

Collaborative Wideband Signal Decoding using Non-coherent Receivers

Roberto Calvo-Palomino
IMDEA Networks Institute, Spain
Universidad Carlos III, Madrid Spain
roberto.calvo@imdea.org

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

Fabio Ricciato
University of Ljubljana, Slovenia
fabio.ricciato@gmail.com

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

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

ABSTRACT

In recent years we are experiencing an important growth of interest for sensing the electromagnetic spectrum and making its access more agile. Emerging initiatives use low-cost receivers in large deployments for sensing the radio spectrum or collecting air-traffic signals at large scale. One of the major drawbacks of low-cost spectrum receivers is their limited sampling rate, which does not allow to *decode wideband signals*. In order to circumvent the hardware limitations of single receivers, we envision a scenario where non-coherent receivers sample the signal collaboratively to cover a larger bandwidth than the one of the single receiver and then, enable the signal reconstruction and decoding in the backend. We present a methodology to enable the signal reconstruction in the backend by multiplexing in frequency a certain number of non-coherent receivers in order to cover a signal bandwidth that would not otherwise be possible using a single receiver. We propose a method that does not use the knowledge of the modulation scheme, and has been designed to be transparent to the subsequent decoding process. As such, it is equivalent to the reception of the signal by a high-end receiver. We demonstrate and evaluate our approach with two non-coherent receivers which collaboratively sample an aviation signal of almost twice the bandwidth of each receiver. The experimental results show that, using two non-coherent receivers, our method is able to reconstruct and decode correctly more than 80% of data.

CCS CONCEPTS

- Computer systems organization → Embedded and cyber-physical systems; Embedded software;

KEYWORDS

Signal processing, Crowdsourcing, SDR, Mode-S, open source.

ACM Reference Format:

Roberto Calvo-Palomino, Héctor Cordobés, Fabio Ricciato, Domenico Giustiniano, and Vincent Lenders. 2019. Collaborative Wideband Signal Decoding using Non-coherent Receivers. In *The 18th International Conference on*

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

IPSN '19, April 16–18, 2019, Montreal, QC, Canada

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6284-9/19/04...\$15.00

<https://doi.org/10.1145/3302506.3310387>

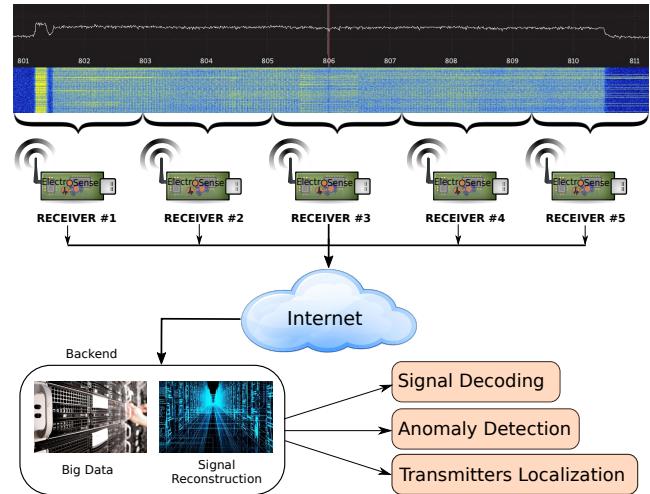


Figure 1: Several receivers distributed in a geographical area cover a bandwidth larger than the single spectrum sensor. By letting each spectrum sensor sample different adjacent portions of the wideband signal, we can reconstruct and decode a signal, which would not be possible using one single receiver.

Information Processing in Sensor Networks (co-located with CPS-IoT Week 2019) (IPSN '19), April 16–18, 2019, Montreal, QC, Canada. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3302506.3310387>

1 INTRODUCTION

We are in the age where crowdsourcing systems are helping the society to solve complex problems by collecting data, using low-cost devices massively deployed around the world and exploiting collaborative techniques among them. These systems can measure different indicators as pollution, temperature, solar radiation, and air-traffic signals. More recently, we are experiencing an important growth of interest for crowdsourced sensing of the electromagnetic spectrum and democratize the access to the community. Several projects such as Electrosense [1] rely on volunteers that host low-cost receivers to sense the electromagnetic spectrum at large scale. Having RF receivers deployed massively enables new strategies of spectrum analysis. Yet, as Analog-to-Digital Converter (ADC)

are costly, said receivers are affected by a low sampling rate. The consequence is that they do not fulfill the Nyquist-Shannon theorem for several types of wideband signals and are not able to decode them.

In particular, what has strongly gained attention in the research community is the ability to sense the spectrum using low-cost SDR (Software Defined Radio) receivers, such as the RTL-SDR [2]. These receivers are used in large deployments such as RadioHound [3], Electrosense [1] (for sensing radio spectrum) and OpenSky [4] (for collecting air-traffic signals). RTL-SDR receivers have an acceptable performance but obviously due to their low quality components they have serious limitations in terms of ADC resolution, frequency range or maximum signal bandwidth, and lack calibration. The maximum bandwidth of low-cost RTL-SDR usually is 2.4 MHz [2], meaning they cannot decode signals such as LTE (10 MHz), WiFi (20 MHz), WiMAX (20 MHz) or air traffic signals such as Mode-S uplink (4 MHz) [5].

In order to solve this problem, we envision a scenario where low-cost RF receivers deployed distributively in an area work collaboratively to cover a bandwidth larger than the one of a single receiver. As a crowdsourced approach, we use such a distributed sensor arrangement to maximize the signal coverage of the system, even if some sensors may suffer reception issues in the band of interest (higher-capability receivers are less suited due to the increased cost of geographical scalability). A pictorial representation is given in Fig. 1. Several RF receivers located in different places are multiplexed in frequency, and take care of a specific portion of the signal. The spectrum data obtained by the sensors is sent to the backend where the signal is reconstructed. This scenario presents important challenges among non-coherent receivers that must be solved in order to guarantee the signal integrity and the reconstruction in the backend using the partial view of the signal seen by each sensor. The signal reconstruction in the backend is performed in an *agnostic* way, i.e., the system is not aware of the specific modulation scheme. As such, the system must deal with the fundamental signal information (amplitude, frequency, phase...) provided by each sensor to make the signal recombination feasible in the backend.

In this paper we aim to provide a generic methodology and architecture to enable the signal reconstruction in the backend by multiplexing in frequency a certain number of receivers in order to cover a signal bandwidth that would not otherwise be received using a single sensor. Our contributions are as follows:

- We propose a distributed frequency multiplexing mechanism for covering bandwidths higher than what a single receiver can.
- We identify the main challenges in signal processing to solve in order to enable the collaborative signal reconstruction.
- We present an architecture for collaborative signal reconstruction performed in a common backend that does not use information from the modulation scheme.
- We evaluate our solution in a real scenario using low-cost receivers for reconstructing and decoding collaboratively air traffic signals (Mode-S [6]) in the backend and we compare our solution against high-performance receivers.

2 MOTIVATION

The collaborative and distributed **Signal Decoding** using non-coherent and low-cost receivers is the main scope of this work. Decoding signals of larger bandwidth than a single receiver in a distributed network is the main application that is presented in this work. These sensors are independently deployed by users in a given area and connected to the backend over the internet. Yet, the ability of decoding signals larger than what a single spectrum sensor can do is a *primitive* that can enable other key applications, a foundation that exploits the crowdsourcing and the distribution of costs among all participants to now be able to do more with a sensor, and can provide incentives for people in the same neighborhood to join the initiative. The applications below are presented for completeness.

- **Anomaly detection.** Anomaly detection can be more powerful if part of the data can be labelled (e.g. signal-modulation classification [7]). Yet, if the signal is of wider bandwidth than the receiver, the backend cannot decode the data and label it properly, limiting the applicability of methods already proposed [8] to detect anomalies in the spectrum.

- **Localization.** For signals of larger bandwidth than the receiver, the ability to decode the signal can be considered as a sanity check that the different receivers are synchronized. By exploiting the position of known transmitters, the receivers can then solve the delay from the transmitter to each receiver and from the receiver to the backend. Also, the fact that a subset of sensors yields the best collaborative decoding rate of a known signal may be used as an indicator that this group of sensors could be the optimum to localize the transmitter.

- **Selection of sensors.** Usually crowdsourced and collaborative systems face the problem of choosing the most favourable set of sensors to achieve a specific goal as in spectrum monitoring, transmitter localization or detection of anomalies in the spectrum. Choosing the closest sensors not always ensures the best performance, due to interference or antenna location. Receivers can be instead better grouped depending on how good they perform in the process of decoding collaboratively a known signal. The problem of choosing the best subset of sensors for this task is not trivial. If one of those sensors reports bad information it may taint the final goal. Besides the great value and contribution of collaboratively decoding a signal using different low-cost sensors, this method can help in ranking and grouping sensors in crowdsourced spectrum monitoring networks.

- **Enhanced diversity.** By having receivers located away from each other we can implement a diversity schema to increase the signal-to-noise ratio. The methods exposed in this work can be used to align signals, so they can be further processed in equal-ratio, maximal-ratio or selection combination.

3 DECODING WIDE SIGNALS WITH DISTRIBUTED SENSORS

The objective of this work is to use different low-cost receivers to cover a high bandwidth that would not otherwise be possible using one single receiver given its limited ADC sampling rate. The first problem to address is what type of receiver to use, and two options

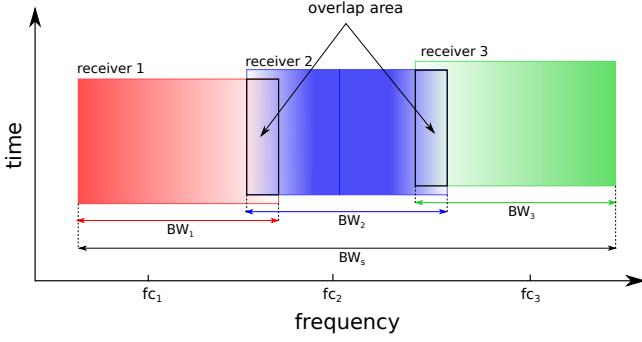


Figure 2: High-level representation of the frequency division approach to deal with wideband signals and low-cost receivers. The proposed frequency division approach makes use of overlap areas in frequency between different independent spectrum sensors that are tuned at different, yet adjacent, frequencies. Collected samples in the overlap areas are used to synchronize and equalize the signals in the backend.

are possible. On the one hand, coherent-receivers [9] know the type of signal that has been transmitted, and typically use calibration and synchronization techniques in order to synchronize in time and phase and compensate for impairments in the wireless channel. On the other hand, non-coherent receivers do not have instead any knowledge of the signal properties. Spectrum sensors have the capabilities of collecting IQ samples at different frequencies (in the case of the RTL-SDR, with low ADC sampling rate, the range is between 24 and 1766 MHz [10]), and they do not need to interact with the transmitters (no strict time requirements to process the incoming IQ data). Therefore we propose to use non-coherent receivers to *divide the wideband signals in different sub-channels*, each of them monitored by one single receiver, and then reconstruct *said wideband signals* in the backend. In other terms, we apply a *frequency division approach* to multiplex the signal. The advantage of non-coherent receivers is that they do not need to be close to each other and share internal components (e.g., local oscillators), but the disadvantage is that the signals provided by non-coherent receivers are not synchronized.

3.1 Common signals in the frequency domain

In typical non-dense frequency-division multiplexing, the spectrum of each channel is independent. Instead, in order to solve the issue of synchronization among non-coherent receivers, we propose that the sub-channels have a small overlap area. We use this common area, named as *overlap area*, to extract the estimators needed to synchronize the signals. As Fig 2 shows, receivers are located in nearby frequencies in such a way there is a overlap frequency area among them. This overlap area (O) contains common information that will be used to reconstruct the signal as it is explained later in Section 4. Notice that the overlap area size does not depend on the final aggregated bandwidth. The overlap area size can be set depending on the use case or the scenario conditions, e.g. if the noise in the signal is too high the overlap area bandwidth can be increased until enough data for a clean merge is available.

More formally, a given signal $s(t)$ with a specific bandwidth BW_s and center frequency f_{cs} is to be reconstructed and decoded by using N receivers, configured with certain bandwidths (BW_1, BW_2, \dots, BW_n) and center frequencies ($f_{c1}, f_{c2}, \dots, f_{cn}$) in such a way that the complete bandwidth of the signal $s(t)$ is covered. The signal reconstruction using the frequency division approach can be exploited as long as the sum of the bandwidths covered by the receivers is strictly greater than the bandwidth of the original signal as Eq. 1 shows:

$$\sum_{k=1}^N BW_k > BW_s \quad (1)$$

Notice that in the above equation the terms BW_k already include the overlap area bandwidth (c.f. Fig. 2). Therefore, the overlap area between two consecutive receivers, which is the key of the signal reconstruction, given that the frequencies $f_{c(n)}$ are in ascending order, can be expressed as follows:

$$O_{(n,n-1)} = \left(f_{c(n-1)} + \frac{BW_{n-1}}{2} \right) - \left(f_{c(n)} - \frac{BW_n}{2} \right) \quad (2)$$

3.2 Challenges

The usage of distributed non-coherent receivers implies that we need to process the signals in such a way that we can cope with the signal impairments with respect to a pure ideal scenario. The following challenges must be solved in order to reconstruct the signal collaboratively by applying the proposed approach that exploits overlap areas.

Relative time delays between signals. As each signal traverses a different path (different location, antenna, receiver sampling time...), most likely a delay from one signal to the other will be found. The main root causes are: i) *Different time-of-arrival in multi-antennas scenarios*: the electromagnetic signal may reach each antenna at different times, mainly depending on the different path observed by different antennas. On a more generic set-up, the collaborative reception may be performed from distant receivers, so there may appear a time delay between the different received signals because of this distance; ii) *Different receiver architectures*: if using different receivers, the delay from the RF interface to the final acquisition point may differ. Also, as the sampling clock is a synthesized signal, the sampling instant can be additionally delayed from one signal to the other on a sub-sample quantity.

Frequency alignment. The internal clocks used in radio circuitry are affected by tolerances and environmental factors, so we need to handle these effects when combining signals. We have to bear in mind that this is an agnostic reconstruction without any knowledge of the signal properties, so the overall reconstructed signal does not guarantee a specific center frequency better than the clock tolerances. This is not a problem, as in typical low-cost SDR there is no PLL (Phase-Locked-Loop) synchronization to the carrier [11], and demodulators cope with this effect. In fact for contributions from receivers which are not receiving a carrier, performing phase-lock could be unfeasible. The main source of misalignment is the *offset in the local oscillators*. As a simplification on the signal flow, let us consider without loss of generality that on each receiver there is only one local oscillator to down-convert the radio transmission

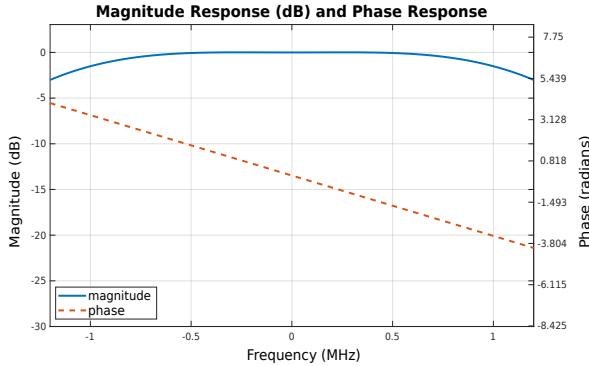


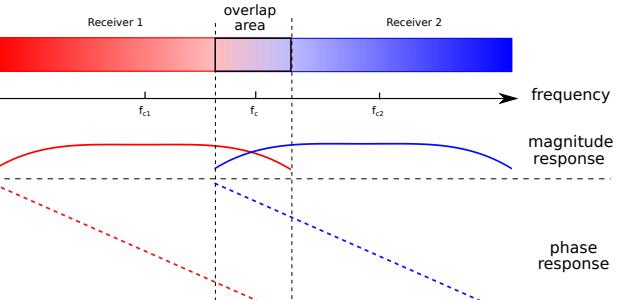
Figure 3: Low-pass filter used for antialiasing and wideband signal distortion in the overlap area (representation in the frequency domain). FIR filter of the RTL-SDR v3 (left) and how the filter is applied in two different receivers in the overlap area (right). As the overlap area is used to synchronize and equalize signals from different spectrum sensors, artifacts such as the internal low-pass filters of each spectrum sensor should be compensated.

into an complex base-band signal. This oscillator may be affected by an offset in its reference clock (such as a quartz crystal, where tuning during manufacture and operating temperature may affect its nominal frequency). Additionally this type of oscillators is generated from a chain of integer multipliers and dividers of the clock frequency, so an additional granularity error may appear as well.

Signal power at the receiver. There could be many different factors that could impact the power of the sampled signal among different receivers (radio channel differences, antenna gain, amplifier configurations, etc.), so it will be needed to equalize the signal power from each spectrum sensor so that the reconstruction does not distort the original signal.

Receiver frequency response equalization. The receivers apply an antialiasing filter prior to sampling the signal. As an example, Fig. 3 shows the final magnitude and phase response of the FIR (Finite Impulse Response) filter applied in the RTL-SDR-v3 [2] receiver in the frequency range of interest (using the maximum sampling rate available). In the overlapping ranges (Fig. 3) we aim to compare and fuse contributions from different receivers, so it will be necessary to invert the filter response to better reconstruct the signal. In case the receiver implements a FIR filter the phase response can be considered just a delay in the time domain. If not, a complementary phase correction at this stage will make sure that the final phase response is linear with the frequency.

Phase matching. Once the receiver filter response in the receiver has been equalized, and the signals received in the overlap areas from the non-coherent receivers have been corrected in time and frequency, the phases of the signals in this overlap areas have to be matched. The phase difference may be caused by different reasons. As the final synthesized signal from the local oscillators is generated by a series of multipliers and dividers, without a PLL on the received signal, we cannot guarantee the value of the local oscillator phase, even in shared-local-clock scenarios. So we will not have a *phase synchronization of the local oscillators*. This will be seen as a relative phase rotation between the signals (a constant phase difference in the same frequency components from the signals). We may also



find a *subsample-delay phase change*, as we cannot control the exact sampling instant among all the signals (even with synchronized clocks). Because of that, we may still find some additional subsample delay that can be interpreted as a linear phase change. Also some *different phase contribution during combination* may appear. If there is still some linear-phase response in the chain (some delays unaccounted for, other filters, etc.) that has not been completely corrected, as the final phase integration of the contributions reflect different bands, we may find an additional separation of the phases. To understand this difference, we can see the distance between the phase responses (red and blue straight lines) in the overlap area in Fig. 3 (right). This height is proportional to the separation of the tuning frequencies (f_{c1}, f_{c2}).

4 COLLABORATIVE SIGNAL RECONSTRUCTION

In this section we present our methodology to collaboratively reconstruct a signal in the backend using partial representations captured by different receivers.

The proposed methodology only assumes the following signal properties to achieve a correct collaborative signal reconstruction in the backend: (1) The Signal-to-Noise Ratio (SNR) must be good enough in order to detect a potential packet on the overlap area. (2) The signal must allow a clear cross-correlation maximum computation in the overlap area to estimate the coarse time synchronization. (3) The signal must occupy a continuous portion of the spectrum with no gaps bigger than the overlap area, allowing the merge using that portion of the signal. (4) The channel response in the overlap area must be similar for both receivers.

The overview of the signal reconstruction methodology is shown in Fig. 4. In the previous section, we discuss the case of two signals, as the case of more signals can be regarded as an inductive extension (adding any additional received signal to the previously reconstructed one). Our solution does not make any assumption on the signal besides the frequency range and maximum expected length of the packets. The architecture consists of four different

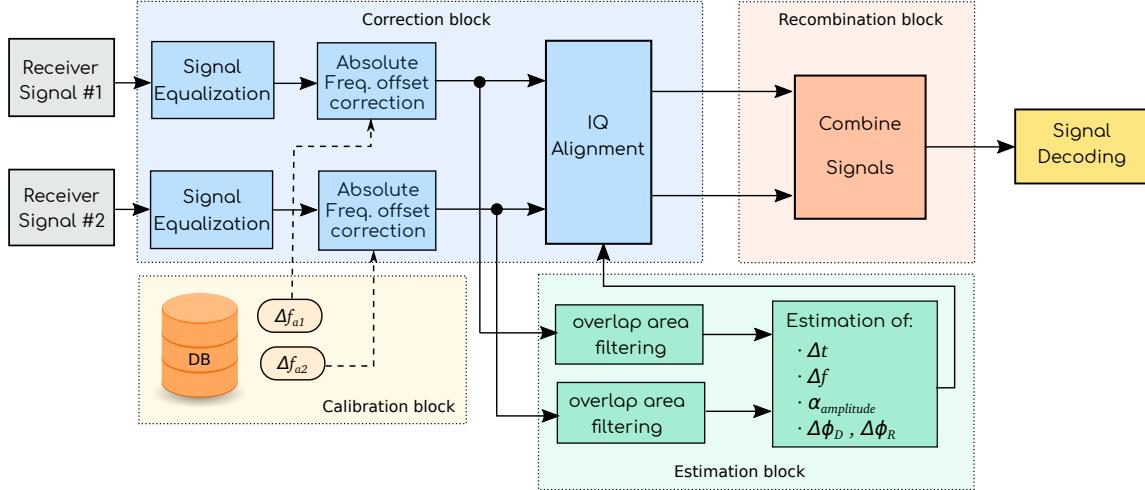


Figure 4: Signal reconstruction workflow.

blocks: Calibration, Correction, Estimation and Recombination. The *calibration block* computes the intrinsic parameters of the receivers such as the absolute frequency offset (Δf_a). The *correction block* is responsible for modifying the signals according to the parameters provided by the estimation block. The *estimation block* computes the parameters to synchronize both signals as time offset (Δt), relative frequency offset (Δf), amplitude ratio (α) and phase offset ($\Delta\phi$). All these parameters are computed using only the overlap area, where the received signals share a common frequency range. Lastly, the *recombination block* takes two signals already synchronized and fuses them to generate a signal with the specific bandwidth required.

For the simplest model of two different receivers (more receivers could be considered sequentially once the first two portions have been integrated), let us consider the signal received on the overlapping area. From a complex signal transmitted originally as

$$x(t) = \Re\{s(t)e^{j2\pi f t}\} \quad (3)$$

the received signals $x_r(t)$ can be modeled like this to account for time, amplitude and phase differences in the receivers, assuming that the frequency deviation Δf is actually an error induced by the local oscillator, and the channel transfer function is flat.

$$x_1(t) = \Re\{s(t)e^{j2\pi f t + \phi_1}\} + \sigma_1(t) \quad (4)$$

$$x_2(t + \Delta t) = \alpha[\Re\{s(t)e^{j2\pi(f + \Delta f)t + \phi_2}\}] + \sigma_2(t) \quad (5)$$

We will define $s_r[n]$ as the complex baseband, discrete-time signal provided by the receiver r , which have been obtained by tuning the receivers to the frequencies $f_{c,r}$ and a sampling rate f_s,r in accordance with the receiving bandwidth BW_r .

As the original signal $s(t)$ is not available, our goal is to find the parameters so we can correct the *differences* between both signals in the overlapping area, and subsequently apply these parameters to correct the whole signals $s_r[n]$.

4.1 Calibration Block

There are some intrinsic parameters in the receivers that do not change substantially over time. One of these parameters is the frequency offset on the receiver caused by the internal oscillator. Nowadays, most receivers integrate a temperature compensated crystal oscillator (TCXO) to mitigate the fluctuations of the internal oscillator due to temperature changes.

This block is responsible for estimating, in a calibration process, the absolute frequency offset (Δf_a) of the receiver using a known and precise signal. We rely on LTESS-Track [12] in order to estimate the frequency error of the receiver. LTESS-Track takes advantage on the synchronization signals transmitted by LTE base stations as reference in order to provide a frequency offset estimation with sub-ppm (parts-per-million) accuracy. The absolute frequency offset (Δf_a) for each receiver will be used to compensate the frequency error between the receiver and the nominal frequency of the transmitter, providing the *coarse frequency synchronization* of the system. Notice that this calibration process is run only once per receiver and therefore the Δf_a value can be used as long as the receiver is not replaced.

Since Δf_a may vary slightly due to the tolerance error of the local oscillator, the variance on the calibration, environment factors [12], and the intrinsic uncertainty of real oscillators, a *fine frequency synchronization* is needed as is explained in Sec 4.3. The calibration process is needed since two given receivers may have a significant frequency error (for instance 100ppm). As the reconstruction process matches the frequency components between the sensors, but not to the original f_c , the final reconstructed signal will show the same frequency error shift.

4.2 Correction Block

The correction block receives IQ data directly from the receivers. The Analog-to-Digital Converters make use of an antialiasing[13] filter prior to obtaining the samples that will be the discrete representation of the signal. As an example, the magnitude and phase

response of the FIR filter applied in the RTL-SDR v3 is shown in Fig. 3. The first step is to equalize the signal to correct the possible changes made by this filter, in such a way that the rest of the reconstruction steps can use a signal without distortions (or reasonably minimized). It comes without saying that for this process to work the filter response must be known for all the different devices involved in the reconstruction task. Recalling that to perform the signal recombination we will use the overlap frequencies between the two receivers, we see the band of the common area in every receiver has been modified by a different band of the filter, as Fig. 3 shows. This difference makes it clear why this effect must be handled and minimized.

The next action is to correct both signals in frequency given the $\Delta f_{a,n}$ provided by the calibration block. The goal is to obtain a signal as close to the raw ideal as possible so to perform a precise synchronization in the *estimation* block.

The *IQ alignment* block then performs the operations needed over one of the signals using the estimated corrections provided by the *estimation* block that is disclosed in the next subsection.

4.3 Estimation Block

The *estimation* block receives the signals already equalized as input. The discrete-time signals present in the overlap areas (o_1 and o_2) are extracted from the received signals (s_1 and s_2) by extracting the higher and lower frequencies of the signals respectively as Fig. 4 shows. The overlap area represents the common frequencies in both receivers where all the synchronization operations will take place. The width of the area is directly related to the receiver bandwidth (BW_r) and the tuning frequencies of the receivers (f_c, r).

First, the **coarse time synchronization** (Δt) is computed by cross-correlating the overlap areas of both signals in the time domain using *chunks* of data of certain duration as Eq. 6 shows. As at this point we cannot guarantee that the I and Q components from both signals are matched (as there might be a phase rotation between them, ϕ_n values are unknown), this operation is calculated over the norms. Considering Δt in number of samples:

$$\hat{\Delta t} = \underset{\tau}{\operatorname{argmax}}(|o_1| \star |o_2|)[\tau] \quad (6)$$

We assume the signals s_r have a Signal-to-Noise Ratio (SNR) high enough to detect the presence of a transmission. Other characteristics from $s(t)$, such as the modulation scheme, are not needed to estimate the corrections. For every *chunk* of data already synchronized in time an energy detector is applied in order to get the interesting time ranges of the signal called *bursts*. Then, a *burst* is defined as a time range of the signal, considered in the overlap area, where a potential packet (P) can be and therefore where the synchronization parameters will be estimated. A couple of bursts (b_1 and b_2) are shown in Fig. 5-a). At this point is where the synchronization parameters needed for this potential packet (P) can be estimated, in order to combine and match the signals.

Secondly, the **power normalization** of the bursts is done by computing the scale factor α . The original energy in the overlapping area is nominally the same, so we can estimate the α parameter as the square root of the power ratio of the two signals:

$$\hat{\alpha} = \sqrt{\frac{E[|b_2|^2]}{E[|b_1|^2]}} \quad (7)$$

Third, the **relative frequency shift** (Δf) is estimated providing the *fine frequency synchronization* in the system. To compute the frequency misalignment between both signals we may use the discrete Fourier transform of the signals. Given that the signals have not been distorted we may use their frequency representations to find the best match. As the values for ϕ_r are still unknown, once again it is needed to use the norm of the functions to calculate cross-correlation. Given that $b_r[n] \xrightarrow{DFT} B_r[k]$, and by cross-correlating the overlap areas in the frequency domain:

$$\hat{\Delta k_r} = \underset{\theta}{\operatorname{argmax}}(|B_1| \star |B_2|)[\theta] \quad (8)$$

It is worth noting that the bursts have been extracted so to contain some guard time before and after the packet P . This helps to reduce spurious components in the DFT, as it behaves as a windowing in time. The relation with the real frequency is then, given the length of the DFT as M :

$$\hat{\Delta f_r} = \frac{f_s \hat{\Delta k_r}}{M} \quad (9)$$

From this point forward, let us define $B'_2[n]$ as the frequency-corrected equivalent of $B_2[n]$.

Fourth, we perform the **phase difference estimation**. In order to have a coherent recombination process of the signal, the phases of both bursts must be matched. This step is important since phase alignment represents both a matching of I and Q components from each $s_r[n]$ (neutralizing the relative phase rotation between them), and a fine (subsample) time synchronization. Fig. 5-b) represents the phase of the bursts in the frequency domain. Notice that we only take those bins whose amplitude values are above a certain threshold (A_{th}) since low amplitude values usually introduce noise in the phase component that can lead to an erroneous phase estimation.

After that, we can study the phase of the bursts visually, by unwrapping the phases to get a continuous representation of the phase over the frequency bins as Fig. 5-c) shows. It is clear that phases are not equal and therefore we compute the phase difference over all the frequency components of both bursts and then we obtain the phase difference shown in Fig. 5-d). This difference should take care of the local phase conditions ϕ_n , possible fine time adjustments and phase rotations. As we saw in section 3.2 the expected difference is a linear function of the frequency. Thus we will first calculate the phase difference:

$$\Delta\phi[k] = \angle B_1[k] - \angle B'_2[k] \quad (10)$$

And given that the k elements are ordered such to keep the DC component in the center, then the linear regression over $\Delta\phi$ could be expressed as:

$$\hat{\Delta\phi}[k] = \hat{\Delta\phi}_R + k \hat{\Delta\phi}_D \quad (11)$$

where $\hat{\Delta\phi}_R$ can be regarded as a rotation that applies to all the frequency components, and the slope $\hat{\Delta\phi}_D$ as a time delay, with a subsample resolution.

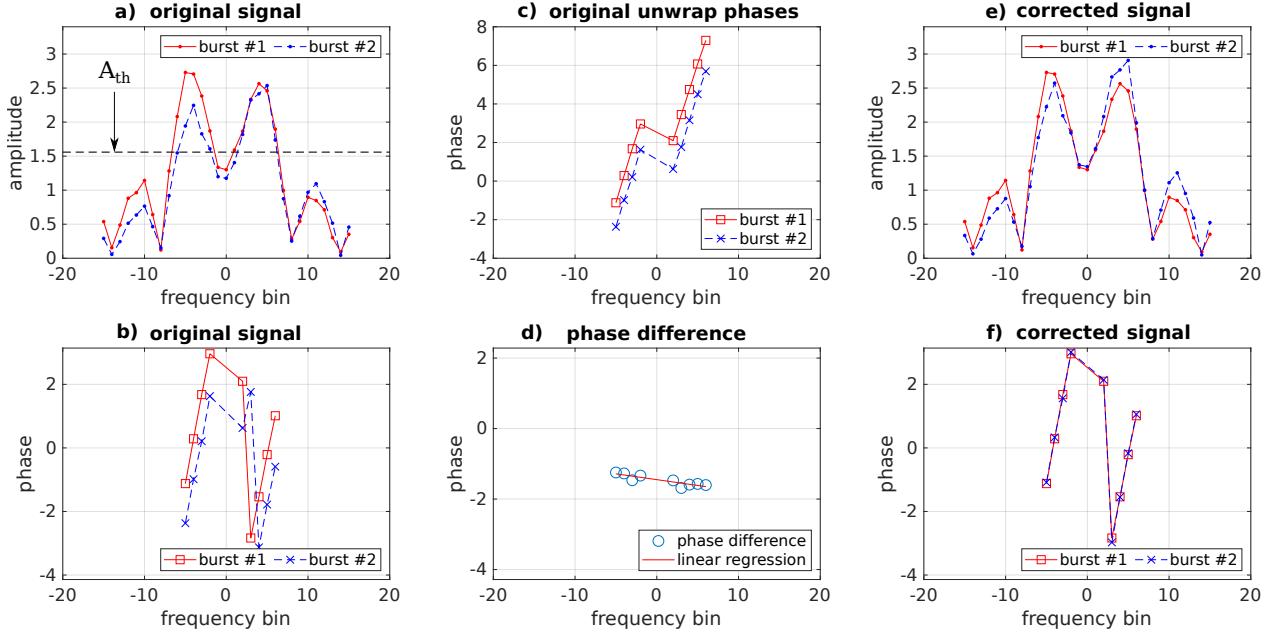


Figure 5: Amplitude and phase representation of the original bursts (left), phase difference estimation (center) and bursts already aligned (right).

Fig. 5-d) shows the value of the burst phase difference and its linear regression. Fig. 5-e) and Fig. 5-f) show the bursts normalized in amplitude and synchronized in phase after applying the correction values, $\hat{\alpha}$ and $[\Delta\hat{\phi}_D, \Delta\hat{\phi}_R]$ respectively.

We note that the estimation block does not have the notion of *packet*. In particular, this block performs a signal-level synchronization only using the bursts (portion of signal in the overlap area). The packet-level synchronization and its detection are performed by the decoder ("Signal decoding" block in Fig. 4), which is independent from the signal reconstruction scheme.

4.4 Recombination Block

The *recombination* block takes as input the signals $s_r[n]$ of bandwidth BW_r , which were tuned to in $f_{c,r}$. The first step is to resample both signals to increase the sampling rate. The minimum upsampling is that which obtains a sampling frequency enough to contain the full reconstructed signal. After that, both signals are relocated so their components of f_c match the equivalent f_c of $s(t)$, by shifting $f_c - f_{c,1}$ and $f_c - f_{c,2}$ respectively. The frequency components on the overlapping area are halved, as that area will receive the energy from the two signals. After that, both signals are summed and downsampled if needed to match the specific bandwidth desired and therefore the reconstructed signal is complete as Fig. 4 shows.

4.5 Decoding

The last step of the process is to decode the signal that has been reconstructed. Decoding the signal provides a good indicator that the reconstruction is done properly in the backend. It is worth

mentioning that the *signal decoding* block (Fig. 4) is not aware about the origin of the signal, e.g., the *decoder* does not know if the signal either has been reconstructed by our method or has been collected by a regular receiver.

5 EVALUATION

This section describes how we evaluate our methodology for signal reconstruction in real scenarios. We use *real signals* to evaluate our collaborative signal decoding, specifically we use Mode-S uplink [5], [6]. These signals are sent by aircraft and uses 1030 MHz frequency for transmissions and 4 MHz of signal bandwidth. The receivers used for the signal reconstruction are the well-known RTL-SDRv3, whose maximum available signal bandwidth is 2.4 MHz. Therefore this is a very complete set-up for testing and evaluating our collaborative signal reconstruction methodology in a real-world and complex scenario (amplitude-phase modulation, signal sent from moving transmitters, different SNRs, etc.). In the following subsections, Mode-S and the experimental setup are explained and after that the most important evaluation results are shown.

5.1 Real use case: Mode-S uplink

Aircraft are constantly sending air traffic signals that contain useful information for the safety and optimization of the flight routes. Several initiatives as OpenSky [4], Flightradar [14] or Flightaware [15] have already deployed low-cost sensors based on RTL-SDR devices for collecting avionic signals whose signal bandwidth is smaller than 2.4 MHz, e.g., Mode-S downlink channel. All those platforms could get benefit of our contribution of using non-coherent receivers for decoding signals with a higher bandwidth than 2.4 MHz.

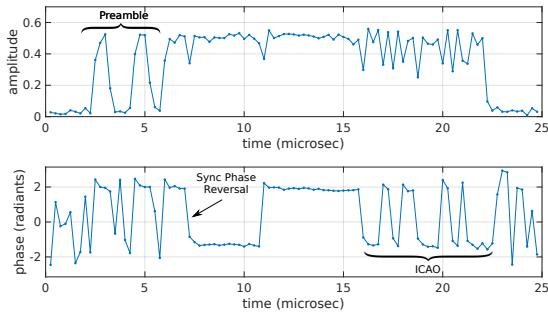


Figure 6: Amplitude representation (top) and phase representation (bottom) of a Mode-S uplink packet captured at 1030 MHz @ 4 MHz by an USRP.

Mode-S [5][6] is a *Secondary Surveillance Radar* (SSR) process that allows selective interrogation of aircraft according to the unique address assigned to each aircraft named *ICAO*. The ICAO is a 24-bit field used to identify every aircraft uniquely worldwide. Mode-S uses two different channels, one for interrogating (uplink channel, 1030 MHz / 4 MHz) and another for answering (downlink channel, 1090 MHz / 2.4 MHz). The aircraft replies to interrogations initiated from either a ground station or other aircraft. The aircraft interrogation is performed using the uplink channel and has the capability for selective interrogation of individual Mode-S transponders. This functionality allows to interrogate a specific aircraft as needed, avoiding the response from nearby aircraft and therefore an unnecessary saturation of the channel. The messages sent by the uplink channel are the ones that cannot be decoded currently by one single RTL-SDR due to the signal bandwidth limitation.

Mode-S messages can be 56 or 112 bits long. As Fig. 6 shows, Mode-S uplink messages consist of a preamble part (used for message detection), a Sync Phase Reversal (SPR) (used for time synchronization), and the payload encoded using *differential phase shift keying* (DPSK) where the ICAO of the addressed aircraft is indicated. Other important piece of information encoded in the payload is the uplink format, reply length, acquisition special or designator id [6][16]. It is worth mentioning that our generic solution proposed in Section 4 and the evaluation performed in Section 5.3 are not aware about the signal specifications explained here. Our solution is able to reconstruct the signal from two different channels without any knowledge about Mode-S uplink signals.

5.2 Mode-S Decoder

We have implemented the first open source decoder for Mode-S uplink using an initial and experimental implementation provided by [17]. The decoder has been implemented following the standard described in [6][16]. Our decoder implements the following features:

- Preamble detection.
- Size packet detection: supports lengths of 56 or 112 bits.
- Sync Phase Reversal (SPR) detection which is located at 4.75 μ s from the beginning of the packet.
- Sliding window implemented to analyze data continuously.

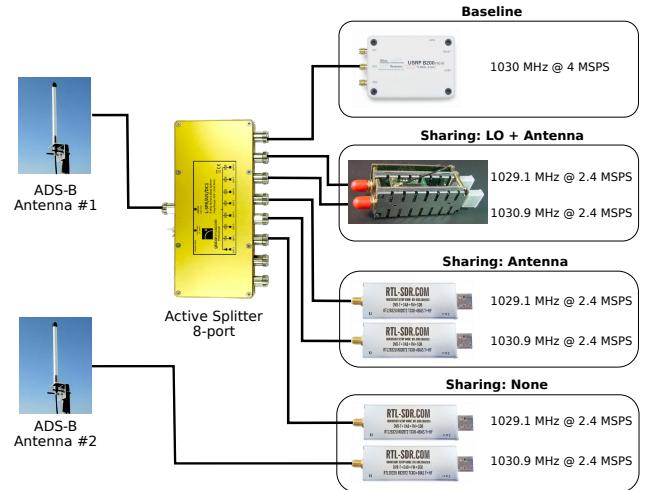


Figure 7: Testbed.

- Support input signals of 4 MHz and 8 MHz bandwidth.
- DPSK demodulator for payload extraction.
- ICAO of aircraft: The last 24-bits of the payload are the parity bits xored with the ICAO address of the aircraft. The CRC is computed as specified in [16] and aircraft's address is retrieved.
- Symbol timing recovery implementation since transmitter and receiver are not synchronized.

Our decoder is the first open source implementation¹ and is able to decode properly the 92% of the Mode-S packets detected.

5.3 Testbed

The experimental set-up consists of 3 pairs of sensors with two RTL-SDRv3 "Silver" model attached to a Raspberry Pi-3 with different ADS-B antennas and internal oscillator configurations. The maximum bandwidth supported by the RTL-SDRv3 is 2.4 MHz, therefore every pair of sensors has the mission of reconstruct the Mode-S uplink packets sent at 1030 MHz with 4 MHz of signal bandwidth. In order to do that, both receivers have to tune at 1029.1 MHz and 1030.9 MHz respectively in such a way the overlap area is 0.6 MHz and the total bandwidth covered is 4.2 MHz. Fig. 7 shows the different configurations tested on every pair of receivers depending on the resources shared:

- Local oscillator and antenna shared ("local oscillator + antenna"): This is the most favourable set-up where we use a clock-shared receiver based on two RTL-SDR² that shares the local oscillator (TCXO @ 28.8 MHz). They also share the same ADS-B antenna thought the 8-port active splitter.
- Antenna shared ("Antenna"): This configuration uses two vanilla RTL-SDRv3 receivers without sharing the internal clock and whose only shared resource is the ADS-B antenna thought the splitter.

¹<http://github.com/openskynetwork/modes-uplink-decoder>

²<https://coherent-receiver.com/products/coherent-receivers/>

1-supervisory-and-4-coherent-channels

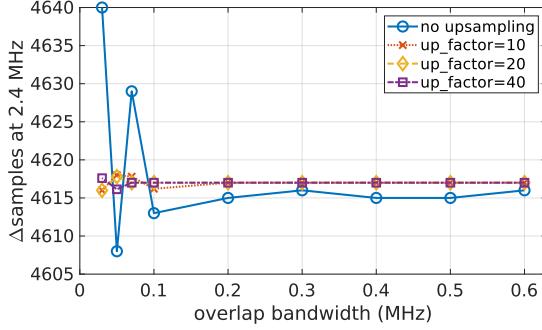


Figure 8: Sample synchronization vs. overlap area.

- Non-shared resources ("None"): This configuration consists of two vanilla RTL-SDRv3 receivers without sharing the internal clock and attached to different antennas that are 1 meter away.

Our results are based on multiple traces collected in Madrid (Spain) where more than 30,000 Mode-S uplink messages were recorded. In order to compare the performance in every configuration we use a baseline simultaneously created by a high-end receiver, USRP Ettus B210³ that is set at 1030 MHz and covering the whole signal bandwidth (4 MHz). The USRP is also connected to the splitter to share the same ADS-B antenna that the RTL-SDR receivers use. Therefore, the baseline is defined as the number of Mode-S uplink packets correctly decoded with the USRP samples. Since we do not know whether the ICAO encoded in the packet belongs to a real aircraft, we cross-verify that ICAO was also detected by the OpenSky [4] network in the Mode-S downlink channel at the time and place the trace is collected.

5.4 Evaluation metrics

In this section we evaluate the most important parameters of the system and how they can play an important role for the final signal reconstruction and decoding.

As we described in Section 4, one of the first synchronization steps is the time alignment between the signals. This step must be successfully performed to estimate the amplitude, frequency and phase offset accurately. All the synchronization steps over the signal are performed on the overlapping frequency area (which is common among the receivers). Therefore the width of the overlap area becomes a significant parameter of the system to be evaluated. The overlap area width optimization will also optimize the effective bandwidth available for every receiver, thus covering wider bandwidths with the same number of receivers. Fig. 8 shows the estimated time difference between the two signals at 2.4 MHz for different widths of the overlapping area. Note that to illustrate with comparable magnitudes, the y-axis units are samples of the original signal. The smaller the overlap area is, the less signal data we have to perform the time synchronization, and as expected by the time-frequency relation, the coarser time resolution we obtain.

To mitigate this effect, and obtain a better time representation of the signal edges, we can upsample [18] the original signal in

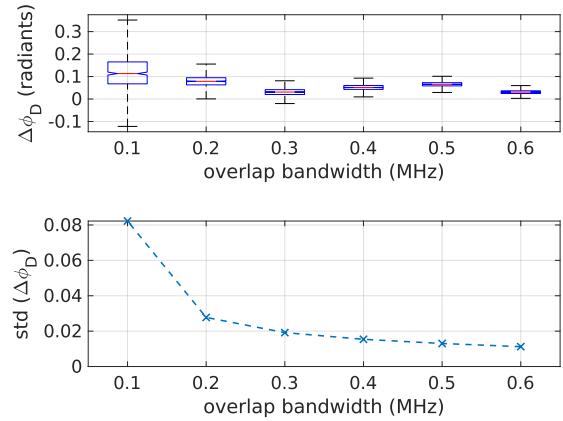


Figure 9: Phase delay evaluation vs. overlap area.

Receiver	Local Oscillator stability	Frequency error at 1030 MHz
RTL-SDR-v3	± 2 PPM	± 2,1 kHz
Ettus B210 (USRP)	± 2 PPM	± 2,1 kHz
HackRF One	± 20 PPM	± 20,06 kHz
Adalm-Pluto	± 25 PPM	± 25,75 kHz
RTL-SDR (blue edition)	± 60 PPM	± 61,80 kHz
RTL-SDR (black edition)	± 120 PPM	± 123,60 kHz

Table 1: Different receiver models and its frequency stability of the local oscillator.

post-processing. Fig. 8 shows how for the "no-upsampling" scenario the $\Delta_{samples}$ estimation fluctuates depending on the overlap bandwidth size, and how this behaviour is much more contained in the "upsampling" scenarios. This is an indicator that we need more accuracy in the $\Delta\hat{t}$ since the existence of synchronization errors at sample level could hinder the final collaborative reconstruction of the signal. For higher upsampling factors we can see how the $\Delta_{samples}$ remains stable for overlap bandwidths from 0.3 MHz onwards.

We have also performed an evaluation of the phase estimation of our system that becomes important for two reasons: 1) it gives the sub-sample synchronization among signals and 2) allows us to reconstruct and decode signals with phase coded information. Fig. 9 (top) shows the phase delay estimation between bursts (potential packets) using different overlap area sizes. We assume that the phase delay estimation among packets will be similar for each pair of receivers once the signal has been corrected in time. For overlap bandwidths over 0.3 MHz we can observe that the phase delay values are contained in the same range with no important fluctuations. However for overlap bandwidths smaller than 0.3 MHz we can observe how the values are spread over a large range, meaning that the estimation is not performed accurately. The smaller the overlap area is, the bigger the standard deviation of $\Delta\phi_D$ appears

³<https://www.ettus.com/product/details/UB210-KIT>

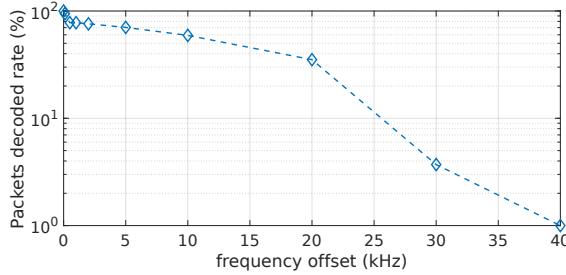


Figure 10: Frequency offset evaluation vs. packets decoded rate.

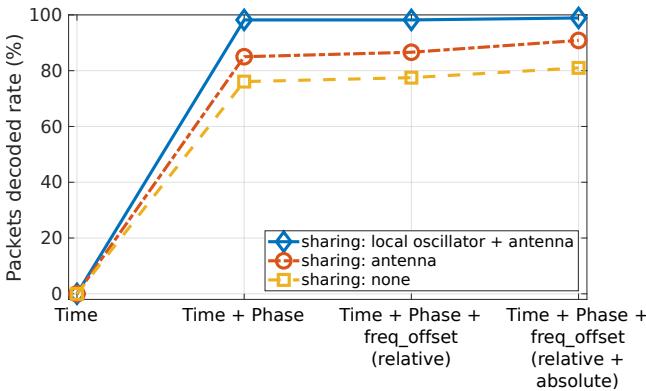


Figure 11: Mode-S uplink packets decoding rate in different scenarios (USRP used as baseline).

to be (see Fig. 9 bottom), due to the fewer data that the system has to properly perform the initial time offset estimation. It is possible this impact the subsequent phase delay estimation as the linear regression might have been computed with fewer data.

As we explained in Section 4, the final goal is to decode a signal that has been reconstructed collaboratively in the backend. Therefore, the packets decoded rate is a metric that must be evaluated in our system. As we mentioned before we use the Mode-S uplink decoder that we implemented (see Section 5.2) to evaluate the decoding of the signal. Fig. 10 shows the performance of the message decoding depending on the frequency offset among signals. In essence, we manually introduce a frequency offset in one of the signals and we check how the decoder performs in these scenarios. We can observe how the packet decoded rate is close to 100% when the frequency offset is below 1 kHz. The higher the frequency offset, the smaller the decoding packet rate. This proves the importance of a proper estimate for the frequency offset among signals caused by the local oscillator of the different receivers (see Table 1).

Fig. 11 shows the decoding packet rate for different experiment configurations (described before) and different signal synchronizations steps (explained in Sec. 4). The gentle case ("local oscillator + antenna") shows that is able to yields a 98% ratio of properly decoded packets once the time and phase are properly aligned. No improvements are visible even if the relative frequency offset is corrected since both receivers are sharing the local oscillator and

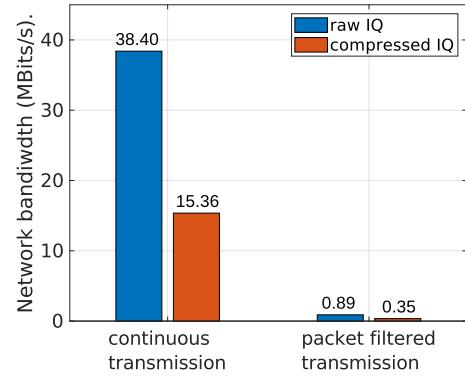


Figure 12: Network bandwidth used by one single sensor to deliver IQ data to the backend.

therefore their relative frequency offset is negligible. For the shared antenna ("antenna") scenario using non-coherent receivers we may observe that it reaches 85% of decoded packets ratio, and in this case correcting the frequency offset helps to reach the 90%. For the most unfavourable and challenging case ("none"), in which antennas are not shared and non-coherent receivers are used, we are able to reach more than 80% of correct packet decoding ratio by applying all the signal alignment steps as Fig. 11 shows.

5.5 Network Bandwidth

In order to perform the signal reconstruction in the backend, each sensor needs to send IQ data through the Internet connection. Our solution is based on a crowdsourcing approach where sensors are distributed among different locations and therefore they may use different Internet connections. Nevertheless the network bandwidth used for each sensor is a concern due to the IQ data transmission. Fig. 12 shows how one single sensor sampling at 2.4 MSPS with 8-bit resolution would need to transmit $\sim 39\text{Mbit/s}$, or $\sim 15\text{Mbit/s}$ if data is compressed with a generic DEFLATE [19] algorithm. Going further, packet detection is performed on the sensors, and then only small portions of IQ data are sent where potential packets can be decoded. This alleviates the network bandwidth used by each sensor. Assuming a detection packet rate of 700 packets/second and knowing that a Mode-S packet is $40\mu\text{s}$ long [6], the network bandwidth used for each sensor decreases to 0.89Mbit/s (0.35Mbit/s compressed-data), which is affordable over most domestic Internet connections.

6 DISCUSSION AND LIMITATIONS

In this paper, we have presented a crowdsourcing approach for collaborative signal decoding using non-coherent receivers, which has been tested in a real-world scenario. Further research is needed to understand the scope, opportunities and limitations of our approach by evaluating the following paths:

Validation in more complex scenarios. The evaluation of our approach has been performed in a scenario with two SDR receivers for collaborative signal decoding. Further experimentation using a greater number of receivers is needed to understand the scalability

of the approach, mainly when the synchronization parameters have to be estimated for signal recombinations requiring several overlap areas. In like manner, distance among sensors must be evaluated to understand the limit-range of deployment.

Testing different signals. Our evaluation has been conducted using real signals collected from aircraft to test our approach. However, future works should focus on evaluating our solution using other different and complex modulation schemes. In addition to that, an interesting future line of work would be to determine how often should synchronization parameters be estimated in continuous transmissions.

Increasing the decoded packet rate. Our approach is based on the idea of having the least possible information about the signal that is being reconstructed. This lack of knowledge may result in a smaller decoded packet rate. Future work on this topic may focus on increasing this rate by applying a closed-loop feedback control system that takes advantage of modulation specifications (e.g. by applying modulation classification algorithms [7]) and the result of the decoding process.

Network Bandwidth. As in any crowdsourcing system, the network bandwidth used for every sensor is a concern. We have proposed a solution (Section 5.5) to decrease drastically the network bandwidth for packet transmissions. However, further work on this topic may focus on decreasing the network bandwidth when the signal is based on continuous transmission (LTE, DVB-T, etc.), and therefore it is not possible to partially extract data to be sent to the backend.

7 RELATED WORK

Monitoring the electromagnetic spectrum at large scale is more challenging than traditional approaches. Several platforms use expensive and specialized hardware for sensing the spectrum at large scale such as DARPA’s Spectrum Challenge, Google Spectrum Database [20] and Microsoft Spectrum Observatory [21]. Our solution relies on low-cost receivers for sensing the spectrum distributively. In [22], the authors used a collaborative spectrum sensing approach for detecting unauthorized transmissions using Power-Spectral-Density (PSD) data. In this work, we propose to work with IQ data what allows us to reconstruct and decode signals. As IQ data can imply a deluge of information sent to the backend, we expect that it can be applied either in presence of very high uplink bandwidth or, most likely, for a short period of time, with the objective of decoding specific messages.

The authors of [23] proposed a method to receive and decode weak signals, by applying maximum ratio combining in the cloud. [24] proposed to save uplink bandwidth in each receiver by multiplexing the signal in time. Both papers used also IQ data, yet they tuned all receivers in the same frequency and assumed that the receiver can sample the whole signal of interest. Our problem space is complementary and requires dedicated solutions, as we deal with signals of higher bandwidth that the one single receiver can provide, a problem that emerges with the advent of low-cost spectrum sensors and has been so far largely ignored in the context of signal decoding.

Past work that considered wider bands focused on the problem of extracting information about the spectrum usage and which bands

are more interesting to monitor (e.g., applying a multi-armed bandit game [25]). Some other work proposed instead to use the sparse Fourier transform to decode signals [26]. However, they did not look at the problem of decoding signals that are of larger bandwidth than the single receiver, and required access to dedicated USRP hardware rather than low-cost commercial off-the-shelf hardware as in this work.

Different works [27], [28] had the need of using coherent-receivers to achieve a precise signal synchronization over different channels, e.g., passive radars using low cost receivers. In [9], authors used coherent-receivers and a GPS-discipline oscillator in order to synchronize the ADC clock in each channel. In contrast with these works, our solution uses non-coherent receivers and relies on the job done in the backend for the final synchronization and decoding of the signals.

8 CONCLUSION

We have studied the problem of collaborative sampling the spectrum using low-cost and distributed receivers to decode a wider signal bandwidth than the one of the single receiver. We have introduced the main concept of dividing the wideband signal in sub-frequency channels with a small overlap area, and addressed the main challenges to reconstruct the wideband signal in the backend. We have then presented a distributed architecture with the needed synchronization steps in order to align the sampled IQ signals from independent receivers and decode the whole signal in the backend. We have tested and evaluated our solution with real signals (Mode-S) sent by aircraft that are used for collision avoidance. We have implemented a Mode-S uplink decoder that is released as open source. Our results show that is feasible to use low-cost and distributed non-coherent receivers for decoding wideband signals, getting more than 80% of packets correctly decoded, and reaching nearly 100% of packets decoded when receivers share the local oscillator. Collaborative signal decoding using low cost receivers has been successfully proved and can enable multiple applications for those existing systems that deploy low-cost receivers such as RTL-SDR at large scale.

ACKNOWLEDGMENTS

This research work by IMDEA Networks Institute was sponsored in part by armasuisse under the Cyber and Information Research Program, the NATO Science for Peace and Security Programme under grant G5461, and Madrid Regional Government through TAPIR-CM project S2018/TCS-4496.

REFERENCES

- [1] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van den Bergh, H. Cordobés, D. Giustiniano, S. Pollin, and V. Lenders, “Electrosense: Open and big spectrum data,” *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, 2018.
- [2] *Rtl-sdr silver v3 specifications*, 2016. [Online]. Available: <http://www rtl-sdr com buy rtl-sdr-dvb-t-dongles/>.
- [3] N. Kleber, A. Termos, G. Martinez, J. Merritt, B. Hochwald, J. Chisum, A. Striegel, and J. N. Laneman, “Radiohound: A pervasive sensing platform for sub-6 ghz dynamic spectrum

- monitoring,” in *Dynamic Spectrum Access Networks (DySPAN), 2017 IEEE International Symposium on*, IEEE, 2017, pp. 1–2.
- [4] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, “Bringing up opensky: A large-scale ads-b sensor network for research,” in *Proceedings of the 13th international symposium on Information processing in sensor networks*, IEEE Press, 2014, pp. 83–94.
- [5] *Technical provisions for mode s services and extended squitter*, Doc 9871, First Edition, International Civil Aviation Organization, 2008.
- [6] I. C. A. Organization, *Aeronautical communications, volume iv, surveillance and collision avoidance systems*, 2007.
- [7] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, “Deep learning models for wireless signal classification with distributed low-cost spectrum sensors,” *IEEE Transactions on Cognitive Communications and Networking*, 2018.
- [8] S. Rajasegarar, C. Leckie, and M. Palaniswami, “Anomaly detection in wireless sensor networks,” *IEEE Wireless Communications*, vol. 15, no. 4, 2008.
- [9] K. Jamil, M. Alam, M. A. Hadi, and Z. O. Alhekail, “A multi-band multi-beam software-defined passive radar: part i: System design,” 2012.
- [10] D. Pfammatter, D. Giustiniano, and V. Lenders, “A software-defined sensor architecture for large-scale wideband spectrum monitoring,” in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN ’15, Seattle, Washington: ACM, 2015, pp. 71–82, ISBN: 978-1-4503-3475-4. doi: 10.1145/2737095.2737119. [Online]. Available: <http://doi.acm.org/10.1145/2737095.2737119>.
- [11] S. Haykin, *Communication Systems*, 5th. Wiley Publishing, 2009, ISBN: 0471697907, 9780471697909.
- [12] R. Calvo-Palomino, F. Ricciato, D. Giustiniano, and V. Lenders, “Ltess-track: A precise and fast frequency offset estimation for low-cost sdr platforms,” 2017.
- [13] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989, ISBN: 0-13-216292-X.
- [14] *Flightradar24 - track air traffic in real time*, 2018. [Online]. Available: <https://www.flightradar24.com/>.
- [15] *Flightaware - the world's largest flight tracking data company*, 2018. [Online]. Available: <http://www.flightradar24.com/>.
- [16] *Minimum operational performance standards for air traffic control radar beacon system / mode select (atcrbs / mode s) airborne equipment*, DO-181E, RTCA, 2011.
- [17] V. Dimitrakis, “Expanding the View on Avionics Communication The Mode-S Uplink,” Master’s thesis, ETH Zurich, 2017.
- [18] F. J. Harris, *Multirate Signal Processing for Communication Systems*, ser. Mobility ’08. Prentice Hall, 2004.
- [19] P. Deutsch, *DEFLATE Compressed Data Format Specification version 1.3*, RFC 1951 (Informational), Internet Engineering Task Force, May 1996. [Online]. Available: <http://www.ietf.org/rfc/rfc1951.txt>.
- [20] (2016). Google spectrum database, [Online]. Available: <https://www.google.com/get/spectrumdatabase/>.
- [21] (2015). Microsoft spectrum observatory, [Online]. Available: <http://observatory.microsoftspectrum.com/>.
- [22] A. Chakraborty, A. Bhattacharya, S. Kamal, S. R. Das, H. Gupta, and P. M. Djuric, “Spectrum patrolling with crowd-sourced spectrum sensors,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 1682–1690.
- [23] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, “Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks,” in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IEEE Press, 2018, pp. 60–71.
- [24] R. Calvo-Palomino, D. Giustiniano, V. Lenders, and A. Fakhredine, “Crowdsourcing spectrum data decoding,” in *INFOCOM 2017-IEEE Conference on Computer Communications*, IEEE, 2017, pp. 1–9.
- [25] L. Shi, P. Bahl, and D. Katabi, “Beyond sensing: Multi-ghz realtime spectrum analytics,” in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, Oakland, CA: USENIX Association, 2015, pp. 159–172, ISBN: 978-1-931971-218. [Online]. Available: <https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/shi>.
- [26] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, “Ghz-wide sensing and decoding using the sparse fourier transform,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 2256–2264. doi: 10.1109/INFOCOM.2014.6848169.
- [27] A. Capria, M. Conti, D. Petri, M. Martorella, F. Berizzi, E. Dalle Mese, R. Soletti, and V. Carulli, “Ship detection with dvb-t software defined passive radar,” in *IEEE Gold Remote Sensing Conference*, 2010.
- [28] G. Alois, V. Loscří, A. Borgia, E. Natalizio, S. Costanzo, P. Pace, G. Di Massa, and F. Spadafora, “Software defined radar: Synchronization issues and practical implementation,” in *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*, ser. CogART ’11, Barcelona, Spain: ACM, 2011, 48:1–48:5, ISBN: 978-1-4503-0912-7. doi: 10.1145/2093256.2093304. [Online]. Available: <http://doi.acm.org/10.1145/2093256.2093304>.