the observations have no skew; however, with time-skewed observations, the `Naive` approach can result in high inaccuracy, as observed in our simulations (see §IV-B).

Basic High-Level Idea. Let us assume a skew of of at most θ between any pair of sensors. The key idea behind our proposed `Group-Based` Algorithm approach is to partition the sensors into disjoint groups such that each group consists of sensors that "receive" transmission (i.e., receive a power more than noise) from the *same set* of intruders (note that not every sensor receives transmission from every intruder, due to the signal attenuation, and a particular sensor may receive transmission from multiple intruders). Once such groups have been formed, we consider the groups that correspond to the smallest set of intruders and for each group g independently, we use the observations of sensors in g to localize the corresponding intruders. Thus, our above approach can be looked upon as a divide and conquer approach. More formally, our algorithm can be described as a sequence of following steps.

1) **Creating Groups of Sensors.** Partition the set of sensors S into groups $G_1, G_2, \ldots, G_k$ such that two sensors s_x and s_y are in the same group if and only if their observation sequence can be perfectly aligned, i.e., they are receiving pulses from the same set of intruders, and hence, are in the "vicinity" of the same set of intruders. In particular, to check if observation sequences of two sensors s_x and s_y can be perfectly aligned, we compare the observation sequence of s_x with the observation sequence of s_y shifted by z units for all z less than the maximum skew. We discuss how to do this comparison later. Note that the number of groups is at most equal to the number of sensors. For each group G_r, we denote the set of intruders whose pulses are received by the sensors in G_r by I_r.
2) **Directed Graph Over Groups.** Create a partial order over the above groups based on the intruder-subset relationship, i.e., create a directed graph (partial order) $\mathcal{G}$ over nodes $G_1, G_2, \ldots, G_k$ with directed edges (G_r, G_s) if and only if for the corresponding set of intruders $I_r \subseteq I_s$. We discuss later how to determine the edges in the above graph.
3) **Expand the Directed Graph $\mathcal{G}$.** We then expand the graph $\mathcal{G}$ by adding additional "virtual" nodes. We discuss this step in further detail below, but the purpose is to create new groups that represent other sets of intruders not already captured in $\mathcal{G}$.
4) **Localizing Intruders Corresponding to Source Groups.** Note that $\mathcal{G}$ is a DAG. For each node G_r in $\mathcal{G}$ that is a source (i.e., no incoming edges), localize I_r by using observations of I_r in G_r as well G_r's ancestors in $\mathcal{G}$. We discuss this step in further detail below. Let L_r be the set of locations obtained by localizing intruders from observations in G_r.
5) **Final Result.** Since the source groups considered above correspond to disjoint sets of intruders, we return the union of L_r's as the final result.

We now discuss some of the details skipped in the above description.

Steps (1) and (2) Details. Comparing Received Signals. Now, we explain how we compared received signals of two sensors and check if they both receive powers from the same set of transmitters. First, let us assume that there is only one transmitter. In this simple case, sensors that are able to receive power from the transmitter, will receive "approximately" the same signal, though with some path loss attenuation. The fact that sensors use a sensing window of $10\mu s$ would only result in some approximation of the transmitted signal, since the transmitter maintains every power level for sufficiently long time (at least a few 10s of microseconds). Now, if there are multiple *non-overlapping* (see below) transmitters, then the received signals is just a simple "combination" of the signals due to each transmitter. Now, to check if two sensors receive power from the same set of transmitters, we shift one of the sensors' signal by all values less than the skew, and for each value, calculate the *normalized cross-correlation* between the signals. For two signals to be from the same set of transmitters, the correlation value should be close to one for at least one shift. Now, finally, let's relax the assumption of non-overlapping signals. If transmitters can have overlapping transmissions, then the received signal at a sensor can potential have power level changing at an unbounded rate and the above technique may not work due to the sensing window being of 10 microseconds. However, for a sufficiently large observation window (depending on the maximum number of transmitters around a sensor), with high probability there will always be a part of the signal that does not exhibit the above rogue behavior—and this is sufficient for our purposes. We omit the tedious details.

Step (3) Details. Expanding $\mathcal{G}$. Consider two nodes G_r and G_s in $\mathcal{G}$ such that the corresponding sets of intruders I_r and I_s are not disjoint and one is not a subset of other. In this case, we create a *virtual* group G_{rs} corresponding to the set of intruders $I_r \cap I_s$. This virtual group did not exist in the original graph $\mathcal{G}$, since there are no sensors that observe just the set $I_r \cap I_s$ of intruders. However, we can still create and make use of such virtual groups, if we are able to extract/derive observation sequence(s) received from $I_r \cap I_s$. These observation sequences for these virtual groups can actually be extracted by finding the observations that are

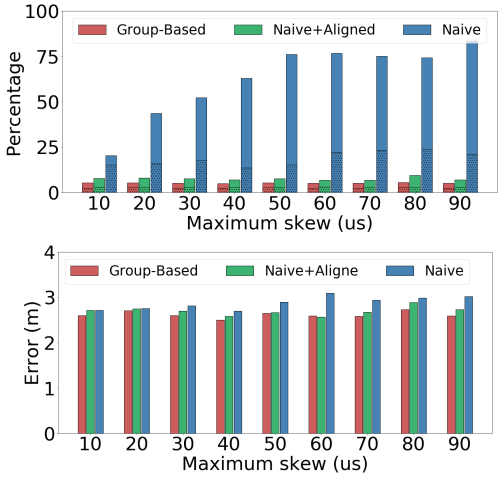

Fig. 2: Localization performance vs. Skew in localizing 7 intruders, Percentage of miss and false alarms(above). The filled region in each bar represents the misses and unfilled region represents false alarm, Localization error(bottom).

common in the sensors of G_r and G_s. More precisely, for every sensor s_i in $G_r \cup G_s$, we "align-intersect" its observation sequence O_i with the observation sequence of *some* sensor in G_s – to derive an observation *subsequence* O_i^{rs}. The set of such observation subsequences O_i^{rs} correspond to the virtual group G_{rs}. Above, by "align-intersection" of an observation sequence O_i with an observation sequence O_j, we mean the following: First, we align the two sequences O_i and O_j, and then extract the pulses from O_i that has a corresponding pulse in O_j at the same time instant (after alignment).

Step (4) Details. Localizing Intruders Corresponding to Source Groups. First, note that each source node G_r in $\mathcal{G}$ clearly has a minimal set of transmitters. Now, to localize intruders in I_r as accurately as possible, we first try to extract as many observation sequences as possible corresponding to the intruders in I_r. We do this by extracting appropriate observation subsequences from sensors in the groups/nodes G_s such that there is a path from G_r to G_s in $\mathcal{G}$ (and thus, I_s is a superset of I_r). Finally, we use the set of these subsequences to localize I_r (by first aligning them near-perfectly, localizing for each of the aligned observation vectors, and then clustering the results).

IV. Evaluation

We evaluate `Group-Based` Algorithm on a small test-bed as well as over a large scale simulation environment, and compare it with two versions of the `Naive` scheme of §II.

Performance Metrics. We evaluate the localization performance of the schemes in terms of the following metrics: in terms of three metrics: Localization error (in meters), and number of misses and false alarms as a percentage of actual number of intruders. For a given solution S of predicted intruder locations and the actual ground truth T of actual intruder locations, we compute these metrics as follows: For each intruder $t \in T$, we find the closest predicted intruder in S and if this intruder is within a certain threshold, then we report the difference as the localization error. The number of intruders in S that do not find a match as above are reported as misses, and the unmatched predictions in T are false alarms.

TABLE I: Localization performance on real data.

Algorithm	Misses (%)	False Alarm (%)
`Naive`	28	9
`Group-Based`	5	6

TABLE II: Initialize Parameters

Parameters	Values
grid size	50m*50m
cell size	1m*1m
# of cells	2500
# of sensors	100
arrival rate(Poisson distribution)	λ = 8
pulse length	[20μs, 30μs]
pulse power	[-36dBm, -32dBm]
noise floor	-80dBm
path loss coeff	4.9
# pulses for intruders	[10, 15]
window size(for doing FFT)	10μs

Schemes Compared. We compare our `Group-Based` Algorithm with `Naive` (§II) and a variant of `Naive` algorithm called `Naive+aligned` algorithms which differs from `Naive` in that it is given *fully aligned* observation sequences of sensors. Note that `Naive+aligned` is not a fair algorithm for comparison, since it is essentially given a fully-aligned input, but our purpose behind comparing with `Naive+aligned` is to separate the reasons behind the good performance of `Group-Based` Algorithm.

A. Real Testbed Experiments

We set up a small test-bed with two transmitters and four sensors in a university campus lab. We use RTL-SDR [13] and USRP [12] boards as sensors and transmitters respectively. The transmitters generate an intermittent tone of 10μs in 915MHz ISM frequency band. The sensors are tuned this center frequency with a sampling rate of 1Msps. The key result from the test-bed experiments is that the average skew among these sensors is as high as 100μs. We next study the localization performance using these skewed sensors. Table I shows the average localization performance in terms of error, miss and false alarm. The results shows that our `Group-Based` Algorithm performs significantly better than `Naive` algorithm.

B. Large-Scale Simulations

We study the performance of our `Group-Based` Algorithm on a large-scale simulation platform based on synthetic data for more practical scenarios such as large number of sensors, intruders, and varying values of skew. To this end, we create simulator that operates on a grid of 50m×50m environment with parameters as described in Table II.

Clearly, the performance of a localization is a function of key factors such as the amount of skew among the sensors, the number of intruders and sensors in the field, and the range of number of intruders given to the `Naive` schemes (note that `Naive` schemes depend on k-means clustering, and require the range of number of intruders as an input). We experiment by varying each of these parameters to understand their

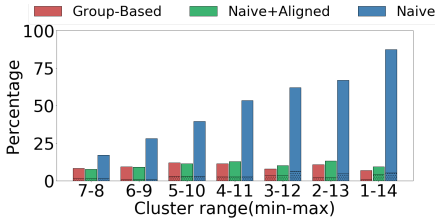

Fig. 3: Percentage of miss and false alarms vs. ranges of transmitters input.

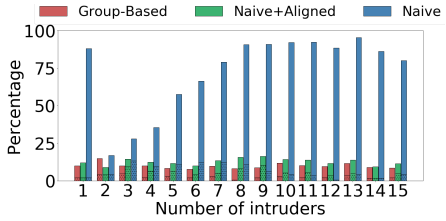

Fig. 4: Percentage of miss and false alarms vs. different number of intruders.

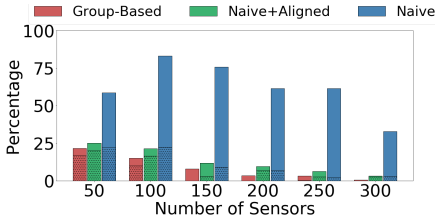

Fig. 5: Percentage of miss and false alarms vs. different number of sensors.

impact on localization performance. For all the experiments, the intruders transmit at least 10 pulses with random length between 20-30μs. We use Poisson distribution of $\lambda = 8$ for arrival rate of pulses.

Varying Skew. Figure 2 shows the localization performance against different skew values from 10-90μs. The key takeaway here is that `Group-Based` Algorithm performs significantly better than two other algorithms. For example, irrespective of the skew, `Group-Based` Algorithm results in no more than 10% of misses and false alarm together. On the other hand, the miss and false alarm performance of the `Naive` algorithm increased from 20% to to more than 80% when the skew is increased from 10μs to 90μs. Note that the performance of the `Naive+aligned` algorithm is also slightly worse than `Group-Based` Algorithm, which suggests that, in addition to handling misaligned observations, `Group-Based` Algorithm's method of localization is also superior than that of `Naive` algorithm.

The localization error of all schemes is almost similar, with `Group-Based` Algorithm still performing slightly better than the other schemes. This is likely because the underlying localization approach is the same for all the methods. Also, note that `Group-Based` Algorithm specifically targets to improve the accuracy in detecting the *number* of intruders. For the following experiments, the localization error plot show the similar relative performance, thus, we do not show the localization error plots, in interest of space.

Varying Input Range of number of Intruders, Number of Intruders, and Sensor Density. Figure 3 shows the percentage of miss and false alarms under different ranges of transmitters as input. As described in §II, the performance of the `Naive` algorithm gets significantly affected if we do not provide enough information about the number of intruders. On the other hand, `Group-Based` Algorithm consistently achieves less than 10% miss and false alarms together even if the input cluster range is high as 1-14. These significant benefits come from filtering the individual (or minimal-sized sets of) intruders and localizing them independently (see §III).

Finally, Figure 4 and 5 show the performance for varying number of intruders and sensors with 50μs skew. As expected, the `Naive` algorithm suffers significantly with more intruders and fewer sensors. For example, the miss and false alarm percentage is more than 75% and 80% with 15 intruders and 100 sensors respectively. Even with 300 sensors, the `Naive` algorithm results in more than 30% miss and false alarm percentage. In contrast, our proposed `Group-Based` Algorithm needs as few as 50 sensors to localize all the intruders with the miss and false alarms percentage at most 25%, which comes down to less than 4% with the increase in the number of sensors to 300.

V. Conclusion

In this paper, we have addressed the problem of multiple-intruder localization in face of time-skewed sensor observations; to the best of our knowledge, ours is the first work on this problem, which arises naturally in the context of crowd-sourced spectrum monitoring using inexpensive distributed sensors. We believe that our work can have implications on better provisioning and efficient deployment of spectrum sensors for spectrum patrolling. In our future work, we plan to address other challenges and problems that arise due to time-skewed observations by spectrum sensors, e.g., in creation of spectrum occupancy maps in a dynamically changing spectrum [17], real-time localization, and localization of mobile intruders.

References

[1] K. Zeng *et al.*, "All your GPS are belong to us: Towards stealthy manipulation of road navigation systems," in *USENIX Sec. Symp.*, 2018.

[2] H. Sathaye *et al.*, "Wireless attacks on aircraft instrument landing systems," in *USENIX Security Symposium*, 2019.

[3] M. Khaledi *et al.*, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *ACM MobiCom*, 2017.

[4] M. Group *et al.*, "Microsoft spectrum observatory," 2013.

[5] A. Chakraborty *et al.*, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM*, 2017.

[6] A. Chakraborty *et al.*, "Spectrum patrolling with crowdsourced spectrum sensors," in *IEEE INFOCOM*, 2018.

[7] M. Dasari *et al.*, "Spectrum protection from micro-transmissions using distributed spectrum patrolling," in *PAM*, 2019.

[8] M. Lévesque and D. Tipper, "A survey of clock synchronization over packet-switched networks," *IEEE Comm. Surveys & Tutorials*, vol. 18, no. 4, 2016.

[9] M. Lipiński *et al.*, "White rabbit: A ptp application for robust sub-nanosecond synchronization," in *IEEE PCSMCC*, 2011.

[10] R. Calvo-P, D. G, V. L, and A. F, "Crowdsourcing spectrum data decoding," in *INFOCOM*, IEEE, 2017.

[11] A. Fakhreddine, *Opportunistic timing signals for pervasive mobile localization*. PhD thesis, Universidad Carlos III de Madrid, Spain, 2018.

[12] U. B210, "https://www.ettus.com/product/details/ub210-kit."

[13] RTL-SDR, "https://osmocom.org/projects/rtl-sdr/wiki/rtl-sdr."

[14] ODROID-C2, "https://wiki.odroid.com/odroid-c2/odroid-c2."

[15] A. Nika *et al.*, "Empirical validation of commodity spectrum monitoring," in *ACM SenSys*, 2016.

[16] J. Nelson *et al.*, "A quasi EM method for estimating multiple transmitter locations," *IEEE Signal Processing Letters*, 2009.

[17] M. S. Rahman, H. Gupta, *et al.*, "Creating spatio-temporal spectrum maps from sparse crowdsensed data," in *IEEE WCNC*, 2019.