ORAN-Sense: Localizing Non-cooperative Transmitters with Spectrum Sensing and 5G O-RAN

Yago Lizarribar*†, Roberto Calvo-Palomino‡, Alessio Scalingi*†
Giuseppe Santaromita*, Gérôme Bovet§, Domenico Giustiniano*

*IMDEA Networks, Spain
†Universidad Carlos III de Madrid, Spain
‡Universidad Rey Juan Carlos, Spain
§armasuisse, Switzerland

***Abstract*—Crowdsensing networks for the sole purpose of performing spectrum measurements have resulted in prior initiatives that have failed primarily due to their costs for maintenance. In this paper, we take a different view and propose ORAN-Sense, a novel architecture of Internet of Things (IoT) spectrum crowdsensing devices integrated into the Next Generation of cellular networks. We use this framework to extend the capabilities of 5G networks and localize a transmitter that does not collaborate in the process of positioning. While 5G signals can not be applied to this scenario as the transmitter does not participate in the localization process through dedicated pilot symbols and data, we show how to use Time Difference of Arrival-based positioning using low-cost spectrum sensors, minimizing hardware impairments of low-cost spectrum receivers, introducing methods to address errors caused by over-the-air signal propagation, and proposing a low-cost synchronization technique. We have deployed our localization network in two major cities in Europe. Our experimental results indicate that signal localization of non-collaborative transmitters is feasible even using low-cost radio receivers with median accuracies of tens of meters with just a few sensors spanning cities, which makes it suitable for its integration in the Next Generation of cellular networks.**

***Index Terms*—O-RAN, 5G, Localization, Spectrum Sensing**

I. Introduction

Wireless communication infrastructures are growing at a frantic pace [1], and this trend will continue with the ongoing deployments of 5G networks. As RF spectrum is scarce, monitoring it on a large scale is fundamental for gaining insights into the Electromagnetic (EM) spectrum usage and its evolution, as well as detecting threats attempting to cause disruption. Network providers spend billions of dollars in spectrum auctions, making the problem of protecting their spectrum and localizing non-cooperative transmitters, a key pillar to running their business. Attacks on the spectrum can also be quite common in areas of conflict: they can target general communications like stated in [2].

However, localizing non-cooperative transmitters is a difficult task, and it has been so far decoupled from 5G deployments. Current methods require network operators to deploy temporary, bulky, and expensive equipment if an anomaly in spectrum usage has been found around a particular area. This method has several disadvantages, as it is impossible to continuously monitor the spectrum and operational costs are high. The advent of IoT devices in the last years equipped with Software Defined Radio (SDR) capabilities has significantly influenced the approach to spectrum monitoring, as they can establish a network of low-cost distributed sensors that transfer data in the cloud for processing, revolutionizing the spectrum monitoring and breaking down cost barriers.

In the past few years, several crowdsourced spectrum monitoring solutions have emerged using low-cost devices and custom backends, such as Microsoft Spectrum Observatory [3] and Electrosense [4]. However, the maintenance costs of such platforms, particularly single-purpose dedicated backends, are often too high, and these initiatives do not last long. Several of these initiatives have shut down after years of effort.

Rather than taking the traditional approach to spectrum crowdsensing of having a third party that processes spectrum data in its dedicated backend, we propose utilizing the backend infrastructure of the Radio Access Network (RAN) and maintained by the network operator themselves. We envision that IoT spectrum sensors will be deployed at users' facilities and directly controlled by network operators. This is similar to femtocell technologies adopted by network operators where existing broadband connections are used to connect small form-factor base stations to the operator Core Network [5]. This way, network operators would have access to sites for deploying spectrum sensors that would not be available without using a crowdsensing paradigm. On the other hand, the users will benefit from access to similar services as in prior initiatives (e.g., decoding public signals and their audio and visual representation [6]). The resources in the backend would now be shared among several software services following the 5G paradigm, significantly cutting operational costs.

To transform this idea into practical realization, we leverage the fact that the next generation of the cellular network is going under a radical evolution as the current architecture will be replaced by programmable and virtualized infrastructure, including the RAN [7], [8]. The Open Radio Access Network (O-RAN) is the primary driver behind this deployment as it promotes softwarization, disaggregation, openness, and programmable hardware [9]. Furthermore, the O-RAN Alliance [10] is currently defining the concept of the O-RAN paradigm to streamline the implementation, testing, and de-

ployment of flexible solutions at scale. One of the peculiarities of O-RAN is the introduction of RAN Intelligent Controller (RIC), programmable components that can run optimization routines to control and orchestrate the RAN effectively. Our idea is to leverage the open interfaces and RICs to perform the localization of non-cooperative transmitters, enriching cellular networks' capabilities that can already localize cooperative transmitters through 5G signals [11].

In summary, this paper proposes merging crowdsensing networks with the O-RAN interfaces to make a robust system that leverages existing infrastructure to monitor the spectrum and collect data to perform more advanced analytics tasks. The contributions of this paper are the following:

- A novel framework to integrate IoT devices into O-RAN networks to perform spectrum monitoring tasks at a more sustainable cost than in prior initiatives.
- We develop ORAN-Sense, an architecture that leverages this framework and can use these IoT spectrum sensors to localize uncooperative transmitters within a city.
- We provide an experimental evaluation, with sensors deployed in the wild in 2 different European cities, and show that it is possible to build and use such a network.

The remainder of this paper is organized as follows: Section II provides background insights into O-RAN and its interfaces. Section III proposes an architecture that addresses the challenges in the design and the detailed steps to perform geolocation at scale, and Section IV provides an evaluation of the different components of our architecture. Section V dives into O-RAN and its use cases and discusses the main localization approaches described in the literature. Last, Section VI provides the conclusions.

II. O-RAN Primer: Components and Interfaces

This section introduces background knowledge on the specific components of O-RAN networks and how the design can support the entities for localization.

A. Components

An overview of the general architecture is depicted in Fig. 1. O-RAN incorporates the 3GPP NR 7.2 logical split [12], dividing base station functionalities at different network locations. For instance, the CUs and DUs can be virtualized and deployed on different servers; meanwhile, the RUs are placed in proximity to the RF antennas [7].

Service Management and Orchestration Framework (SMO). There are numerous management domains in a service provider's network. The main goal of this component is to manage and orchestrate the RAN domain [13], e.g., Cloud Management, Orchestrator, and Workflow Management and resources optimization with RICs.

Collector in the SMO domain. The SMO hosts the Non-Real-Time RIC (Non-RT RIC). However, there might be other network functions that may provide additional services. Notably, the standard does not establish a formal boundary with the Non-RT RIC and other logical functions within the SMO framework [14]. In this context, an Operations and Maintenance (OAM) Function can be used to manage components, like the RIC or the connection with external services (e.g., Cloud) [15], that we call Collector. It can gather information from external services, such as custom back-end external applications, and interact with controllers to deploy or update RAN policies.

Non-RT RIC. The Non-RT RIC is another key component of O-RAN enabling closed-loop control of the RAN with timescales larger than 1 s. Its main goal is to optimize the RAN intelligently by providing policy-based guidance and enrichment information to the Near-Real-Time RIC (Near-RT RIC) function. This is achieved through the standard open interfaces, using declarative policies expressed with formal statements, allowing Non-RT RIC to guide Near-RT RIC for specific optimizations, such as localization algorithms or particular areas and frequencies to explore.

Near-RT RIC. Like the Non-RT RIC, the Near-RT RIC enables control and optimization of RAN functions and resources in near-real-time. It facilitates data gathering and actions via the E2 interface, with a control loop ranging from 10ms to 1s. One or more *xApps*, i.e., AI Models or algorithms, can be hosted by the Near-RT RIC, and the E2 interface enables a direct association between xApp and E2Node functions to consume the data.

B. E2 Interfaces, E2 Node and Service Model

The components introduced in Section II-A are connected with open interfaces, following the specifications established by the O-RAN Alliance.

The E2 Interface is the open interface between two endpoints, the Near-RT RIC and the E2Nodes. It enables the collection of metrics from the RAN to the Near-RT RIC, either periodically or after pre-defined trigger events.

An E2Node, comprising DUs and CUs, is a logical node and offers a variety of *RAN functions*, i.e., services or capabilities, and supports the connection with E2 Interface [16]. For example, DUs from different vendors might provide their ability to gather and report varying performance metrics. During the connection, the E2Node (the gNB) exposes the RAN Functions to which the Near-RT RIC can subscribe.

The E2 Service Model (E2SM) in the O-RAN E2 interface is a protocol detailing the E2Node's capabilities to the Near-RT RIC. It is a list of *RAN function* offerings that the E2Node makes available, including data collection definitions. The communication between an E2 Node and Near-RT RIC uses a publish-subscribe paradigm. The Service model defines whether the reporting is periodic or trigger-based.

Currently, the standard defines guidelines to monitor the Key Performance Measurements (E2SM KPM) provided to the xApps [16]. However, these are limited upper-layer metrics to optimize scheduling policies or RAN slicing [9]. In this paper, we propose enhancements with *a novel Service Model that enables the E2 interface to transmit localization-specific RF measurements from IoT devices*, broadening the service's utility model.

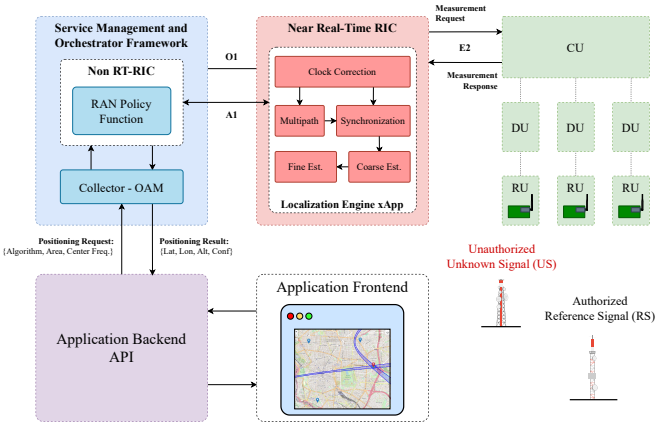


Fig. 1: Proposed architecture for ORAN-Sense, a Time Difference of Arrival (TDOA)-based localization approach of non-cooperative transmitters that can be embedded in O-RAN by using its open interfaces.

III. ORAN-SENSE

The architecture of ORAN-Sense is shown in Fig. 1. We propose that low-cost sensors act as E2Nodes, consisting of a commercial off-the-shelf antenna, an RTL-SDR, and a Raspberry Pi (Fig. 2). Like in femtocell technology [5], network operators connect these sensors in different locations like homes or offices via Ethernet or wirelessly to increase the monitoring coverage. Localization starts in the Non-RT RIC by declaring the policy of the area and the frequencies of interest. The policy commands the E2 nodes, specifically those DUs equipped with sensors, that expose the network function to sense and gather In-Phase and Quadrature (IQ) measurements for a specific short time. Then, the data are consumed by the localization xApp within the Near-RT RIC. The localization stages are explained in the following paragraphs.

The localization architecture proposed in this paper is divided into different parts: the **Sensor** that we consider as the entity within the E2Node with only reception capabilities, the localization **xApp** that runs in the Near-RT RIC and the **SMO** which contains a campaign manager.

Spectrum sensors only measure IQ samples with a 2 MHz bandwidth (larger bandwidths are possible with higher-performing SDRs), tuned to the center frequency of the band to be measured. This also means that if the signal of interest has a larger bandwidth than the sampling rate of the RTL-SDR front-end, information outside this band is not used by the RTL-SDR-based spectrum sensor for localization purposes. As our system does not perform any decoding of the reference and target signals, this issue is irrelevant in the context of localization of non-cooperative transmitters.

To transmit the IQ measurement command, we use the E2 interface defined by the O-RAN. When the measurement command is received, sensors first estimate their local oscillator offset, similarly to [17]. After the given measurement time elapsed, these sensors send IQ samples and their corresponding local oscillator offsets, measured in Parts Per Million (PPM).

Fig. 2: The RU is comprised of a Raspberry Pi and an RTL-SDR

Once data is gathered in the Near-RT RIC, the localization process is subdivided into the following steps: *Frequency Offset Correction*, *Synchronization and TDOA Estimation* and *Multilateration*. More details are provided in the remainder of this section. We conclude the section by providing technical details on the O-RAN integration.

A. Frequency Offset Correction

At operating temperatures, SDR receivers generally have constant drifts in their clocks. For example, the latest versions of the RTL-SDR are guaranteed to have ± 1 PPM, but this drift can be even higher when it is heating up [17]. We cannot assure that the SDR will be at the correct operating temperature at any given time, so we need to estimate the drift accordingly. The following sections show that the clock drift can significantly influence the measurement results.

Our system corrects both the center and the sampling frequency offsets. We assume that in the IoT receiver, a single Local Oscillator (LO) feeds both the down-conversion and sampling stage. Therefore, we assume that both errors in the center and sampling frequencies are similar:

$$\frac{\Delta f_s}{f_s} \approx \frac{\Delta f_c}{f_c} \approx \varphi \tag{1}$$

where f_s, f_c and φ are the sampling frequency, center frequency and PPM offset respectively.

To correct the center frequency, f_c, we must note that the actual measured center frequency, f_c', is:

$$f_c' = (1+\varphi) \cdot f_c \tag{2}$$

where $\Delta f_c = \varphi f_c$. Each sample is received at time $n \cdot T_s'$, where T_s' is the observed sampling period, which can be related to the desired sampling period, T_s, as $T_s' = T_s/(1+\varphi)$.

We must perform a frequency shift of the samples of the measured signal denoted as $s'[nT_s']$. We can then generate the new signal, shifted to the desired center frequency:

$$s''[nT_s'] = s'[nT_s'] \cdot e^{-j2\pi \cdot nT_s' \cdot \varphi f_c} \tag{3}$$

Now that the center frequency f_c is corrected, we can correct the sampling frequency f_s. If the received signal were continuous, it would be enough to take the samples at the time instants corresponding to the desired sampling rate. As the received signal is not continuous, we re-sample the data at the correct rate by interpolating samples. Thus the corrected

signal $s[nT_s]$ can be regarded as changing from sampling rate $nT_s/(1+\varphi)$ to nT_s:

$$s[nT_s] = I(s''[nT_s'], \frac{nT_s}{1+\varphi}, nT_s) \tag{4}$$

B. Synchronization and TDOA Estimation

TDOA based systems require sensors to be time synchronized. One of the most standardized manners to synchronize the clocks over the Internet is to use the Network Time Protocol (NTP), which can achieve accuracies of a few milliseconds. However, for a localization system, this is not satisfactory.

Some sensor networks rely on GPS as their source of synchronization. There exist commercial solutions offering errors of around 50 ns. However, *the sensor cost would increase drastically*, as commercial solutions providing high accuracy are in the order of more than 1,000.00 USD[1]. We do not intend to rely on any GPS source, so even the lower end of SDR receivers like the RTL-SDR can be utilized. We also note that sensors cannot use fine network synchronization, as they are connected to the CU only through standard Internet connectivity. To solve this challenge, we propose an approach that utilizes NTP to coordinate the sensing processes of different sensors and then uses a Reference Signal (RS) as a signal of opportunity for synchronization, and the Unknown/Target Signal (US), which we aim to locate.

When a signal is transmitted and captured by one spectrum sensor, the model for the received signal time is:

$$t_i = t_0 + \frac{||\mathbf{x_i} - \mathbf{x_t}||}{c} + \delta_i \cdot t + \theta_i + \eta_i = t_0 + \frac{d_i}{c} + \delta_i \cdot t + \theta_i + \eta_i \tag{5}$$

where t_0 is the time at which the original signal is transmitted, $\mathbf{x_i}$ and $\mathbf{x_t}$ are the position of the receiver and transmitter respectively (distance that we can represent by d_i), δ_i and θ_i are the drift and time offset associated with the receiver i, and η_i is the noise. Since we correct the frequency offset (see Section III-A), the remaining drift is negligible ($\delta_i \approx 0$), so we can remove it from the model.

For each pair of IoT receivers, their sampled signals are sent to the Localization Engine xApp, where they are subtracted from one another. In this way, we can remove the transmit time from the equation:

$$\tau_{ij} = t_i - t_j = \frac{d_i}{c} - \frac{d_j}{c} + (\theta_i - \theta_j) + \eta_{ij} \tag{6}$$

We multiply both sides by the speed of light c and denote $d_{ij} = d_i - d_j$ as the distance difference between receivers. We can apply this to both the RS and US:

$$\begin{aligned} c \cdot \tau_{ij}^{us} &= d_{ij}^{us} + c \cdot (\theta_i - \theta_j) + c \cdot \eta_{ij}^{us} \\ c \cdot \tau_{ij}^{rs} &= d_{ij}^{rs} + c \cdot (\theta_i - \theta_j) + c \cdot \eta_{ij}^{rs} \end{aligned} \tag{7}$$

We can subtract one from the other, and then by knowing the distance difference from sensors i and j from the reference transmitter, we can isolate the desired TDOA value:

$$c \cdot (\tau_{ij}^{us} - \tau_{ij}^{rs}) = d_{ij}^{us} - d_{ij}^{rs} + c \cdot \eta_{ij}^{rs,us} \tag{8}$$

[1]https://www.ettus.com/all-products/gpsdo-mini/

If we now add the distance to the RS, which is known, we are only left with the TDOA associated with our target transmitter:

$$c \cdot \tau_{ij} = d_{ij}^{us} = c \cdot \tau_{ij}^{us} - (c \cdot \tau_{ij}^{rs} - d_{ij}^{rs}) + c \cdot \eta_{ij}^{rs,us} \tag{9}$$

To obtain the delay between two signals s_i, s_j, we calculate the maximum delay τ_{ij} of their cross-correlation:

$$\hat{\tau}_{ij} = \arg\max_{\tau} \left[(s_i \star s_j)[\tau] \right] \tag{10}$$

Traditionally, only raw IQ samples are used for performing cross-correlation. In this paper, we instead evaluate three different possibilities to perform the cross-correlation: using the raw IQ samples (`iq`), using the absolute value (`abs`) or the phase difference (`dphase`). We present these correlation methods in Table I. Note that we rely on the I and Q samples measured by the RTL-SDR front-end in all methods.

One issue to consider relates to the quantization error one might have in a single sample when computing the TDOA. For the RTL-SDR, given that we sample at around 2 MSamples/s, that would mean that a difference of one sample can result in an error of 150 m. As studied in detail in the evaluation section, we propose to mitigate the uncertainty region caused by the low sampling capabilities of low-cost IoT sensors by upsampling the signals and computing the cross-correlation on the upsampled versions.

C. Multipath mitigation

In urban scenarios, signals are expected to be either reflected or entirely occluded by buildings, trees, or any other type of obstacle. This directly affects a time-based localization system like the one proposed in this paper. When sensors are affected by multipath, they receive several copies of the signal at different times. Choosing the wrong copy might result in errors of tens or even hundreds of meters.

Under multipath and Non-Line of Sight (NLOS) conditions, the received signal by a spectrum sensor i with L_i different paths can be modelled as:

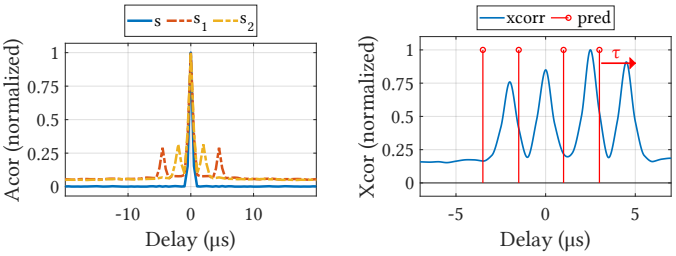
$$s_i = \sum_{l=1}^{L_i} \alpha_{i,l} \cdot s_{i,l}(t - \tau_{i,l}) + \eta_{i,l} \tag{11}$$

where $\alpha_{i,l}$ is the attenuation factor of the lth multipath component and $s_{i,l}$ are the time-delayed versions associated to that component, and $\eta_{i,l}$ is the noise in the measurement.

Differently from TOA measurements, the reflected peaks in TDOA measurements may also appear before the Line-of-Sight (LOS) peak, which further hinders identifying the shortest path. Thus, the naive approach of selecting the highest peak

Name	Computation Method
`iq`	$I[nT_s] + jQ[nT_s]$
`abs`	$\sqrt{I[nT_s]^2 + Q[nT_s]^2}$
`dphase`	$\angle s_{i,iq}[nT_s] - \angle s_{i,iq}[(n-1)T_s]$

TABLE I: Correlation methods studied in this paper.



(a) Autocorrelation (b) Cross-Correlation

Fig. 3: Effect of multipath on cross-correlation between 2 signals. On the left, we have the autocorrelation of each one. The signal affected by multipath will present multiple peaks. On the right is the cross-correlation between both. If the indirect path is stronger (as in this case), we might select the erroneous peak (instead of the one at 0 μs), leading to higher errors in the location estimation.

would not work. The effect of these multiple components on the TDOA estimation can be observed on the *autocorrelation* of signals with the appearance of peaks at times $(0, \pm m_{i,1}, \pm m_{i,2}, \ldots, \pm m_{i,L})$. If signals received from two sensors i and j are affected by multipath, then their *cross-correlation* function appears generally with $L_i \times L_j$ peaks, where L_i and L_j are the numbers of paths received by sensor i and j, respectively (see also the example in Fig. 3).

We estimate the multipath components per sensor from the autocorrelation function (see Fig. 3). This results in an array of the components for sensor i: $[0, m_{i,1}, \ldots, m_{i,L_i}]$. By combining all possibilities for a pair of sensors $\{i, j\}$, we generate a hypothesis of the generated multipath components at a given estimate for the direct path component delay τ:

$$C(\tau) = [\tau - m_{j,L_j}, \ldots, \tau + m_{i,L_i}]; \quad \forall m_{i,l_i}, m_{j,l_j} \tag{12}$$

Then, computing the cross-correlation function between these two sensors, we can obtain a set of the predicted peaks $\hat{C}$. The correct delay τ_{ij} for a pair of sensors $\{i, j\}$ will be such that it minimizes the distance between the hypothesis and the measured vectors:

$$\tau_{ij} = \arg\min_{\tau} ||C(\tau) - \hat{C}|| \tag{13}$$

To visualize the effect of multipath on TDOA measurements, we provide a small example. Let us suppose we receive signal s from 2 receivers (s_1,s_2), and each one has another delayed component by 9 and 4 μs, respectively. To detect those extra components, we rely on the auto-correlation function (Fig. 3a). Receiver 1 will have components $[0, 9]$ and receiver 2 $[0, 4]$. If we compute the cross-correlation of both signals, peaks should appear at $[-4 + \tau, \tau, 5 + \tau, 9 + \tau]$, where τ is the TDOA. We can now perform the cross-correlation between both signals and find the peaks (Fig. 3b). We can then apply the minimization routine in Eq. 13 and solve it with a brute-force algorithm.

D. Multilateration

In the last component of the architecture, the actual transmitter positioning is performed. Starting from a weighted sum of squared errors between the measured and the expected TDOA:

$$S(\mathbf{x_t}) = \sum_{\forall i,j} w_{ij} \cdot (\hat{\tau}_{ij} - \tau_{ij})^2 = \sum_{\forall i,j} w_{ij} \cdot r_{ij}^2 \tag{14}$$

where each term of the summation corresponds to the TDOA value per each sensor pair i, j, and w_{ij} is the weight applied to each of the measurements. Note that if $w_k = 1$, then this becomes the L2-norm. For this method, the solution lies at the point where the gradient of $S(\mathbf{x_t})$ is minimum, which can be computed per transmitter coordinate (x_t, y_t) as:

$$\begin{aligned} \frac{\partial S}{\partial x_t} &= -2 \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{x_t - x_i}{d_i} - \frac{x_t - x_j}{d_j} \right) \\ \frac{\partial S}{\partial y_t} &= -2 \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{y_t - y_i}{d_i} - \frac{y_t - y_j}{d_j} \right) \end{aligned} \tag{15}$$

These equations can then be solved by several methods like Gradient Descent, Levenberg-Marquardt, or BFGS, among others [18]. Referring to Fig. 1, we use linear methods as an initial coarse estimate for the position and then fine-tune them with non-linear approaches.

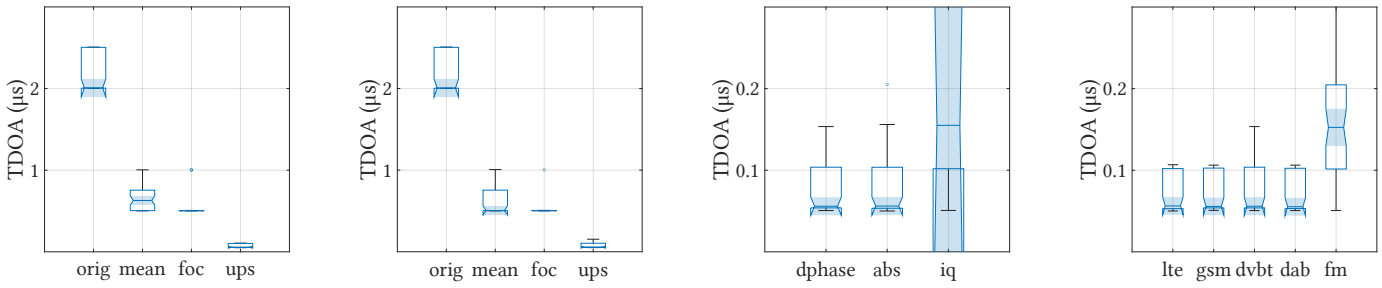
E. Localization embedded into ORAN

The localization procedure starts with a request the Collector receives from another automated RAN function, like anomalous spectrum usage with technology classification [19] or a request by the network's operator through an external service. The request must specify the number of sensors (3 for 2D localization, 4 for 3D), the algorithm, the area, the duration, and the frequencies to inspect (namely, the ones of the reference signal and the unknown one). The collector forwards the information to the Non-RT RIC, requesting RAN data for localization. The Non-RT RIC formulate the *RAN Localization Policy* delivered to the Near-RT RIC through the A1 Interface.

The Near-RT RIC subscribes to the E2Service model provided by the E2Nodes, specifically the CUs or DUs in the area of interest, specifying interested frequencies and trigger-based report measurement. The latter is managed by a timer set within the E2Node.

Upon timer activation, the IoT devices designed for spectrum sensing within the DUs (or in the RUs) tune to the designated frequencies, collecting the necessary measurements that will be packaged in the E2 Report Message. At the expiration of the timer, the E2 Node publishes the localization measurements over the E2 Interface, consumed by the xApp. In the proposed design, *the measurements are not transmitted as a continuous data stream, avoiding overwhelming the network*. The xApp runs the localization algorithm, e.g., TDOA-based algorithm, with the measurements collected over different locations. The outcome of the TDOA localization is forwarded to the Non-RT RIC that either notifies the collector and the External Service along with positioning coordinates.

In our measurements with real data, the highest computational time is spent in the correlation phase. On average



(a) GSM (b) DVB-T (c) Correlation methods (d) Different technologies

Fig. 4: Feasibility study results. The two figures on the left describe the results with GSM and DVB-T, respectively. We observe that by applying frequency-offset correction (`foc`) and upsampling (`ups`), we can obtain synchronization errors lower than 100 ns. In (c), we can see the differences between the proposed correlation methods with DAB signals, and in (d), we compare the different technologies that are present in urban environments.

on a intel i5 laptop we can obtain all correlation pairs in less than 450 ms, with at least 300.000 samples per pair. Multilateration with 6 sensors (the highest number in our scenario) takes slightly less than 50 µs on average. Adding these numbers we observe that we are well below the threshold for the Near-RT RIC, making our approach feasible from the standard's perspective.

IV. Evaluation

This section describes the evaluation of the system using realistic scenarios. To evaluate the system, we define three sets of experiments. The first is the feasibility studies, and we look at how frequency offset correction, upsampling, and different signals could affect the system's performance. In the second set, we evaluate the real-life applicability of different signals in our localization architecture. Last, we study the performance of the different multilateration algorithms using real data from our deployments.

A. Reference Signal Suitability

For this part of the evaluation, we use three RTL-SDR v3 equipped with TCXO and collect samples for four different technologies: Long-Term Evolution (LTE), Global System for Mobile Communications (GSM), Digital Video Broadcasting-Terrestrial (DVB-T), and FM radio. We need to collect reference and target signals in the same trace. To achieve this, we modify *librtlsdr* [20] to allow for a center frequency change while sampling. By continuously sampling data, we avoid losing samples between sensing processes, and thus we can obtain the actual synchronization delay between a pair of sensors. We first tune to the RS, then US, and back to the RS, splitting traces into 3 **chunks** with the same number of samples. We collect 200 traces for each receiver with 3 seconds of data, 1 per signal chunk. We also collect one second of LTE traces for each measurement to calculate the oscillator offsets. The experimental setup is summarized in Table II.

To collect data in this first set of experiments, we connect three RTL-SDR devices to an active splitter, an antenna, and an external power supply. The three SDR devices are then connected to the same machine.

Measuring with this setup, the signal is received by only one antenna, and the differences in the measurements are due only to their internal clock offsets or other hardware imperfections. After collecting data, we analyze all the traces and show the different results in Fig. 4. Since the antenna is the same for all sensors, the observed differences should be 0. Values different than that might be caused due to the offset between sensors not being properly corrected or the signal being too noisy to be suitable for a reference signal.

1) Effects of Offset correction and Upsampling: Figures 4a and 4b show the results of the feasibility studies using GSM and DVB-T respectively in four scenarios: when no offset correction is made (`orig`) when we use the offset in both chunks that contain RS and average the delay (`mean`), when we correct the local oscillator offset (`foc`) and when we correct offset and upsample the signals (`ups`).

In these plots, we can observe that even though the LO errors are relatively small (in our experiments, we do not observe any values above 1 PPM, which matches the specification for the selected RTL-SDR), they can have a great influence on the TDOA values, and thus these errors must be corrected.

One approach to correct these errors is to assume the offset between two sensors is linear, which is not always a reasonable assumption since their oscillator errors can be unstable depending on the conditions. We can, however, have a coarse estimate by averaging the offset between both RS chunks and subtracting it from the delay obtained with the US chunk, and we can improve the results notably in all cases

RS	f_{RS} (MHz)	US	f_{US} (MHz)	Bandwidth (MHz)
LTE	816	LTE	806	2
GSM	938.8	LTE	806	0.2
DVB-T	562	LTE	806	2
DAB	196	LTE	806	1.5
FM	98.8	LTE	806	0.2

TABLE II: Experiment setup for the feasibility studies. We vary the reference signal (RS), and we use as an unknown signal the LTE signal at 806 MHz.

(a) Deployment in City 1 (b) Deployment in City 2

Fig. 5: Sensor locations for the benchmark. The target transmitter (Unknown) is marked in blue and the Reference in pink. For the case of City 1, both Target and Reference are the same.

(this is the method `mean` on Fig. 4).

If we perform the offset correction, we can reduce the delays obtained in almost all experiments to 0. However, using a sampling rate of 2 MSamples/s, our computation is affected by the quantization error of 150 m, which still represents a high uncertainty region. To observe where the actual cross-correlation peaks occur, we upsample the signals with different factors (in these figures, we use a factor of 10). We observe a finer synchronization thanks to this approach, with around 60 ns median error.

2) Different Correlation methods: Fig. 4c compares the three different correlation methods explained in Section III: `dphase`, `abs`, and `iq`. For clearness, we show only the results for Digital Audio Broadcasting (DAB) signal, but the results are very similar to other technologies. From this figure, we can see that `abs` and `dphase` have similar performances (although the former is slightly better), whereas the traditional `iq` method performs poorly on all the analyzed technologies. One of the reasons that explain the worse performance of `iq` method is the fact that it relies on both the magnitude and the phase of the samples, and phase alone is not a suitable descriptor in non-coherent sensor networks.

3) Comparison of technologies: The performance comparison of all the technologies analyzed is shown in Fig. 4d. From these results, we observe that digital signals offer the best performance and FM signals the worst. One explanation is that the former are digital signals with accurate amplitude information and richer phase information, whereas FM radio signals are analog with coarse amplitude information and no information encoded on the phase.

Furthermore, LTE, GSM, DVB-T and DAB have similar performance. We notice that LTE and DVB-T have larger bandwidths with respect to the IQ sampling rate of RTL-SDR front-end (spectrum sensors only measure IQ samples in 2 MHz bandwidths, with frequency tuned to the center frequency of the Reference and Unknown Signals), while the bandwidth of DAB is of about 1.5 MHz (thus it does not exploit fully the RTL-SDR capabilities). Furthermore, there are fewer TV and Digital Radio transmitters, so it is easier to guarantee that spectrum sensors capture signals from the same tower.

B. Deployments

For these experiments, we deploy sensors with the same characteristics as described in Fig. 2 in two different cities: City 1 and City 2 (Fig. 5). The maximum distance between sensors is 20 km and 15 km, respectively. To evaluate the performance of the full architecture, we select existing DAB transmitters as our target frequencies due to the high transmission power and well-defined locations, thus making them appropriate to benchmark the proposed system. For the sensors, we implement an extension to `es_sensor` C++ package from ElectroSense [6]. The resulting platform is similar to the one depicted in Fig. 2. For the localization backend, we implemented versions in MATLAB© and Python.

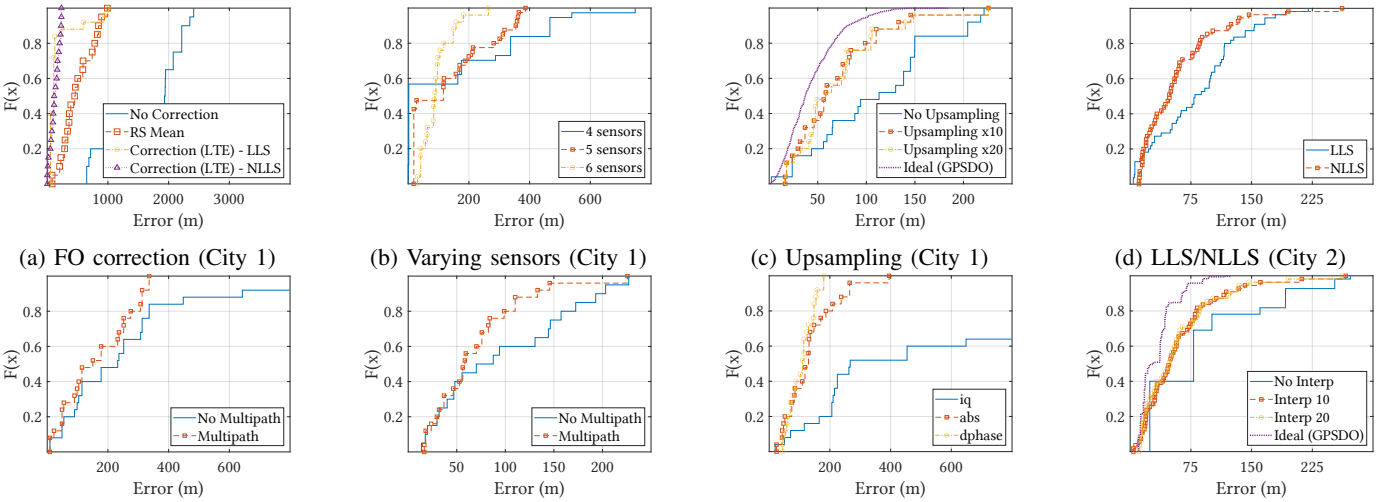
1) City 1 deployment: A total of six IoT sensors are deployed in the City 1 region, each having RTL-SDR v3 as their SDR frontend. The target transmission is the DAB signal with a center frequency of 208 MHz.

As our reference signal, we use the DAB transmission centered at 196 MHz coming from the same transmitter. Due to the limited number of DAB transmitters in the area of deployment, no other transmitters are interfering. By having reference and target on the same location, we can rule out inaccurate locations of transmitters. All the errors should be due to the limitations of the receivers and our approach. In total, up to 70 recordings over several months are gathered, each consisting of 1.5 seconds of data per sensor (divided into three continuous chunks containing RS, US, and RS respectively).

We first compare the results when we correct the frequency offset, Linear vs Non-Linear approaches, and present the results in Fig. 6a. We observe that each step of our approach increases the overall accuracy of our system quite significantly. We observe that no offset correction implies km-level errors even though the transmitter is the same. If we make a rough estimate of the offset by averaging the two RS chunks, we can improve results notably. Correcting the offset provides the best results, and together with Non-Linear optimization methods, we can achieve a median accuracy of 97 meters, with all results located within 220 m of the target. When varying the number of sensors (see Fig. 6b), we observe that with fewer sensors, there are fewer synchronization issues so the lower tail of the ECDF is closer to almost perfect estimation. However, with lower sensors, single sample differences result in higher errors.

Next, we compare the effect of upsampling the correlation outputs and present the results in Fig. 6c. We observe that in this scenario, upsampling does improve results significantly, but there is not much gain from larger upsampling factors, which could be explained by the limits in Carrier Frequency Offset (CFO) correction. In this case, we have a median accuracy of 50 m, and almost 90% of the results lie within 100 m. We compare this to a simulated scenario where sensors would have a commercial, yet expensive, GPSDO with a 50 ns timing alignment error. We observe that our approach does not lie far from that ideal case.

We also compare the effects of multipath correction with 3 and 6 sensors and present the results in Fig. 6e and 6f,



(a) FO correction (City 1) (b) Varying sensors (City 1) (c) Upsampling (City 1) (d) LLS/NLLS (City 2)

(e) Multipath 3 RX (City 1) (f) Multipath 6 RX (City 1) (g) Corr Methods (City 1) (h) Upsampling (City 2)

Fig. 6: Results in real environments. The first 3 columns refer to the results in City 1 and the last to the results in City 2. For the sensors in City 2, we remove the results when no correction is done because the errors are much higher and would distort the visualization. The chosen correlation method is `dphase` since it proves to be the most accurate overall.

respectively. With 3 sensors, the correction does have a more significant effect, and we can improve the whole system accuracy by several 100 m on the distribution tail. For 6 sensors, since not all sensors are affected by multipath/NLOS equally, the localization algorithm helps to reduce the positioning error as the number of sensors increases, those effects are canceled out, but still, the multipath correction does have an effect on the overall accuracy.

Last, we also compare the different correlation methods and, as in Fig. 4, we see in Fig. 6g that `abs` and `dphase` methods have better results than using raw `iq` as the correlation input. Furthermore, `dphase` performs slightly better than `abs` in these real settings, particularly above the 60th percentile.

2) City 2 deployment: Up to three sensors are set in different locations of the City 2 region, two with the latest RTL-SDR v3 and one with the previous version. This is noticeable because the latter has an average frequency offset of 60 PPM. In this deployment, the target and the reference transmitters are two DAB signals located at different positions, with the first one transmitting at 194 MHz and the second one at 208 MHz. 100 measurements are collected with these sensors over two months with the same conditions as in the City 1 deployment.

In Fig. 6d, we can see that, regarding the Non-Linear versus the Linear approaches, there is a significant difference between both methods. With Non-Linear methods, we can achieve a median accuracy of 50 m, 70% of the results within 65 m, and more than 90% within 150 m, which is 1 sample difference at the sampling rates of these sensors.

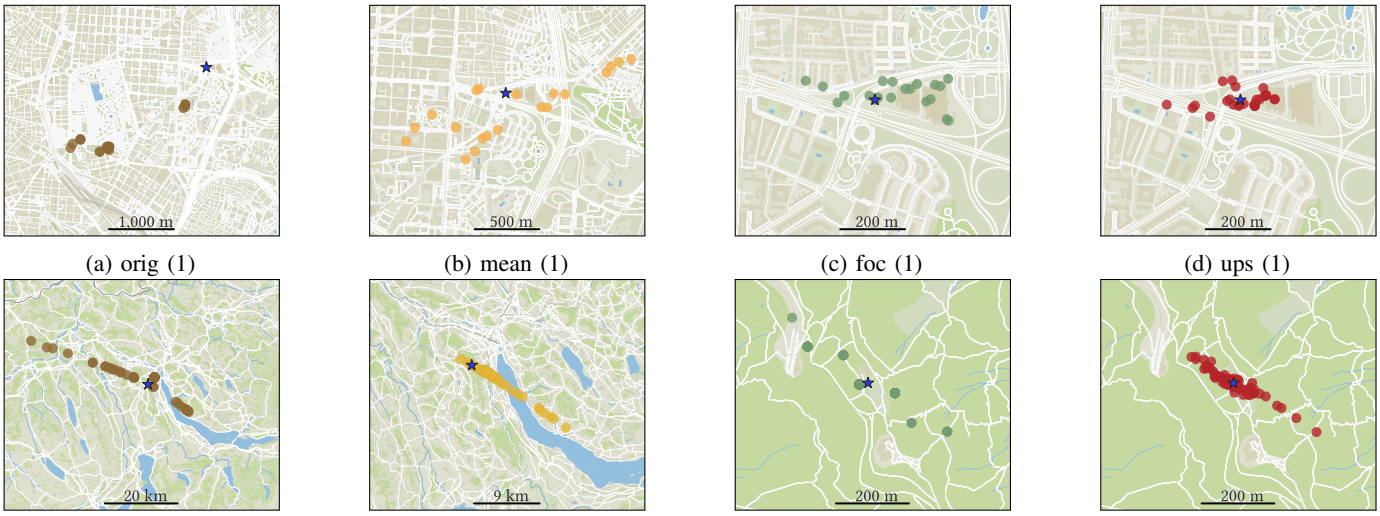
The effect of upsampling can also be seen in Fig. 6h, similar to the results in City 1. Upsampling does allow for more fine-grained positioning compared to no interpolation. In the latter, one sample difference can result in a greater error in the distribution. Last, we can observe the distribution of estimations around the transmitter (Fig. 7). The spatial arrangement matches what we would expect theoretically with such sensor disposition. For instance, City 2 has only three sensors, and the geometry causes errors mainly on one axis.

V. Related work

O-RAN provides a promising and flexible architecture for future cellular networks, and many applications might be possible thanks to it [9]. Localization, in particular, was proposed in [21] and even using passive sensors and signals of opportunity using DVB-T signals to monitor human activity. However, all localization approaches revolve around positioning users connected to the network. In this paper, we propose to extend this emerging architecture by means of passive sensors that can perform different tasks and focus on the positioning of several types of transmissions and modulations.

When tackling the problem of localizing unknown transmitters, we can distinguish two main categories: range-based and range-free methods. Range-based positioning is generally tackled from four perspectives as summarized in [22]. In the following paragraphs, we discuss some of the most relevant research with each of the aforementioned localization methods described. Many articles related to localization are based on theoretical conjectures and simulations, but rarely are these assumptions tested on scalable real deployments or areas greater than 100×100 m^2.

a) Received Signal Strength (RSS) based methods: In recent years, much effort has been focused on RSS based systems [23], [24]. The main appeal of RSS based methods is the low hardware requirements [25], which makes an attractive use case for deployments on large sensor networks. However, these techniques have drawbacks, such as the need for good calibration, complex path-loss models, and lower accuracy when distance increases.



(a) orig (1) (b) mean (1) (c) foc (1) (d) ups (1)

(e) orig (2) (f) mean (2) (g) foc (2) (h) ups (2)

Fig. 7: Positioning mapped for the cities of City 1 (1, first row) and City 2 (2, second row). We compare how the accuracy improves using no correction (orig), averaging reference signal (mean), frequency offset correction (foc), and correction+upsampling (ups). The transmitter position is marked with a blue star, and the dots represent the estimated locations.

b) AOA based methods: Angle of Arrival (AOA) requires complex receiver chains to compute the angle of incidence locally. It also has notable accuracy in theory and practice [26]. One of the main issues of AOA based methods is that to compute the angle of incidence, they require antenna arrays which might be challenging when deploying embedded sensors.

c) TOA & TDOA based methods: Time of Arrival (TOA) has been widely employed as a localization method in sensor networks since it provides accurate results and does not need to make assumptions on the signal [27]. TOA can be implemented using one-way transmissions or with echo techniques [28]. TDOA methods have slightly less accuracy with respect to TOA based localization approaches [27] but only need time synchronization between receivers. We have several examples in [29]–[33]. However, as for RSS based methods, the proposed systems are either focused on one specific technology, use high-end hardware or remain conceptual and do not pursue any larger scale implementation. [32] proposed a hybrid network of high- and low-end receivers, but it focused solely on aerial vehicle tracking and its protocols.

In our work, to minimize bandwidth usage, localization must be triggered only during a short period (e.g., after an algorithm has detected an anomaly in the spectrum). Another way to mitigate this is to compress this data by means of algorithms like the Fast Walsh-Hadamard Transform [34] or compressing sensing [35]. Sensors also need to be accurately geolocated. Work has been realized, for example, to opportunistically exploit time information in aerial signals as shown in [36], [37] without relying on GPS.

VI. Conclusion

ORAN-Sense is a novel architecture that leverages low-cost IoT spectrum sensors and the O-RAN to perform localization of non-cooperative transmitters in the next generation of cellular networks, enhancing their current capabilities. This study lays its foundations through design exploration. From the technical point of view, our system design considers aspects such as correcting sampling and frequency offsets with imperfect hardware and upsampling the signals. We also observed that signals with rich phase information performed better than those without. In our real-world experiments in two European cities with IoT sensors spanning areas of tens of kilometers, we show that it is possible to localize signals with median accuracies of tens of meters and 1.5 s of data, and these results replicate over the cities we have studied, even with little control over the position of IoT sensors, under large frequency offsets of their low-cost front-ends, as well as in the presence of multipath. Overall, ORAN-sense approach can be an effective solution to localize non-cooperative transmitters that could threat the scarce RF spectrum resources.

Acknowledgements

The research conducted by IMDEA Networks was sponsored in part by armasuisse under the Cyber and Information Research Program, in part by the project MAP-6G, reference TSI-063000-2021-63, granted by the Ministry of Economic Affairs and Digital Transformation, and the European Union-NextGenerationEU through the UNICO-5G R&D program of the Spanish Recovery, Transformation and Resilience Plan, and in part by the FPU19/03102 scholarship from the Spanish Ministry of Universities (MIU).

References

[1] L. Ceci, "Mobile internet usage worldwide-Statistics & Facts," https://www.statista.com/topics/779/mobile-internet/, 2023.

[2] "The Fall and Rise of Russian Electronic Warfare - IEEE Spectrum," https://spectrum.ieee.org/the-fall-and-rise-of-russian-electronic-warfare.

[3] S. Roy, K. Shin, A. Ashok, M. McHenry, G. Vigil, S. Kannam, and D. Aragon, "Cityscape: A metro-area spectrum observatory," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–9.

[4] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. V. D. Bergh, H. Cordobes, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and Big Spectrum Data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, 2018.

[5] F. Al-Turjman, E. Ever, and H. Zahmatkesh, "Small Cells in the Forthcoming 5G/IoT: Traffic Modelling and Deployment Overview," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 28–65, 2019.

[6] R. Calvo-Palomino, H. Cordobés, M. Engel, M. Fuchs, P. Jain, M. Liechti, S. Rajendran, M. Schäfer, B. V. den Bergh, S. Pollin, D. Giustiniano, and V. Lenders, "Electrosense+: Crowdsourcing radio spectrum decoding using IoT receivers," *Computer Networks*, vol. 174, p. 107231, Jun. 2020.

[7] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," *Computer Networks*, vol. 182, p. 107516, Dec. 2020.

[8] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, D. Maas, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L. B. Stoller, J. Van der Merwe, K. Webb, and G. Wong, "POWDER: Platform for Open Wireless Data-driven Experimental Research," in *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH'20. New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 17–24.

[9] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[10] "O-RAN ALLIANCE e.V," https://www.o-ran.org/.

[11] S. Bartoletti, L. Chiaraviglio, S. Fortes, T. E. Kennouche, G. Solmaz, G. Bernini, D. Giustiniano, J. Widmer, R. Barco, G. Siracusano, A. Conti, and N. B. Melazzi, "Location-Based Analytics in 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 38–43, Jul. 2021.

[12] 3rd Generation Partnership Project (3GPP), "Study on new radio access technology: Radio access architecture and interfaces," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.801, April 2017.

[13] O-RAN Alliance, "Wg1: Use cases and overall architecture workgrou," Specification, 2023, available at https://www.o-ran.org/specifications.

[14] ——, "Wg2: Non-real-time ran intelligent controller and a1 interface workgroup," Specification, 2023, available at https://www.o-ran.org/specifications.

[15] ——, "Wg10: Oam for o-ran," Specification, 2023, available at https://www.o-ran.org/specifications.

[16] ——, "Wg3: Near-real-time ric and e2 interface workgroup," Specification, 2023, available at https://www.o-ran.org/specifications.

[17] R. Calvo-Palomino, F. Ricciato, D. Giustiniano, and V. Lenders, "LTESS-track: A Precise and Fast Frequency Offset Estimation for low-cost SDR Platforms," in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 51–58.

[18] H. P. Gavin, "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems," *Department of Civil and Environmental Engineering, Duke University*, vol. 19, no. Department of Civil and Environmental Engineering, Duke University, 2019.

[19] A. Scalingi, D. Giustiniano, R. Calvo-Palomino, N. Apostolakis, and G. Bovet, "A Framework for Wireless Technology Classification using Crowdsensing Platforms," in *IEEE International Conference on Computer Communications*, May 2023.

[20] rtlsdrblog, "Rtlsdrblog/rtl-sdr-blog," Jul. 2023.

[21] A. Bourdoux, A. N. Barreto, B. van Liempd, C. de Lima, D. Dardari, D. Belot, E.-S. Lohan, G. Seco-Granados, H. Sarieddeen, H. Wymeersch, J. Suutala, J. Saloranta, M. Guillaud, M. Isomursu, M. Valkama, M. R. K. Aziz, R. Berkvens, T. Sanguanpuak, T. Svensson, and Y. Miao, "6G White Paper on Localization and Sensing," Jun. 2020.

[22] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[23] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2662–2686, 2016.

[24] H. Nurminen, M. Dashti, and R. Piché, "A Survey on Wireless Transmitter Localization Using Signal Strength Measurements," *Wireless Communications and Mobile Computing*, vol. 2017, p. e2569645, Feb. 2017.

[25] N. A. Azmi, S. Samsul, Y. Yamada, M. F. M. Yakub, M. I. M. Ismail, and R. A. Dziyauddin, "A Survey of Localization using RSSI and TDoA Techniques in Wireless Sensor Network: System Architecture," *2018 2nd International Conference on Telematics and Future Generation Networks, TAFGEN 2018*, pp. 131–136, 2018.

[26] J. Yin, Q. Wan, S. Yang, and K. C. Ho, "A simple and accurate TDOA-AOA localization method using two stations," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 144–148, 2016.

[27] R. Kaune, "Accuracy studies for TDOA and TOA localization," in *2012 15th International Conference on Information Fusion*. Singapore: IEEE, Jul. 2012, pp. 408–415.

[28] M. Rea, A. Fakhreddine, D. Giustiniano, and V. Lenders, "Filtering Noisy 802.11 Time-of-Flight Ranging Measurements From Commoditized WiFi Radios," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2514–2527, Aug. 2017.

[29] J. Wei and C. Yu, "Performance analysis of TDoA based localization using SDRs," in *2013 Australian Control Conference*. Fremantle, WA, Australia: IEEE, Nov. 2013, pp. 91–92.

[30] J. Schmitz, M. Hernandez, and R. Mathar, "Demonstration Abstract: Real-Time Indoor Localization with TDOA and Distributed Software Defined Radio," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Vienna, Austria: IEEE, Apr. 2016, pp. 1–2.

[31] F. Šture, T. Morong, P. Kovář, and P. Puričer, "High Performance SDR for Monitoring System for GNSS Jamming Localization," in *2019 International Conference on Wireless Networks and Mobile Communications (WINCOM)*. Fez, Morocco: IEEE, Oct. 2019, pp. 1–5.

[32] H. Sallouha, A. Chiumento, and S. Pollin, "Aerial Vehicles Tracking Using Noncoherent Crowdsourced Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 780–10 791, Oct. 2021.

[33] R. Schreiber and J. Bajer, "Time difference measurement algorithm for TDOA positioning system using RTL-SDR," in *2017 International Conference on Military Technologies (ICMT)*. Brno, Czech Republic: IEEE, May 2017, pp. 608–612.

[34] Y. Zeng, R. Calvo-Palomino, D. Giustiniano, G. Bovet, and S. Banerjee, "Adaptive Uplink Data Compression in Spectrum Crowdsensing Systems," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 1–15, 2023.

[35] J. Schmitz, R. Mathar, and D. Dorsch, "Compressed time difference of arrival based emitter localization," in *2015 3rd International Workshop on Compressed Sensing Theory and Its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*. Pisa: IEEE, Jun. 2015, pp. 263–267.

[36] M. Eichelberger, K. Luchsinger, S. Tanner, and R. Wattenhofer, "Indoor Localization with Aircraft Signals," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–14.

[37] A. Canals, P. Josephy, S. Tanner, and R. Wattenhofer, "Robust indoor localization with ADS-B," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 505–516.