


Article

SCATTER PHY: An Open Source Physical Layer for the DARPA Spectrum Collaboration Challenge

Felipe A. P. de Figueiredo¹  0000-0002-2167-7286, Dragoslav Stojadinovic², Prasanthi Maddala², Ruben Mennes³, Irfan Jabandžić¹, Xianjun Jiao¹, and Ingrid Moerman¹

¹ Ghent University - imec, IDLab, Department of Information Technology, Ghent, Belgium.; [felipe.pereira, irfan.jabandzic, xianjun.jiao, ingrid.moerman]@ugent.be

² WINLAB, Rutgers University, New Brunswick, NJ, USA.; [stojadin, prasanthi]@winlab.rutgers.edu

³ Department of Mathematics and Computer Science, University of Antwerp - iMinds, Antwerp, Belgium.; ruben.mennes@uantwerpen.be

* Correspondence: felipe.pereira@ugent.be

Abstract: DARPA, the Defense Advanced Research Projects Agency from the United States, has started the Spectrum Collaboration Challenge with the aim to encourage research and development of coexistence and collaboration techniques of heterogeneous networks in the same wireless spectrum bands. Team SCATTER has been participating in the challenge since its beginning, back in 2016. SCATTER's open-source software-defined physical layer (SCATTER PHY) has been developed as a standalone application, with the ability to communicate with higher layers of SCATTER's system via ZeroMQ, and uses USRP X310 software-defined radio devices to send and receive wireless signals. SCATTER PHY relies on USRP's ability to schedule timed commands, uses both physical interfaces of the radio devices, utilizes the radio's internal FPGA board to implement custom high-performance filtering blocks in order to increase its spectral efficiency as well as enable reliable usage of neighboring spectrum bands. This paper describes the design and main features of SCATTER PHY and showcases the experiments performed to verify the achieved benefits.

Keywords: Cognitive Radios; Collaborative Intelligent Radio Networks; Spectrum Sharing; Coexistence; Experimental Evaluation.

1. Introduction

In the era of a growing number of wireless communication networks and protocols, spectrum scarcity represents an ever-increasing challenge in the research community. Defense Advanced Research Project Agency (DARPA) has established the Spectrum Collaboration Challenge (SC2) as a collaborative machine learning competition, where different teams from the entire world build their own wireless communication networks to compete with other teams in achieving the highest possible throughput not only for their own team, but also for other teams coexisting in the same spectrum bands. The teams were motivated to set up their networks to be able to not only coexist with other networks, but also use collaboration to ensure that each network maximizes its throughput while facilitating spectrum usage and securing its availability when necessary to other networks and systems. The teams would devise new spectrum access strategies, which will allow radio networks to autonomously collaborate and dynamically determine how the Radio Frequency (RF) spectrum is being used, which enables radios to avoid interference and find better transmission opportunities. The SC2 was run on Colosseum, a testbed with a large number of nodes equipped with Software Defined Radio (SDR) devices, where teams can deploy their networks and use SDRs for their wireless communication. The testbed is able to emulate any desired geographical topology of nodes with

diverse physical environments, enabling DARPA to test teams' networks and their performance with and against other networks in a wide variety of scenarios [1].

The SCATTER team has designed and built a multi-layered system, with physical, MAC, AI and other layers developed and implemented as completely autonomous systems, but interconnected with a publish-subscribe messaging system, in particular – ZeroMQ [2]. Thus, SCATTER's open-source software-defined physical layer (PHY) was designed as an independent system, able to transmit and receive packets over the air using USRP SDR devices. The design of SCATTER PHY uses the srsLTE, an existing software implementation of the LTE protocols, as the basis for its communication, and builds on top of it with various modifications to introduce features and mechanisms targeted for the SC2. The ultimate goal of these features is to present an API of the required physical layer functionality to other layers (mainly the MAC and AI layer) which are oblivious to the implementation details of those functions. For example, SCATTER PHY uses custom-built FPGA filters to reduce out-of-band emissions and maximize channel utilization. When MAC layer requests a change of the channel bandwidth, these filters are automatically adapted to accommodate the new bandwidth. Similarly, the physical layer provides spectrum measurement data averaged over time to the AI layer, which uses it to determine the most efficient channel selection algorithms. Spectrum measurement data is obtained by again using the custom built FPGA blocks which perform the FFT and average the obtained power spectral density over multiple measurements to reduce the amount of data sent to AI. This paper describes the design and features of SCATTER PHY and experimentally verifies the benefits achieved with the outlined design decisions.

The main contributions of this work are listed next.

- Presentation of an open-source highly on-line configurable SDR-based physical layer for intelligent spectrum sharing research and development. The main idea behind the development of the proposed physical layer is to provide developers and researchers with a starting point so that they can mainly focus their attention to developing intelligence and reasoning algorithms to combat spectrum scarcity in the context of next generation wireless and mobile networks. The proposed physical layer can be employed in prototyping and experimentally testing intelligent algorithms for dynamic optimization of spectrum usage. Its source code can be found at its GitHub project page [3].
- Proposal of a two-stage synchronization sequence detection algorithm, which uses a Cell-Average Constant False Alarm Rate (CA-CFAR) algorithm in the second stage in order to improve the detection performance when compared to an approach that only employs plain correlation to detect the presence of the synchronization sequence.
- Extensive experimental results showing the effectiveness of the proposed physical layer. The presented and discussed results include experiments with a filtered form of OFDM waveform, measurements of the throughput and CPU/Memory utilization, etc.

The remainder of the paper is organized as follows: section 2 presents some related pieces of work that apply artificial intelligence algorithms to the spectrum sharing problem, section 3 describes the design of SCATTER PHY in detail and specifies its main features. Section 4 showcases the experimental verification of the achieved benefits. Section 5 concludes the paper.

2. Related Work

In [4] the authors propose the use of deep reinforcement learning to design an universal MAC protocol referred there as Deep-reinforcement Learning Multiple Access (DLMA). They show that the proposed DLMA scheme is able to learn the optimal MAC strategy for harmonious co-existence with TDMA and ALOHA networks.

The authors of [5] employ a deep learning neural network (DLNN) to the slot prediction task. They demonstrate that their trained DLNN does online learning and can accurately predict, through spectrum monitoring, the behavior of spectrum usage (*i.e.*, slot transmissions) one second in advance in the context of Multiple Frequency Time Division Multiple Access (MF-TDMA) networks.

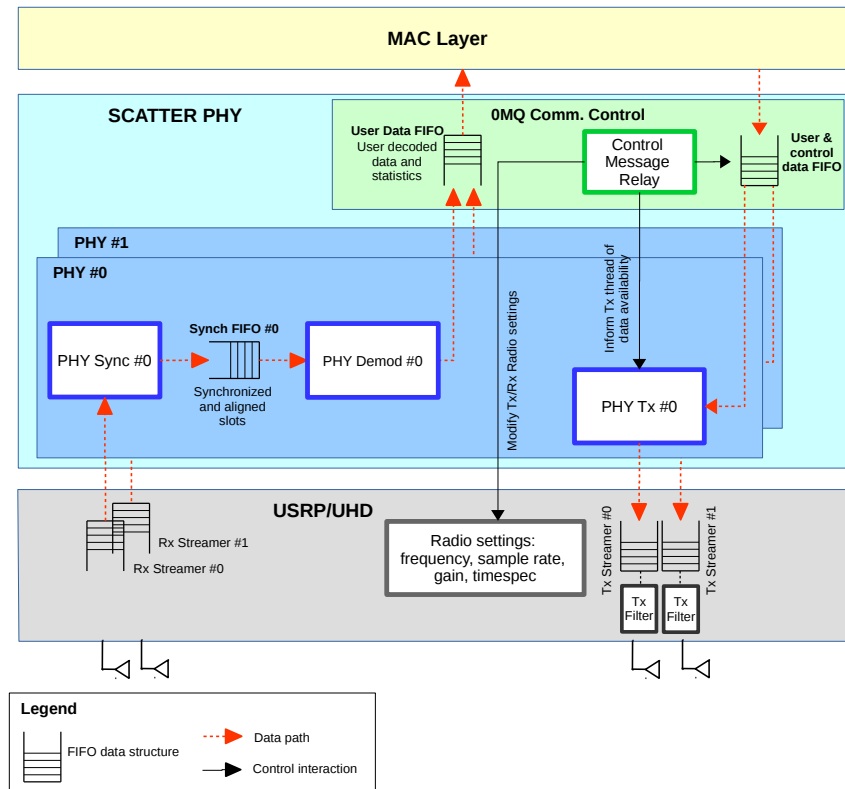


Figure 1. High-level threading architecture of the SCATTER PHY.

In the work presented in [6], the authors deal with the problem involving the recognition of different radio access technologies (RATs) employing Deep Autoencoders (DAEs). Their DAE model is used in a semi-supervised learning (SSL) approach for the recognition of wireless technologies using raw IQ samples. The results show that the proposed DAE approach achieves similar accuracy when compared with supervised learning (SL). They also demonstrate that their approach outperforms the SL approach for negative Signal to Noise Ratio (SNR) values.

3. SCATTER PHY

The high-level architecture of the SCATTER PHY is depicted in Figure 1. The SCATTER PHY is an open-source software-defined physical layer [3], which is implemented using Universal Software Radio Peripheral (USRP) Hardware Driver (UHD) software Application Programming Interface (API) [7,8] and runs on top of Ettus USRP X family of software-defined radio (SDR) devices including NI's RIO platforms [9,10] and communicates with it through the UHD driver and its APIs [11]. As can be seen in the figure, the individual PHY modules are connected to the ZeroMQ (Data/Control) module, also known as 0MQ, which interconnects the SCATTER PHY with the MAC layer through the ZeroMQ bus [12]. This module manages the exchange of control and statistics messages between the SCATTER PHY and the MAC layer.

Communication with the SCATTER PHY is entirely realized through a well-defined interface designed with Google's Protocol Buffers (protobuf) [13] for data serialization coupled with the ZeroMQ messaging library [12] for distributed exchange of control, statistics and data messages. Implementing the ZeroMQ push-pull pattern allows local or remote MAC layer's real-time configuration of several parameters and reading of several pieces of information/statistics provided by the SCATTER PHY. Based on the ZeroMQ logic, PHY and MAC layers are able to exchange control and data messages following a non-blocking communication paradigm. The SCATTER PHY was designed to be totally decoupled and independent of the MAC layer module, not posing any constraints on hardware, software and/or programming language adopted by it.

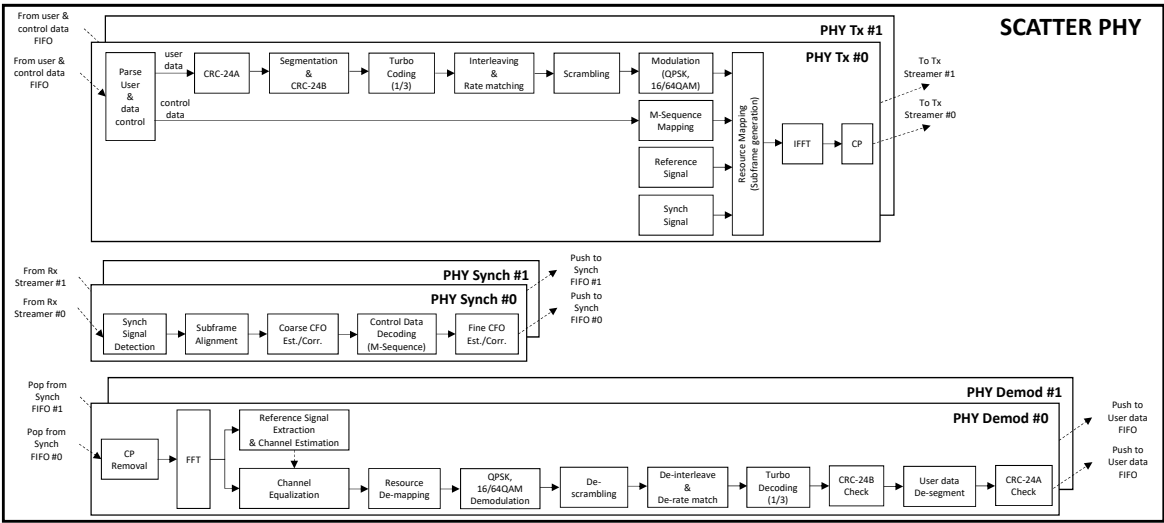


Figure 2. Block diagram of the SCATTER PHY transmitter and receiver sides.

The SCATTER PHY presents the following set of main features:

- Bursty transmissions: with discontinuous transmissions, it is possible to have better use of the available spectrum band and to coordinate its usage with other networks/radios in an opportunistic/intelligent/collaborative way.
- Dual-Concurrent PHYs: a Multi-Concurrent-Frequency Time-Division Multiple Access (McF-TDMA) scheme can be implemented by the MAC layer by having two PHYs simultaneously transmitting and receiving at independent frequencies. This ability for concurrent allocations allows for a smarter spectrum utilization as vacant disjoint frequency chunks can be concurrently used.
- FPGA-based Filtered transmissions: filtering the transmitted signal effectively minimizes out-of-band emissions (OOBE), which allows for a better spectrum utilization, once radios can have their transmissions closer to each other in the frequency domain, reducing spectrum wastage.
- Out-of-Band Full-Duplex operation: both PHYs operate totally independently, meaning, that Tx and Rx modules are able to transmit and receive at different channels, set different Tx and Rx gains and use different PHY BWs.
- Timed-commands: this feature allows the configuration in advance of the exact time to (i) start a transmission and (ii) change Tx/Rx frequencies/gains. This allows the MAC layer to implement a TDMA scheme.

Figure 1 illustrates the different software/hardware layers composing the SCATTER PHY and the threads within each one of them. Red dashed arrows indicate data paths while black arrows indicate control/information interaction between threads.

SCATTER PHY is a discontinuous transmission-based PHY, which transmits data bursts in small transport units called subframes. A subframe is a container through which user and control data is exchanged in the network.

The SCATTER PHY is built upon the srsLTE library [14], and therefore, absorbs and evolves on top of the existing LTE features. srsLTE is a free and open-source LTE software library developed by Software Radio Systems (SRS) [14]. We adopt Orthogonal Frequency-Division Multiplexing (OFDM) as the SCATTER PHY waveform. OFDM is a mature technology, which is vastly implemented in a great number of products due to its several advantages such as robustness to severe multipath fading, low implementation complexity, easy integration with MIMO, simple channel estimation, etc. [15]. User and control data are mapped into subcarriers over 14 OFDM symbols spanning 1 ms. The control

data signal carries the used MCS for the current transmission and the number of subsequent subframes with user data modulated with that specific MCS.

The control data signal is used at each one of the PHY receivers to automatically detect the number of allocated RBs, the location of the allocated RBs in the resource grid, and the MCS used to transmit data of a specific user.

By embedding the control information into the transmitted signal, the Medium Access Control (MAC) layer does not need to know in advance the number of subframes and MCS in a given COT. Upon correct user data decoding, each PHY informs the MAC layer of the number of received bytes, the corresponding MCS, and decoding statistics (*e.g.*, RSSI, SINR, number of detection/decoding errors, etc.).

SCATTER PHY employs discontinuous (*i.e.*, bursty) transmission of subframes. A subframe is the basic transmission unit of the SCATTER PHY and each one is 1 ms long. SCATTER PHY works with two types of subframes, namely, synchronization and data-only subframes. A synchronization subframe carries the synchronization signal, reference signal, control, and user data. A data-only subframe carries the reference signal and user data. The synchronization signal is a 72 symbols long sequence that is generated using Zadof-Chu (ZC) sequences [16,17]. Therefore, the synchronization sequence is generated according to

$$x_s = e^{-j\frac{\pi un(n+1)}{N_{\text{synch}}}}, 0 \leq n \leq N_{\text{synch}} - 1, \quad (1)$$

where u is the ZC sequence index, n is the sequence sample index, and N_{synch} is the length of the synchronization sequence.

The control signal is based on maximum length sequences (M-sequences) [18], where two M-sequences, each of length 31, are used to transmit the information necessary to decode the user data. The reference signal is used to estimate the channel and then equalize the received signal so that channel interference to the desired signal is minimized.

The SCATTER PHY allows bursty transmissions with variable a Channel Occupancy Time (COT), *i.e.*, the number of subframes to be transmitted in sequence without any gap (*i.e.*, idle time) between them is variable. The number of subframes in a COT is derived based on MCS, PHY BW and data length (*i.e.*, number of bits to be transmitted) parameters sent by the MAC layer in the control message. The minimum COT is equal to 1 ms and is equivalent to the synchronization subframe. Variable COT enables the support of different traffic loads and channel detection. Every subframe can carry a pre-defined number of bits, which is based on the MCS and current PHY BW.

Each one of the PHY modules is split into three sub-modules, namely, PHY Tx, PHY Rx Synchronization and PHY Rx Demodulation where each one of them runs on an exclusive, standalone thread. The reason for having a multi-threaded PHY implementation is that it allows independent critical and/or time-consuming tasks to be executed simultaneously (*i.e.*, concurrently), which improves computing performance and efficiency. Allied with multi-core enabled Central Processing Units (CPUs), the multi-threaded PHYs naturally support full-duplex communications mode, *i.e.*, each one of the two PHYs can simultaneously transmit and receive at different frequencies, which consequently results in higher throughput. The PHY Tx threads (#0 and #1) are responsible for modulation and transmission of data (*i.e.*, user and control data). The PHY Rx Synchronization threads are responsible for the detection of the synchronization (Synch) signal, decoding of the control data, Carrier Frequency Offset (CFO) estimation/correction and subframe time-alignment tasks. Detection of the Synch signal is carried out through a two-stage detection algorithm, which at the first stage correlates the received signal with a locally-stored version of the synchronization subframe with no data and control signals. If the peak-to-side-lobe ratio (PSR) is greater than a constant threshold, then, the second stage applies a CA-CFAR algorithm to the OFDM symbol carrying the Synch signal [16]. The two-stage approach employed by the SCATTER PHY improves the Synch signal detection when

Table 1. OFDM Modulation Parameters.

PHY BW [MHz]	1.26	2.7	4.5	9
Subframe duration [ms]	1			
subcarrier spacing [KHz]	15			
Sample rate [Msps]	1.92	3.84	5.76	11.52
FFT size	128	256	384	768
Useful subcarriers	84	180	300	600
OFDM symbols per subframe	14			
CP duration [us]	4.69 (12 symbols) / 5.21 (2 symbols)			

176 compared to a detection approach that only uses the PSR of the correlation calculated at the first stage.
177 The two-stage approach employed by the SCATTER PHY is described in Appendix A.

178 The CFO estimation task is split into coarse and fine estimations/corrections, where the coarse
179 estimation is based on the Synch signal and the fine estimation is based on the Cyclic Prefix (CP)
180 portion of the OFDM symbols [19]. The integer part of the frequency offset (*i.e.*, integer multiples of
181 the subcarrier spacing) is estimated and corrected by the coarse CFO algorithm, which is based on the
182 maximization of the correlation of the received synchronization signal with several locally generated
183 frequency offset versions of it. On the other hand, the fractional frequency offset (*i.e.*, offset values less
184 than one half of the subcarrier spacing) is estimated and corrected by the fine CFO algorithm, which is
185 based on the phase difference of the correlation between the CP and the last part of the OFDM symbol
186 (*i.e.*, the portion used to create the CP). Integer and fractional CFO estimation methods are described
187 in Appendix B.

188 The PHY Rx Demodulation threads take care of user data demodulation, *i.e.*, OFDM demodulation
189 (FFT processing and CP removal), channel estimation/equalization, resource demapping, symbol
190 demodulation, de-scrambling, de-interleaving/de-rate matching, turbo decoding, de-segmentation
191 and, CRC checking. Each PHY receives data and control messages from the ZeroMQ Data/Control
192 module. Decoded user data and statics related to the PHY operation (Rx/Tx statistics) are sent directly
193 to the upper layers through the 0MQ bus. Figure 2 depicts block diagrams of the SCATTER PHY
194 transmitter (PHY Tx thread) and receiver sides (PHY Sync and PHY Demod threads).

195 Regarding numerology, the SCATTER PHY uses a subcarrier spacing of 15 kHz, a CP of 5.2 μ s and
196 supports 1.26, 2.7, 4.5, and 9 MHz bandwidths. Each PHY channel bandwidth (BW) can be changed
197 through the command line at startup, or in real-time, through Tx control messages. Table 1 summarizes
198 the OFDM modulation parameters of the SCATTER PHY.

199 SCATTER PHY frame structure is depicted in Figure 3. As can be seen, synchronization
200 and control data signals are added only to the very first subframe of a COT. With this frame
201 structure, synchronization (*i.e.*, detection of synchronization signal, time-alignment and CFO

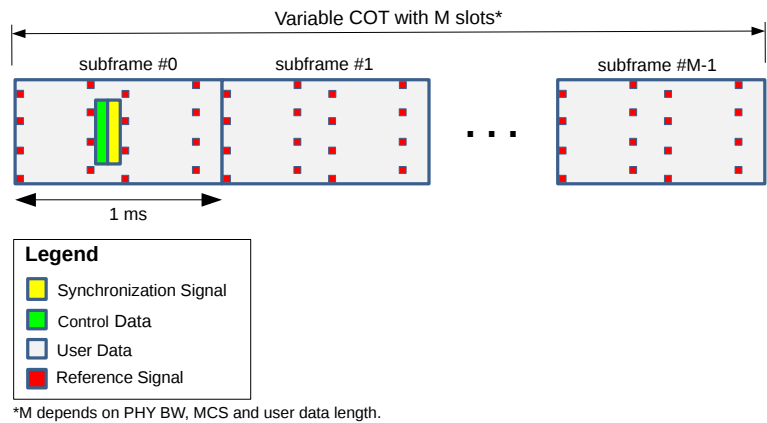


Figure 3. SCATTER PHY frame structure.

Table 2. Modulation Code Scheme and their Respective Code Rate.

MCS	Modulation	Code Rate			
		1.26 MHz	2.7 MHz	4.5 MHz	9 MHz
0	QPSK	0.0857	0.0900	0.0940	0.0940
1	QPSK	0.1143	0.1200	0.1220	0.1220
2	QPSK	0.1357	0.1467	0.1480	0.1480
3	QPSK	0.1714	0.1933	0.1920	0.1920
4	QPSK	0.2107	0.2400	0.2400	0.2440
5	QPSK	0.2571	0.2933	0.2960	0.2920
6	QPSK	0.3071	0.3400	0.3440	0.3440
7	QPSK	0.3571	0.4000	0.4160	0.4130
8	QPSK	0.4071	0.4667	0.4640	0.4690
9	QPSK	0.4714	0.5200	0.5360	0.5330
10	16QAM	0.2357	0.2600	0.2680	0.2665
11	16QAM	0.2571	0.2933	0.2920	0.2905
12	16QAM	0.3000	0.3267	0.3280	0.3305
13	16QAM	0.3357	0.3667	0.3800	0.3825
14	16QAM	0.3857	0.4267	0.4290	0.4298
15	16QAM	0.4286	0.4667	0.4770	0.4718
16	16QAM	0.4429	0.5000	0.5090	0.5078
17	64QAM	0.2952	0.3333	0.3393	0.3385
18	64QAM	0.3238	0.3600	0.3553	0.3625
19	64QAM	0.3524	0.3911	0.4033	0.4087
20	64QAM	0.3905	0.4411	0.4353	0.4407
21	64QAM	0.4286	0.4678	0.4727	0.4727
22	64QAM	0.4571	0.5122	0.5047	0.5100
23	64QAM	0.4952	0.5478	0.5570	0.5642
24	64QAM	0.5333	0.5833	0.5970	0.6042
25	64QAM	0.5714	0.6189	0.6210	0.6308
26	64QAM	0.5905	0.6633	0.6690	0.6770
27	64QAM	0.6190	0.6900	0.7010	0.7010
28	64QAM	0.6571	0.8056	0.8067	0.8178
29	64QAM	0.6952	0.8322	0.8280	0.8552
30	64QAM	0.7333	0.8617	0.8493	0.8925
31	64QAM	0.7714	0.8883	0.8707	0.9240

estimation/correction) and control data decoding only happen for subframe #0, *i.e.*, the synchronization subframe.

The Payload Data Unit (PDU) adopted by SCATTER PHY is a Transport Block (TB). Therefore, a TB is the payload coming from the MAC layer and given to the individual PHYs to be encoded and transmitted over the air. One TB consists of a number of bits that can be accommodated within a 1 ms long subframe given the selected PHY BW and MCS. Therefore, given the PHY BW and the desired MCS, the MAC layer can find the number of bits that can be handled by a 1 ms long subframe. Table 2 presents the coding rate for each one of the 32 defined MCS values.

The communication between the SCATTER PHY and the MAC layer is carried out through the exchange of four pre-defined messages. The first two, namely, Tx and Rx Control messages, are used to manage subframe transmission and reception respectively. The parameters carried by these two messages can be configured and sent to the individual PHYs by the MAC layer before the transmission of every subframe, hence allowing runtime configuration. The other two messages, namely, Tx and Rx statistics messages, are used to provide real-time feedback from each PHY to the MCA layer, yielding vital information necessary for such layer to take actions.

Tx control messages carry the user data (*i.e.*, TB) to be transmitted and Tx parameters related to that transmission, namely, PHY ID, MCS, data length, Tx gain, Tx channel, Tx PHY BW, and transmission timestamp. The transmission timestamp parameter enables time-scheduled transmissions, which allows the MAC layer to implement a Multi-Frequency (MF) Time Division Multiple Access

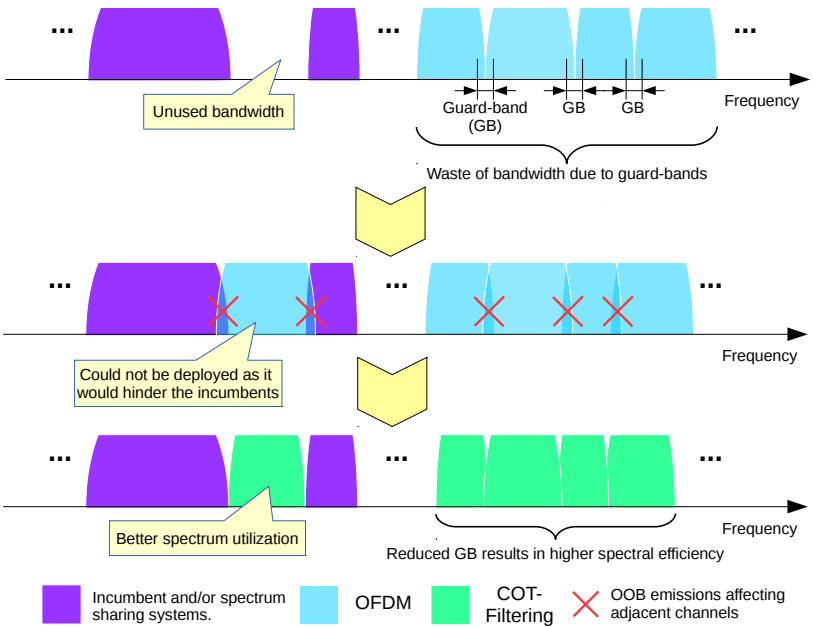


Figure 4. Closer coexistence to other radios due to reduced OOB emissions.

(TDMA) medium access scheme. **Rx control** messages are used to configure Rx channel, Rx gain, Maximum number of turbo decoder iterations, enable or disable Rx combining, and Rx PHY BW of a specific PHY, which addressed through the PHY ID parameter. The PHY ID parameter is used to specify to each one of the two PHYs a control message is meant to.

The other two messages, namely, Rx and Tx statistics, are used to inform the MAC layer of the Rx and Tx processing results of each PHY, respectively. **Rx statistics** messages carry the PHY ID, received data and reception statistics related to the received data such as Channel Quality Indicator (CQI), Received Signal Strength Indication (RSSI), decoded MCS, subframe error counter, decoding time, number of turbo decoder iterations, etc. **Tx statistics** messages inform upper layers of transmission statistics like coding time, the total number of transmitted subframes of each specific PHY. Table 3 summarizes all the real-time configurable parameters and statistics offered by the SCATTER PHY.

3.1. COT-based Filtering

OFDM-based waveforms are not suited for spectral coexistence due to their poor spectral localization [20]. This problem is caused by the rectangular pulse-shape used in OFDM, which leads to a sinc-pulse property in the frequency domain with a very low second lobe attenuation of -13 dB [21].

Therefore, in order to guarantee a optimal spectral localization, *i.e.*, lower out-of-band (OOB) emissions, and maintain the complex-domain orthogonality of the OFDM symbols is to apply some sort of filtering to the time domain subframes. The filtering process is applied to each COT of each PHY independently. The subframes comprising a COT are generated at the SW level and then filtered at the HW level, by a 128 order FPGA-based FIR filter. The COT-based filtering improves the closer coexistence with other radios (either belonging to our team or others), allowing radio transmissions to be closer in frequency. The filter used in the SCATTER PHY is designed and explained in [22].

¹ Depends on the USRP daughter board model installed [11].
² The numbers correspond to the following bandwidths: 1.26, 2.7, 4.5, and 9 MHz respectively.

This co-design SW/HW is used so that fast-processing high order filters can be implemented adding up very low latency to the transmission chain and still allowing the flexibility of the software-defined PHYs.

The filter's coefficients applied to the COT are automatically selected according to the configured Tx PHY BW (*i.e.*, the coefficients are selected in real-time based on the Tx PHY BW field in the Tx control message) as it needs to have its cut-off frequency changed to exactly filter the desired signal's bandwidth.

The normalized FIR filter's coefficients used in the COT-based filtering are given in time-domain as [7]

$$f(n) = \frac{p(n).w(n)}{\sum_k p(k).w(k)}. \quad (2)$$

where $p(n)$ is the impulse response of the sinc signal and $w(n)$ is the window used to truncate the sinc signal. These two signals are given as

$$p(n) = \begin{cases} \sin\left(\frac{\pi[N_U+N_e]n}{N_{\text{FFT}}}\right) / \frac{\pi[N_U+N_e]n}{N_{\text{FFT}}}, & n \neq 0, \\ 1, & n = 0. \end{cases} \quad (3)$$

$$w(n) = \left\{ \frac{1}{2} \left[1 + \cos\left(\frac{2\pi n}{L-1}\right) \right] \right\}^{0.6}, \quad (4)$$

where N_U is the number of useful subcarriers (see Table 1), N_{FFT} is the length of the FFT used in the OFDM modulation (see Table 1), N_e is the excess bandwidth in number of subcarriers, L is the length of the FIR filter, and $-\frac{(L-1)}{2} \leq n \leq \frac{(L-1)}{2}$. The excess bandwidth is used to extend the flat region of the filter so that the subcarriers at the left and right borders of the OFDM symbols suffer less with attenuation.

Table 3. SCATTER PHY real-time configurable parameters and statistics.

Message	Parameter	Type	Unit	Range
Tx control	PHY ID	uint32	-	0-1
	MCS	uint8	-	0-28
	Tx gain	uint32	dB	depends on HW ¹
	Tx channel	uint32	-	≥ 0
	Tx PHY BW	uint8	MHz	0-3 ²
	Transmission timestamp	uint64	s	≥ 0
	User Data length	uint32	-	> 0
	User data	uchar[]	-	uchar range
Rx control	PHY ID	uint32	-	0-1
	Rx channel	uint32	-	≥ 0
	Rx gain	uint32	dB	depends on HW ¹
	Rx PHY BW	uint8	MHz	0-3 ²
	Max. # of turbo decoder iterations	uint32	-	uint32 range
	Enable Rx combining	boolean	-	True/False
Rx statistics	PHY ID	uint32	-	0-1
	CQI	uint8	-	0-15
	RSSI	float	dBW	float range
	Noise	float	dBW	float range
	Decoded MCS	uint8	-	0-28
	Subframe error counter	uint32	-	≥ 0
	Decoding time	uint32	ms	≥ 0
	Data length	uint32	-	≥ 0
	# of turbo decoder iterations	uint32	-	uint32 range
	Received data	uchar[]	-	uchar range
Tx statistics	PHY ID	uint32	-	0-1
	Coding time	uint32	ms	≥ 0
	Number of transmitted subframes	uint32	-	≥ 0

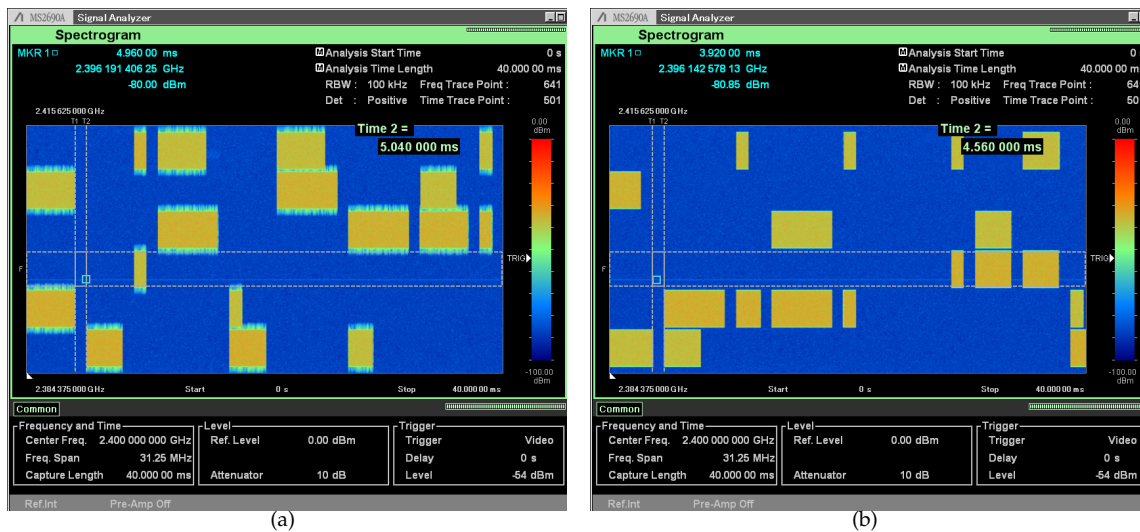


Figure 5. Comparison of McF-TDMA feature without and with filtering. (a) FPGA-based filters disabled. (b) 128 order FPGA-based FIR filter enabled.

The filter, $f(n)$, exhibits the following properties (i) a flat pass-band over the useful subcarriers, (ii) a sharp transition-band, which minimizes the necessary guard-bands, (iii) sufficient stop-band attenuation, and (iv) time duration of only a fraction of the duration of an OFDM symbol [7].

3.2. Benefits of the SCATTER PHY

The first advantage of the proposed PHY is that the COT-based filtering makes the SCATTER PHY more spectral efficient as the OOB emissions are reduced. OOB emissions interfere with nearby radios (*i.e.*, radios with nearby channels), decreasing the quality of their received signals, which consequently impacts the throughput experienced by those radios. The reduced OOB emissions make SCATTER PHY ideal for coexistence with other radios, allowing it to operate closer to them in the frequency domain and consequently, reducing spectrum wastage while increasing the spectral efficiency as shown in Figure 4.

The second advantage offered by SCATTER PHY is the possibility to configure in real-time all PHY parameters through the control messages.

4. Experiment Results

In this section, we present some experimental results in order to demonstrate the effectiveness and usability of the SCATTER PHY. All the experiments presented here were carried out with the framework running on servers with Intel Xeon E5-2650 v4 CPUs (@2.2 GHz, 30 M cache, 9.60 GT/s QPI, Turbo, HT, 12 Cores/24 Threads, 105 Watts) with 128 GB of RAM memory connected to x310 USRPs with 10 Gigabit Ethernet links, and equipped with CBX-120 RF daughterboards [23].

Figure 5 shows two spectrograms collected during 40 ms over a 31.25 MHz bandwidth with both PHYs set to operate with six 4.5 MHz channels. In this experiment, each PHY transmits a random number of subframes at randomly selected channels. The channel number and the number of transmitted subframes, *i.e.*, COT, are randomly selected between the ranges 0-5 and 1-3, respectively. Here, a gap of 1 ms between consecutive transmission is used. The figures were collected with the center frequency of both Tx PHYs set to 2.4 GHz, equal Tx gains of 3 dB, and both USRP Tx outputs connected to the signal analyzer through an RF combiner and a cable with 20 dB of attenuation. As can be noticed in both figures, (a) and (b), SCATTER PHY is able to independently transmit at two distinct channels with a different number of subframes. Figure 5 (a) shows the case where no filtering is enabled. As can be noticed, OOB might cause interference to adjacent channels and consequently

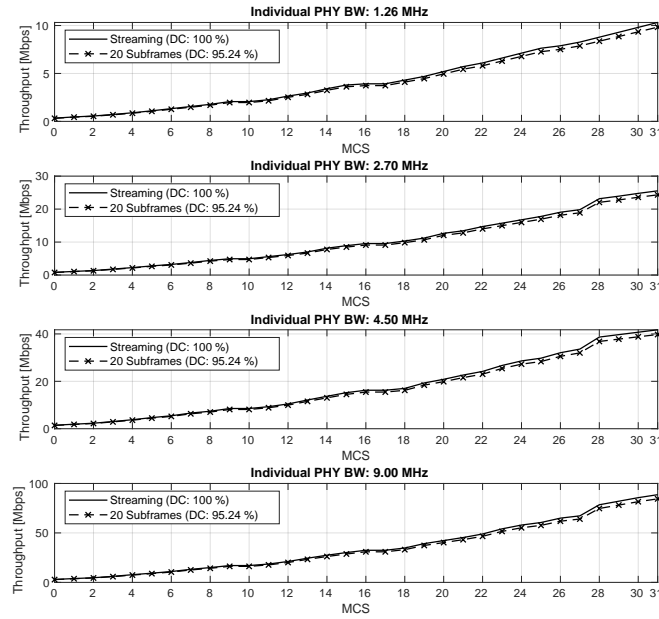


Figure 6. SCATTER PHY throughput for several bandwidth, MCS, and DC values.

decrease other radios' throughput. Figure 5 (b) shows the case when the FPGA-based FIR filters are enabled. As can be seen, when filtering is enabled, the OOB is mitigated and consequently, interference to and from adjacent channels is also mitigated. Additionally, another consequence of the filtering use is that the channel spacing could be made smaller and consequently decrease the spectrum wastage.

Figure 6 depicts throughput measurements taken with the SCATTER PHY for several PHY BW, MCS, and Duty Cycle (DC) values working in full-duplex mode (*i.e.*, the two independent PHYs are simultaneously transmitting and receiving). We employ the full-duplex mode in order to check if it impacts somehow the measured throughput once in full-duplex mode the SCATTER PHY is being fully utilized. The measurement is taken for a COT of 20 ms with a gap of 1 ms between subsequent transmissions. This means a DC of 95.24 %. The throughput is averaged over 10 measurement intervals of 10 seconds each. During one measurement interval, the number of received bits is counted and then divided by the interval to calculate the throughput measured during that interval. In the figure, for comparison reasons, we also add the theoretical maximum throughput achieved by the *Streaming* mode, where there is no gap between subsequent transmissions, *i.e.*, a DC equal to 100 %. The theoretical maximum throughput is calculated by dividing the TB size in bits for each MCS by 1 ms. As expected, the measured SCATTER PHY's throughput is close to the theoretical maximum throughput for all MCS and PHY BW values, yielding more than 84 Mbps for a BW of 9 MHz and MCS 31. Additionally, the operation in full-duplex mode has no visible impact on the achieved throughput. This is due to the powerful server, with 12 cores, used to run the SCATTER PHY.

In Figure 7 we compare the spectrogram and spectrum of non-filtered and filtered transmissions collected with an Anritsu MS2690A Signal Analyzer. The figures were collected with a Tx center frequency of 1.9925 GHz, a Tx gain of 3 dB with the USRP Tx output connected to the signal analyzer through a cable with 20 dB of attenuation. It is easily seen that the OFDM side lobes (*i.e.*, OOB emissions) are reduced with the use of a 128 order FPGA-based FIR filter.

Figure 8 compares the mean absolute percentage error (MAPE) of the fine CP-based fractional CFO estimation method against fractional CFO estimation based on the synchronization signal. The MAPE for the CFO estimation is defined as

$$\text{MAPE \%} = \frac{100}{N} \sum_{i=1}^N \left| \frac{\text{CFO}^{\text{est.}}(i) - \text{CFO}(i)}{\text{CFO}(i)} \right|, \quad (5)$$

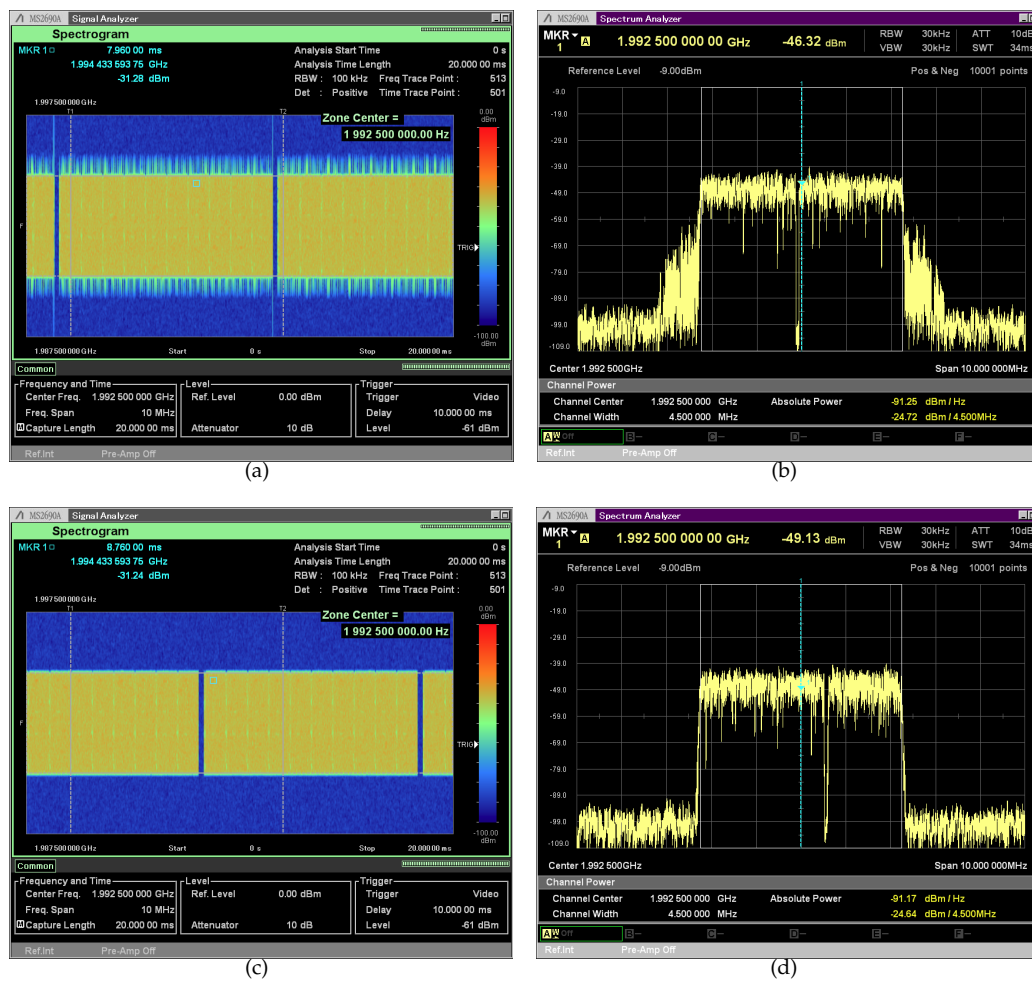


Figure 7. Comparison between non-filtered and filtered-based transmissions. (a) Spectrogram with no filter. (b) Spectrum with no filter enabled. (c) Spectrogram with 128 order FPGA-based FIR filter enabled. (d) Spectrum with 128 order FPGA-based FIR filter enabled.

where $\text{CFO}^{\text{est}}(i)$ is the estimated CFO value for the i th trial, $\text{CFO}(i)$ is the randomly generated CFO, which is applied to the transmitted signal, for the i th trial, and N is the number of trials over which the CFO is averaged. The results were obtained by connecting the Tx port to the Rx port of the same USRP so that the frequency offset caused by the HW is minimal or nonexistent and adding Additive white Gaussian noise (AWGN) plus the desired frequency offset at the SW level just before the subframes are transmitted. The CFO applied to the signal is drawn from a uniform distribution varying from -7.5 kHz to 7.5 kHz, *i.e.*, \pm half subcarrier spacing. As can be seen, the fine CFO estimation algorithm, which is based on the CP portion of the OFDM symbols, outperforms the synch-based CFO algorithm even when only two consecutive CPs are averaged. Additionally, we see that the performance of the CP-based estimation improves as the number of averaged CPs increases, however, the downside of averaging more CPs is an increase in the processing time.

Figure 9 presents the comparison of the packet reception rate (PRR) for each one of the signals carried by a SCATTER PHY subframe, namely, synchronization, control and data signals over several SNR and BW values. The MCS used for modulating the user data is set to 0, which is the most robust coding scheme, allowing SCATTER PHY to decode data in low SNR scenarios. The purpose of this experiment is to identify the lowest possible SNR at which SCATTER PHY can still correctly decode the user data. This experiment is run by adding a channel emulator between the Tx and Rx sides of a single PHY instance. At the Tx side, the generated subframes, instead of being sent to the USRP

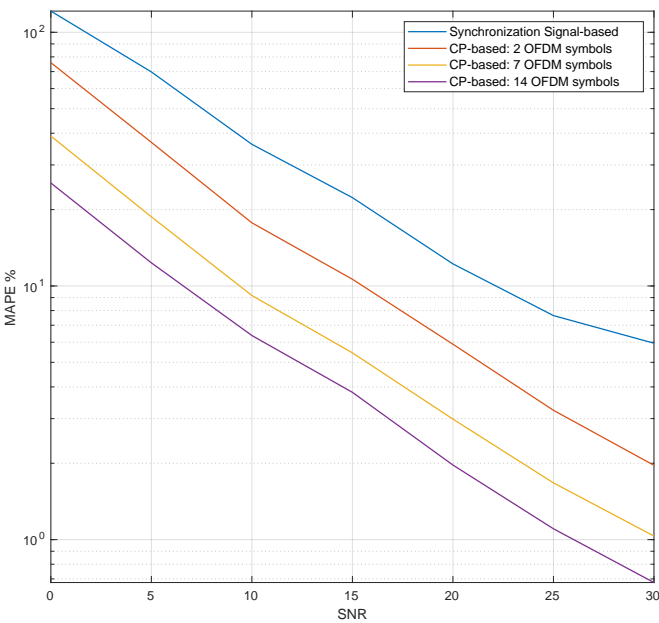


Figure 8. Comparison of CFO estimation error (MAPE) in AWGN channel.

328 HW are sent to an abstraction layer that emulates the HW and adds AWGN noise to the transmitted
329 signal, next, the abstraction layer transfers the noisy signal to the receiving side of the PHY. The PRR is
330 averaged over 10^4 trials, where at each trial, the Tx side of the PHY sends a single synchronization
331 subframe. As can be seen, synchronization and control signals have a better PRR performance than
332 that of the data decoding, however, SCATTER PHY's PRR performance is limited by the ability of
333 correctly decoding the data section of a subframe. Additionally, we see that the Data PRR is better for
334 the 1.26 MHz case, which is due to the fact that compared to the other BW values, MCS 0 for the 1.26
335 MHz case, carries more redundancy bits, as shown by Table 2, making it more robust against noise.
336 The other two signals, synchronization, and control present similar PRR curves for all BW values.
337 Moreover, it is noticeable that the data PRR is equal to 1 for SNR values greater than or equal to 0 dB.

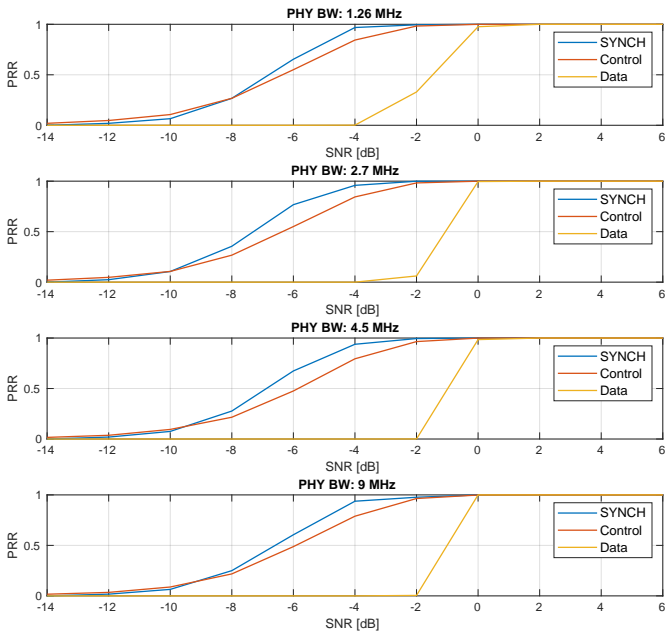


Figure 9. Comparison of the PRR for the different signals carried by a SCATTER PHY subframe.

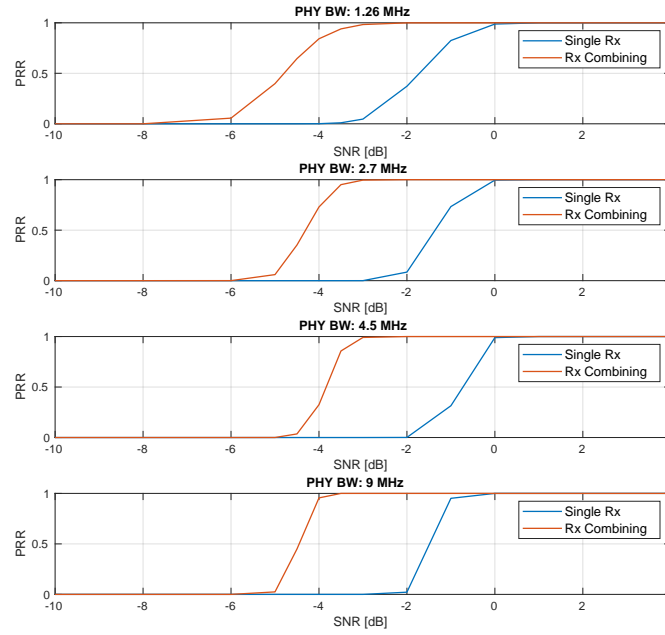


Figure 10. Simple Rx combining with SCATTER PHY for several PHY BW values.

Figure 10 presents the result of the comparison of single Rx data decoding and that of a simple Rx combining scheme that can be implemented with SCATTER PHY. The figure shows data decoding PRR results for both receiving schemes. Here in this case, after synchronization (synch signal detection, CFO estimation/correction, control data decoding and subframe alignment) the two independently synchronized subframes are combined through a simple average of both subframes as defined in (6).

$$r_{\text{comb.}}(n) = \frac{1}{N_{\text{ant.}}} \sum_{i=1}^{N_{\text{ant.}}} r_{\text{Rx}_i}(n), \quad (6)$$

where $N_{\text{ant.}}$ is the number of Rx antennas (in our case $N_{\text{ant.}} = 2$), $r_{\text{Rx}_i}(n)$ is the subframe synchronized at i th antenna, and $r_{\text{comb.}}(n)$ is the resulting combined subframe signal. The MCS value used in

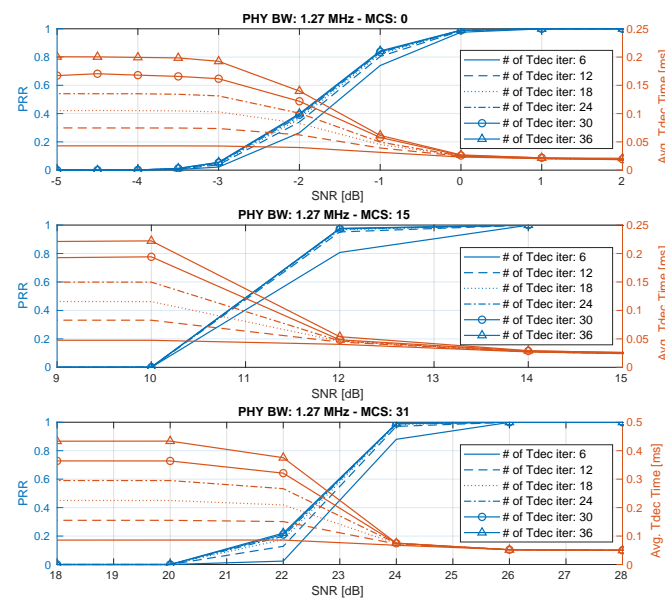


Figure 11. Comparison of the variation of the maximum number of turbo decoding iterations for a PHY BW of 1.27 MHz and 3 different MCS values.

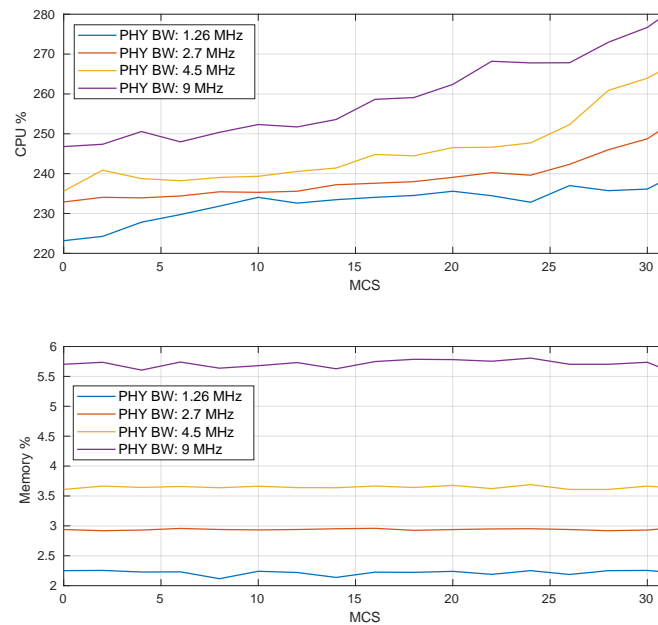


Figure 12. CPU and memory utilization of the SCATTER PHY.

this experiment for all PHY BWs is 0. As can be seen, the Rx combining scheme employed in this experiment provides a processing gain ranging from 2.5 to 3 [dB] over the single Rx approach. With this scheme, SCATTER PHY is able to combine the received subframes for improved performance in low SNR scenarios.

Figure 11 presents the results on the variation of the maximum number of turbo decoding iterations for a PHY BW of 1.27 MHz and 3 different MCS values, 0 (QPSK), 15 (16QAM) and 31 (64QAM). As can be seen, the PRR improves as the number of maximum iterations also increases. As can be also noticed, the PRR improvement is higher for higher MCS values, 15 and 31. This is due to the fact that as the MCS increases, the code rate increases, and consequently, the number of redundancy bits decreases, making the transmit data more prone to errors. Therefore, as the number of maximum iterations increases, the probability of decoding also increases. Another important point is the trade-off between PRR improvement and the increase in decoding time. As can be verified, the PRR improves at the cost of longer decoding times for low to medium SNR values.

Figure 12 depicts the CPU and memory utilization of the SCATTER PHY, operating at 1.27, 2.7, 4.5 and 9 MHz bandwidths for several MCS values. The results in the figure were calculated by averaging CPU and memory usage values sampled every 200 ms during the duration of the experiment. In this experiment, we have the SCATTER PHY operating in full-duplex mode, where one PHY transmits to the other and vice versa. Each one of the PHYs transmits 20 subframes in a row with a 1 ms gap between consecutive transmissions.

As can be seen, both CPU and memory utilization increase as the BW increases, however, there is no CPU or memory starvation issues with any of the selected BW and MCS values. As can be also noticed, for a given BW, the CPU usage increases as the MCS increases. This is due to the fact that as the MCS value increases (*i.e.*, higher data rates), the turbo encoding (at Tx side), synchronization and turbo decoding (both at Rx side) processing tasks become more complex and demand a lot more of CPU for data processing. For a BW of 9 MHz and MCS equal to 31, the SCATTER PHY CPU utilization is of around 280 %, meaning that the processing power of fewer than 3 cores is being used, leaving the other cores in the idle state for large periods.

For the memory utilization case, it can be seen that for a given BW the memory usage is practically constant for all MCS values and therefore, practically independent of the MCS value. This is an expected result as all the memory used by the SCATTER PHY is pre-allocated during its initialization.

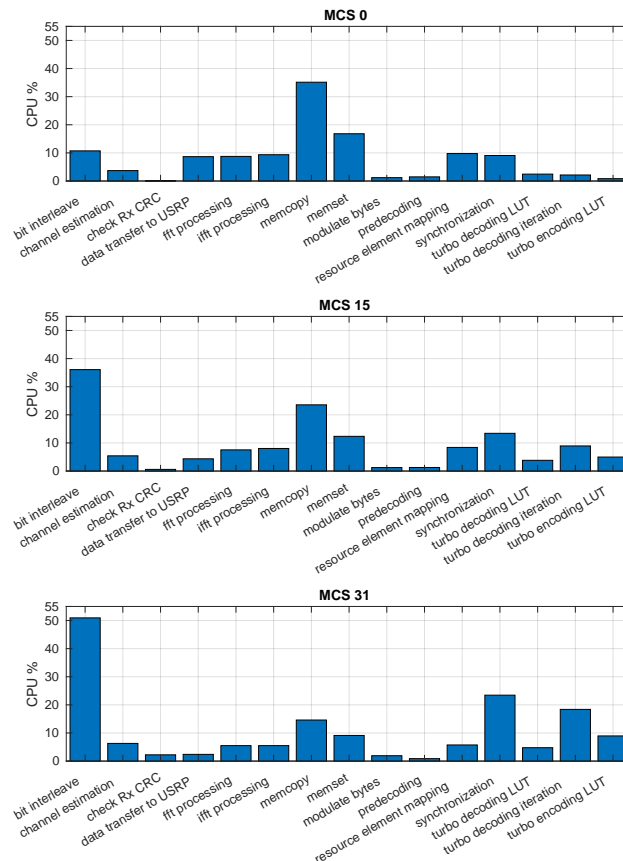


Figure 13. CPU profiling of individual components of the SCATTER PHY for a BW of 4.5 MHz.

Therefore, based on the results presented in Figure 12, it can be concluded that the SCATTER PHY does not exhaust CPU or memory resources as BW or MCS values increase.

In Figure 13, we show the CPU consumption of the main functions making up the SCATTER PHY for a BW of 4.5 MHz and three different MCS values. The CPU consumption assessment uses *valgrind* with its *callgrind* tool [24]. The setup for this experiment is the same as the one used earlier for the CPU and memory profiling. The figure shows Tx and Rx functions that use more CPU processing. By analyzing the figure, we see that the bit interleaving processing uses more CPU as MCS increases, consuming approximately 51 % of CPU time for MCS equal to 31 compared to approximately 11 % when MCS is equal to 0. The synchronization task consumes a lot of CPU time for all the MCS values considered in this experiment, ranging from 9 % to more than 23 % of CPU time. It can be also noticed that the turbo decoding iteration task consumes more CPU time as the MCS increases, going from 2 % (MCS 0) to more than 18 % (MCS 31). Additionally, we see that memory copy and setting functions consume less CPU time as the MCS increases. This is due to the fact that other functions start consuming more CPU when the MCS increases.

5. Conclusion

The paper has described the design, features, and benefits of the SCATTER PHY, which is an open-source physical layer that uses a filtered form of OFDM as its waveform.

The main features include using Google's Protobuf to define the message format for interaction with higher layers of the system, the ability to transmit/receive two different types of packets, which can include a synchronization signal, control data, and user data. The PHY layer communication operates with various modulation and coding schemes, several different channel bandwidths, and automatically improves the carrier frequency offset correction by utilizing the OFDM CP in addition to the synchronization subframe detected in the signal. SCATTER PHY minimizes OOB by utilizing

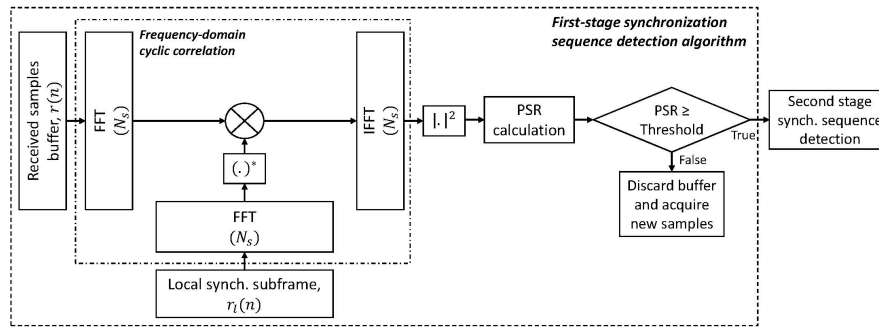


Figure A1. Summary of the first-stage synchronization sequence detection algorithm.

custom-built FPGA-based FIR filters on the transmitter side, configured for each channel bandwidth at run-time time. Additionally, two independent physical interfaces are available, which effectively enable the usage of full-duplex and theoretically double the possible throughput achieved with one interface.

The described benefits are verified experimentally. The throughput achieved with actual physical transmissions is compared to a theoretical maximum, defined by using the entire available time for communication with an effective duty cycle of 100%. The verification shows that using full-duplex makes no impact on the achieved throughput in each interface. Spectrometer measurement results showcase the capabilities of the FPGA-based FIR filter, resulting in negligible OOB, which in turn, increases channel utilization and facilitates communication in neighboring channels by other radios. Using CP-based CFO estimation and correction is demonstrated to outperform the sync-signal based correction in all cases, at the expense of higher processing time. Finally, the packet reception rate for all types of transmitted signals is presented, showing that the SCATTER PHY can successfully decode user data at SNR values as low as 0 dB.

Appendix A Two-stage Detection Algorithm

The very first processing executed by SCATTER PHY is the detection of incoming synchronization subframes, which might be followed by data-only subframes. This is a task of utmost importance once the correct detection of subframes is crucial in establishing links in scenarios with low SNR or limited by interference. In this appendix, we describe the proposed detection algorithm employed in the SCATTER PHY.

For improved performance, the proposed algorithm is split into two stages. In the first stage, a buffer containing received samples with the length of a subframe is correlated with a local version of the synch subframe, and then, if the Peak to Side-lobe ratio (PSR) is greater than or equal to a predefined threshold, $\text{Thresh}_{1\text{st_stage}}$, then the detection algorithm proceeds to the second stage. The power delay profile (PDP) between the received signal and the local version of the sync subframe containing only the synchronization signal is defined as

$$\text{PDP}_{1\text{st_stage}}(l) = \left| \sum_{n=0}^{N_s-1} r[n] r_l^*[(n+l)_{N_s}] \right|^2, \quad 0 \leq l \leq N_s - 1, \quad (\text{A1})$$

where $r(n)$ is the received signal, $r_l(n)$ is the local version synch subframe, N_s is the length of a subframe, $(\cdot)_{N_s}$ denotes that this is a cyclic correlation, and $(\cdot)^*$ is the complex conjugate. Therefore, the detection algorithm proceeds to the second stage if the PSR of the first stage is greater than or equal to the first stage threshold, as defined in (A2). Figure A1 summarizes the processing carried out in the first stage of the synchronization sequence detection algorithm. As it is seen in the figure, we use frequency-domain cross-correlation as its implementation with FFTs and an IFFT is faster than the time-domain version of the correlation [25].

$$\text{PSR}_{1\text{st_stage}} = \frac{\arg \max_l \text{PDP}_{1\text{st_stage}}(l)}{\arg \max_{i, i \neq l} \text{PDP}_{1\text{st_stage}}(i)} \geq \text{Thresh}_{1\text{st_stage}}. \quad (\text{A2})$$

In the second stage, if $\text{PSR}_{1\text{st_stage}} \geq \text{Thresh}_{1\text{st_stage}}$, then the l th lag that maximized the numerator in (A2) possibly points to the middle of the synchronization subframe. Based on the position of the l th

lag, it is possible to extract (*i.e.*, convert the synch OFDM symbol into frequency domain and extract the 72 complex symbols carrying the synch sequence) the 72 symbol long synchronization sequence and correlate it with its local version as defined next. The PDP between the received synch sequence and its local version is defined as

$$\text{PDP}_{2\text{nd_stage}}(m) = \left| \sum_{n=0}^{N_{\text{synch}}-1} S[n] S_l^*[(n+m)_{N_{\text{synch}}}] \right|^2, \quad (A3)$$

$$0 \leq m \leq N_{\text{synch}} - 1,$$

where $S(n)$ is the received synch sequence, $S_l(n)$ is the local version of the synch sequence, N_{synch} is the length of the synch sequence, $(\cdot)_{N_{\text{synch}}}$ denotes that this is a cyclic correlation, and $(\cdot)^*$ is the complex conjugate. The signal processing carried out at second stage is based on $\text{PDP}_{2\text{nd_stage}}(m)$. For the sake of brevity we use $\text{PDP}(m)$ to denote $\text{PDP}_{2\text{nd_stage}}(m)$ in what follows.

The second stage of the proposed detection algorithm is based on the CA-CFAR algorithm [16]. It is employed to detect the presence of the synchronization sequence within a given received signal buffer. The algorithm is used to calculate a variable detection threshold, which varies according to the noise power, based on the second stage PDP set of samples, $\text{PDP}_{2\text{nd_stage}}(m)$. The algorithm is split into three steps: (i) removal of values that are not considered as only being noise observations, also known as the censoring step, (ii) calculation of a noise-power threshold, and (iii) the actual detection of the synchronization sequence. These steps make it possible to reliably decide whether the synchronization sequence is present or not in a given received buffer. Figure A2 summarizes the three steps composing the synch sequence detection algorithm employed in the second stage. Next, we describe each one of the three steps composing the second stage synch detection algorithm.

The procedure followed in the second-stage of the synchronization sequence detection algorithm consists basically of hypothesis testing following the Neyman&Pearson lemma [26]. This lemma establishes that the synchronization sequence detection testing is based on the following likelihood ratio inequality

$$\frac{\text{PDP}_{H_1}}{\text{PDP}_{H_0}} > \alpha, \quad (A4)$$

where the hypothesis H_0 is rejected in favor of the hypothesis H_1 when the PDP set contains desired signal, *i.e.*, the synchronization sequence. Therefore, the hypothesis, H_0 is the one when only noise is present. This hypothesis testing is optimum only when the cumulative distribution function (CDF) of the ratio in (A4) given the hypothesis H_0 is known, so that it is possible to calculate a threshold value that satisfies:

$$P \left\{ \frac{\text{PDP}_{H_1}}{\text{PDP}_{H_0}} > \alpha \mid H_0 \right\} = P_{FR}, \quad (A5)$$

for a given probability of false rejection, P_{FR} . Therefore, the derivation of the detection threshold assumes knowledge of the probability distribution function (PDF) of both random variables PDP_{H_1} and PDP_{H_0} .

Appendix A.1 Censoring and Noise Power Reference Calculation Step

In this step, corrupted PDP samples are removed from the reference set, *i.e.*, the PDP set. Corrupted samples are caused by the presence of samples containing noise plus a signal (*i.e.*, a signal different from noise). For the censoring of PDP samples, we employ the Forward Consecutive Mean Excision (FCME) algorithm [27]. Initially, the values in the PDP set are defined as

$$\left\{ \text{PDP}(m) \mid m = 0, 1, \dots, N_{\text{synch}} - 1 \right\}, \quad (A6)$$

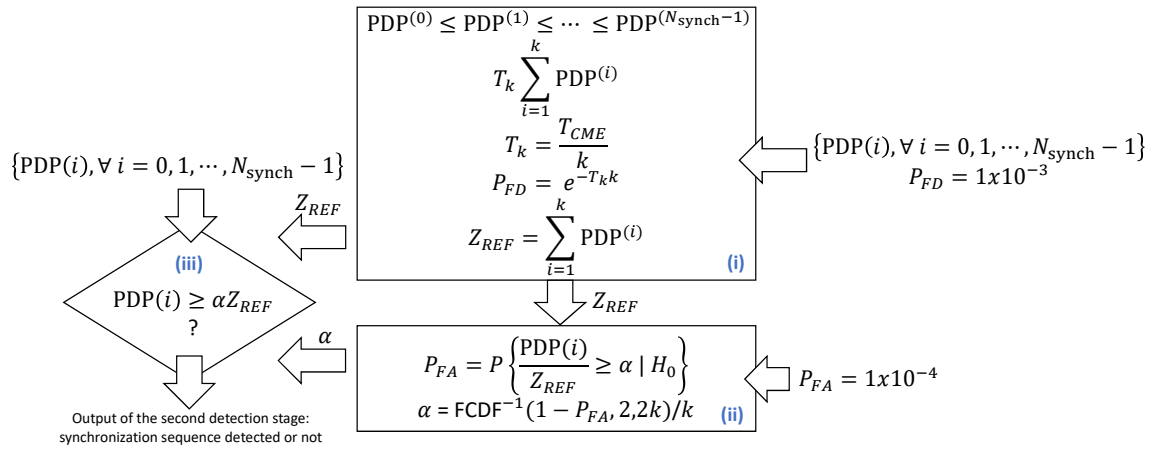


Figure A2. Summary of the second-stage synch sequence detection algorithm: step (i) is used to calculate a reference for the noise-power, Z_{REF} ; step (ii) is used to calculate the detection threshold, α ; and step (iii) applies a hypothesis test to all the elements of the second stage PDP.

and then are sorted in ascending order, so that

$$\{PDP^{(i)} \mid i = 0, 1, \dots, N_{\text{synch}} - 1\}, \quad (\text{A7})$$

where

$$PDP^{(0)} \leq PDP^{(1)} \leq PDP^{(2)} \leq \dots \leq PDP^{(N_{\text{synch}}-1)}. \quad (\text{A8})$$

The censoring at the k -th iteration is performed with the following inequality

$$PDP^{(k+1)} \geq T_k \sum_{i=1}^k PDP^{(i)}, \quad (\text{A9})$$

where T_k is the censoring scaling factor at the k -th iteration. If the inequality in (A9) is satisfied, all the values greater than $PDP^{(k+1)}$ are decided to be corrupted. The test of (A9) is performed iteratively, being necessary to calculate the censoring scaling factor, T_k , for each new iteration. The censoring scaling factor, T_k , controls the properties of the censoring step, and it is normally chosen so that the probability of false disposal, P_{FD} , has a pre-defined value. P_{FD} is the probability of making an incorrect decision during the k -th iteration of the censoring step. The initial T_k value is calculated assuming that $PDP^{(k+1)}$ is a sample that only contains noise observation, *i.e.*, a sample that is free of the presence of the synchronization sequence. Under this assumption, the probability that the inequality in (A9) is true corresponds to a probability of false disposal, P_{FD} , which is defined by

$$P_{FD} = P \left\{ PDP^{(k+1)} \geq T_k \sum_{i=1}^k PDP^{(i)} \mid H_0 \right\}. \quad (\text{A10})$$

Each iteration of the censoring step starts with k equal to the smallest assumed clean set of PDP samples. The larger the smallest assumed clean set of PDP samples, the better the censoring algorithm works [27]. However, the larger the assumed clean set of PDP samples, the higher is the probability that corrupted PDP samples will be part of the initial clean set. The testing of (A9) continues until it is true for some value of k or all reference PDP samples are decided to signal-free.

Considering the H_0 hypothesis (*i.e.*, observation of samples only containing noise) and that noise follows a complex Gaussian distribution, $\mathcal{CN}(0, \sigma^2)$. Therefore, each one of the PDP values follows a Gamma distribution, $\Gamma(1, N_{\text{synch}}\sigma^2)$ with mean given by

$$\mathbb{E}[PDP^{(\cdot)}] = N_{\text{synch}}\sigma^2. \quad (\text{A11})$$

Therefore, since that

$$\frac{\sum_{i=0}^{k-1} \text{PDP}^{(i)}}{k} \approx \mathbb{E}[\text{PDP}^{(\cdot)}] = N_{\text{synch}} \sigma^2, \quad (\text{A12})$$

the equation for the probability of false disposal, P_{FD} , can be approximated as

$$P_{FD} \approx P \left\{ \text{PDP}^{(k+1)} \geq T_k k N_{\text{synch}} \sigma^2 \right\}. \quad (\text{A13})$$

This approximation becomes better as the number of reference PDP samples, k , increases. Additionally, given that $\text{PDP}^{(k+1)}$ is a Gamma random variable, $\Gamma(k, \theta)$, with shape parameter being a positive integer number, 1, its cumulative distribution function (CDF) can be simplified and expressed in closed form as [28,29]

$$F(y, k, \theta) = P(Y \leq y) = 1 - e^{-y/\theta} \sum_{i=0}^{k-1} \frac{1}{i!} \left(\frac{y}{\theta} \right)^i, \quad (\text{A14})$$

where k and θ are the shape and scale parameters respectively.

Therefore, the probability of false discard (or censoring) in (A13) can be calculated as

$$P_{FD} = 1 - P \left\{ \text{PDP}^{(k+1)} \leq T_k k N_{\text{synch}} \sigma^2 \right\} = e^{-T_k k}, \quad (\text{A15})$$

where in (A14), y was made equal to $T_k k N_{\text{synch}} \sigma^2$. Here we define $T_{CME} = k T_k$, which represents a input parameter for the FCME algorithm and can be found by solving (A15). The probability of false disposal, P_{FD} , can be understood as the desired clean sample rejection rate.

The censoring threshold is defined as $T_k \sum_{i=1}^k \text{PDP}^{(i)}$ and therefore, PDP samples that have value greater or equal than the threshold are discarded and then, the new set of PDP samples consists of the remaining ones, which are assumed to contain only noise observations. The output of the censoring step is composed by the number of noise-only PDP samples, k , and the *noise-power reference*, Z_{REF} , which is defined by the summation of all PDP samples that have value below the censoring threshold. Z_{REF} is defined as

$$Z_{\text{REF}} = \sum_{i=1}^k \text{PDP}^{(i)}. \quad (\text{A16})$$

Appendix A.2 Detection Threshold Calculation Step

After selecting the set of PDP samples that are considered as only containing noise, the next stage consists of calculating the synchronization detection threshold, α . The detection threshold, α , is calculated according to the decision method known as the Cell Averaging (CA) method [27].

The detection threshold, α , is calculated under the assumption that there is no synchronization sequence present in the reference PDP set of samples for a given probability of false alarm, P_{FA} , which is defined as

$$P_{FA} = P \left\{ \frac{\text{PDP}^{(i)}}{Z_{\text{REF}}} \geq \alpha \mid H_0 \right\}. \quad (\text{A17})$$

We assume that the PDP samples and the the noise-power reference, Z_{REF} , follow scaled central Chi-squared distributions, $\frac{N_{\text{synch}} \sigma^2}{2} \chi^2(2)$, $\frac{N_{\text{synch}} \sigma^2}{2} \chi^2(2k)$ with 2 and $2k$ degrees of freedom, respectively. Additionally, it is known that the ratio between two central Chi-squared random variables results in a random variable that follows a *Fisher* distribution [27,30], whose CDF is defined as

$$\text{FCDF}(\alpha) = 1 - P \left\{ \frac{\text{PDP}^{(i)}/2}{Z_{\text{REF}}/2k} \geq k\alpha \right\}, \quad (\text{A18})$$

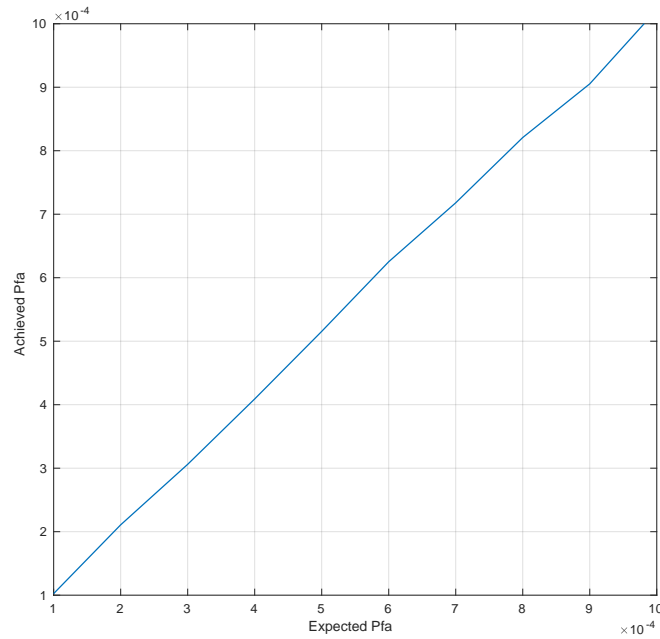


Figure A3. Comparison between the expected P_{FA} and the actual value for a SNR of 0 [dB].

cropped_pfa_vs_achieved_pd.pdf

Figure A4. Comparison between P_{FA} and the actual P_{CD} value for a SNR of 0 [dB].

therefore, the detection threshold, α , is calculated as

$$\alpha = \text{FCDF}^{-1}(1 - P_{FA}, 2, 2k) / k. \quad (\text{A19})$$

where FCDF is the *Fisher* CDF. Note that the scaling factor, $\frac{N_{\text{synch}}\sigma^2}{2}$, is the same for $\text{PDP}(\cdot)$ and Z_{REF} , and therefore, it disappears when the ratio is taken.

Appendix A.3 Synchronization Sequence Detection Step

After calculating the detection threshold, α , and the noise-power reference, Z_{REF} , the final decision if the synchronization sequence is present or not is made by evaluating the following hypothesis test

$$\text{PDP}(i) \geq \alpha Z_{\text{REF}}, \forall i = 0, 1, \dots, N_{\text{synch}} - 1. \quad (\text{A20})$$

If the test is true, the synchronization sequence plus noise hypothesis, H_1 , is chosen. Otherwise, the noise-only hypothesis, H_0 , is decided to be true and the received signal buffer is declared not having the presence of the synchronization sequence.

Appendix A.4 Results of Synchronization Sequence Detection Algorithm

Figure A3 presents a comparison between the expected P_{FA} and the actual P_{FA} value. The expected P_{FA} is varied from 10^{-5} to 10^{-4} . The number of Monte Carlo trials is set to 10^6 iterations and an AWGN channel is assumed. As can be seen, the actual P_{FA} value stays very close to the expected one.

Figure A4 depicts P_{FA} versus the probability of correct detection, P_{CD} . The P_{FA} is varied from 10^{-5} to 10^{-4} , the SNR is set to 0 [dB] and the number of Monte Carlo trials is equal to 10^5 iterations. The figure shows that the presence of a synchronization sequence is always detected independently of the P_{FA} value set.

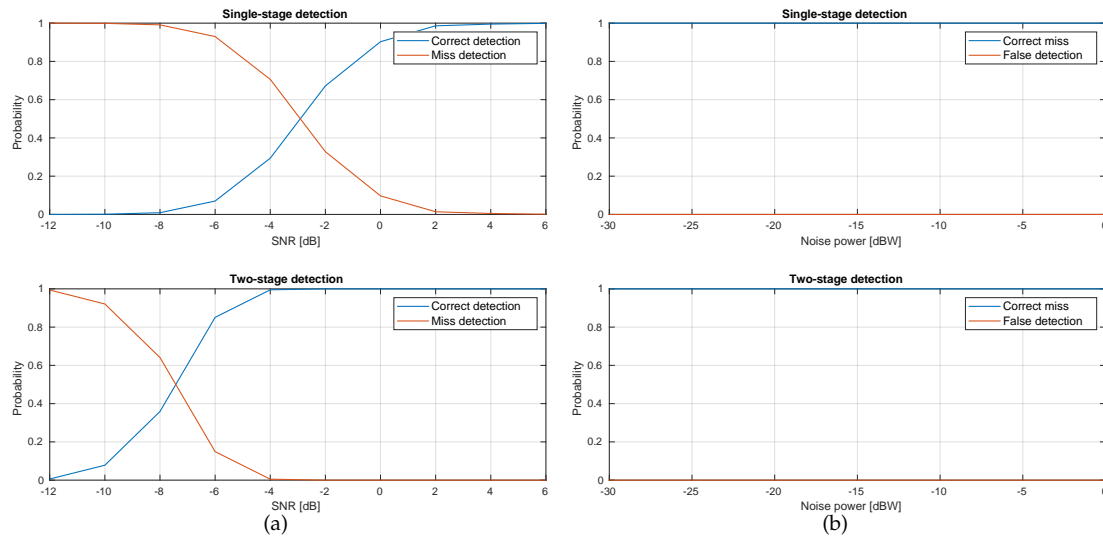


Figure A5. Comparison between the single and two-stage synchronization sequence detection schemes: (a) correct and miss-detection probabilities (b) correct miss and false detection probabilities.

Next, we present some results comparing the detection performance of the proposed two-stage detection algorithm against the single-stage one. The number of Monte Carlo trials here is set to 10^5 iterations and the AWGN channel is used. For both results shown in the Figure A5, the probability of false alarm P_{FA} is set to 10^{-4} and the probability of false disposal P_{FD} is set to 10^{-3} . Figure 5(a) shows the correct and miss detection probabilities for the SNR varying from -12 [dB] up to 0 [dB]. As it can be noticed, the proposed two-stage detection scheme presents a probability of correct detection of 1 for SNR values greater than -4 [dB], *i.e.*, the presence of the synchronization sequence is always detected, while the single scheme only achieves a probability of 1 for SNR values greater than 2 [dB]. Figure 5(b) shows the correct miss and false detection probabilities when the noise power varies from -30 [dBW] up to 0 [dBW]. As can be seen, both schemes present correct miss probability equal to 1 and false detection probability equal to 0 for all noise power values considered in this experiment. It is important to be mentioned that the combined working of the two detection stages decreases the probability of false detection (or also known as a false alarm) of the two-stages algorithm when compared to the probability of false detection of the second stage alone (see Figure A3).

Appendix B CFO Estimation

Appendix B.1 Integer Estimation

After synch signal detection and subframe alignment, the integer (*i.e.*, coarse) CFO estimation is carried out through by correlating the received OFDM symbol carrying the synch signal with several local versions of it with different integer frequency offset applied to each of them. The PDP between the received and a given offset local version of the synch OFDM symbol is defined as

$$\text{PDP}(l, k_{\text{IFO}}) = \left| \sum_{n=0}^{N_{\text{FFT}}-1} y[n] x_l^*[(n+l)_{N_{\text{FFT}}}] e^{\frac{j2\pi k_{\text{IFO}} n}{N_{\text{FFT}}}} \right|^2, \quad (A21)$$

$$0 \leq l \leq N_{\text{FFT}} - 1$$

$$-36 \leq k_{\text{IFO}} \leq 36,$$

where $y(n)$ is the received synch OFDM symbol signal, $x_l(n)$ is the offset local version of the synch symbol, N_{FFT} is the FFT length, $(\cdot)_{N_{\text{FFT}}}$ denotes that this is a cyclic correlation, k_{IFO} is the integer

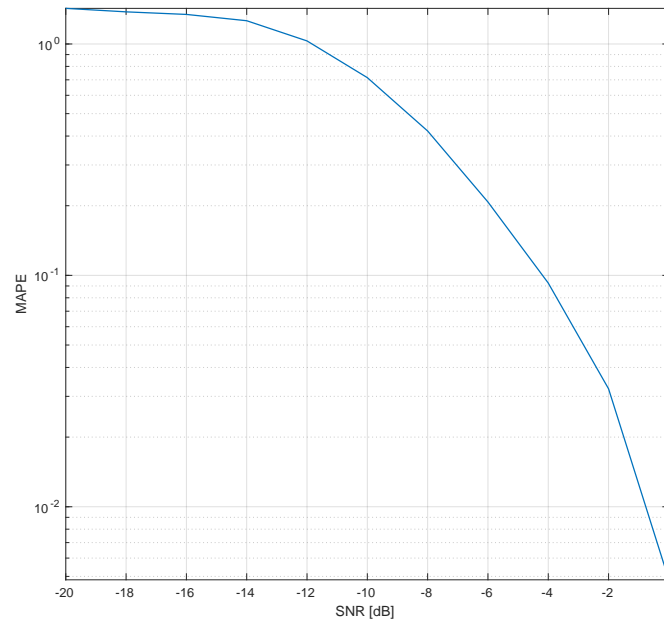


Figure A6. Performance of the integer CFO estimation algorithm.

frequency offset, and $(\cdot)^*$ is the complex conjugate. Therefore, the integer frequency offset can be estimated by solving the following maximization problem

$$\hat{\epsilon}_{k_{\text{IFO}}} = \arg \max_{k_{\text{IFO}}} \left(\arg \max_l \text{PDP}(l, k_{\text{IFO}}) \right). \quad (\text{A22})$$

Figure A6 shows the performance of the integer CFO estimation algorithm described above for an AWGN channel over several SNR values and a PHY BW of 4.5 MHz.

Appendix B.2 Fractional Estimation

As shown by the results in Figure 8, the CP-based fractional CFO estimation outperforms the Synch-based fractional CFO one and therefore is the method adopted by the SCATTER PHY. Next we describe the CP-based fractional CFO estimation.

$$\psi(k) = \sum_{n=0}^{N_{\text{CP}}-1} \frac{r_d^*(n + k(N_{\text{FFT}} + N_{\text{CP}})) r_d(n + k(N_{\text{FFT}} + N_{\text{CP}}) + N_{\text{FFT}})}{2\pi}, \quad (\text{A23})$$

$$\hat{\epsilon}_{\text{FFO}} = \frac{1}{N_{\text{symb}}} \sum_{k=0}^{N_{\text{symb}}-1} \arg(\psi(k)), \quad (\text{A24})$$

where $r_d(n)$ is the detected, integer CFO corrected, and aligned subframe, N_{symb} is the number of OFDM symbols used to average the CFO estimation, and N_{CP} is the length of the OFDM cyclic prefix. As can be seen, the employed CP-based fractional CFO estimation is obtained by averaging the estimated fractional CFO over the selected number of OFDM symbols.

Author Contributions: conceptualization, methodology, software, validation, investigation, formal analysis, data curation, and writing—original draft preparation Felipe A. P. de Figueiredo, Dragoslav Stojadinovic, and Prasanthi Maddala, writing—review and editing Ruben Mennes, and Irfan Jabandžić, resources, supervision, project administration, and funding acquisition Xianjun Jiao, and Ingrid Moerman.

Acknowledgments: This work was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732174 (ORCA project).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. DARPA SC2, *DARPA Spectrum Collaboration Challenge*, [Online]. Available: <https://spectrumcollaborationchallenge.com/>, Accessed on September 2019.
2. Ruben Mennes, Maxim Claeys, Felipe A. P. De Figueiredo, Irfan Jabandzic, Ingrid Moerman, and Steven Latre, *Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments*, IEEE Access, vol. 7, no. 2019, pp. 45818-45830, April 2019.
3. Felipe A. P. de Figueiredo et. al., *SCATTER PHY: An Open Source Physical Layer for the DARPA Spectrum Collaboration Challenge*. [Online]. Available: <https://github.com/zz4fap/scatter-phy>, September 2019.
4. Yiding Yu, Taotao Wang, and Soung Chang Liew, *Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks*, IEEE International Conference on Communications (ICC), May 2018.
5. Ruben Mennes, Miguel Camelo, Maxim Claeys and Steven Latre, *A Neural-Network-based MF-TDMA MAC Scheduler for Collaborative Wireless Networks*, IEEE Wireless Communications and Networking Conference (WCNC), April 2018.
6. Miguel Camelo, Adnan Shahid, Jaron Fontaine, Felipe A. P. De Figueiredo, Eli De Poorter, Ingrid Moerman, and Steven Latre, *A Semi-Supervised Learning Approach Towards Automatic Wireless Technology Recognition*, IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 2019.
7. Felipe A. P. de Figueiredo, R. Mennes, X. Jiao, W. Liu, and I. Moerman, *A spectrum sharing framework for intelligent next-generation wireless networks*, IEEE Access, vol. 6, pp. 60704-60735, November 2018.
8. Ettus Research LLC, *USRP Hardware Driver*, [online]. Available: <http://files.ettus.com/manual/>
9. Ettus Research LLC, *USRP X Series*, [online]. Available: <https://www.ettus.com/product/category/USRP-X-Series>
10. National Instruments, *Overview of the NI USRP RIO Software-Defined Radio*, White Paper, December 2015. [online]. Available: <http://www.ni.com/white-paper/52119/en/>
11. Ettus Research LLC, *Products*, [online]. Available: <https://www.ettus.com/product>
12. P. Hintjens, *ZeroMQ Messaging for Many Applications*, O'Reilly Media, March 2013.
13. Google, *protobuf - Protocol Buffers - Google's data interchange format*, [online]. Available: <http://code.google.com/p/protobuf/>, 2011.
14. I. Gomez-Migueluez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, *srsLTE: An Open-Source Platform for LTE Evolution and Experimentation*, ACM WiNTECH Workshop, October 2016.
15. Sravanthi Kanchi, Shubhrika Sandilya, Deesha Bhosale, Adwait Pitkar, and Mayur Gondhalekar, *Overview of LTE-A technology*, IEEE Global High Tech Congress on Electronics (GHTCE), March 2014.
16. Felipe A. P. de Figueiredo, Fabbryccio A. C. M. Cardoso, Jose A. Bianco F., Rafael M. Vilela, Karlo G. Lenzi, *Multi-stage Based Cross-Correlation Peak Detection for LTE Random Access Preambles*, Revista Telecomunicacoes, vol. 15, September 2013.
17. David C. Chu, *Polyphase codes with good periodic correlation properties*, IEEE Transactions on Information Theory, vol. 18, no. 4, pp. 531-532, July 1972.
18. D. V. Sarwate and M. B. Pursley, *Crosscorrelation properties of pseudorandom and related sequences*, Proc. IEEE, vol. 68, no. 5, pp. 593-619, May 1980.
19. Feng Wang, and Yu Zhu, *An efficient CFO estimation algorithm for the downlink of 3GPP-LTE*, International Conference on Wireless Communications and Signal Processing (WCSP), November 2011.
20. Quentin Bodinier, Faouzi Bader, and Jacques Palicot, *On Spectral Coexistence of CP-OFDM and FB-MC Waveforms in 5G Networks*, IEEE Access, vol. 5, July 2017.
21. Hao Lin, and Pierre Siohan, *Major 5G Waveform Candidates: Overview and Comparison*, Book chapter in Signal Processing for 5G, Wiley, August 2016.
22. Felipe A. P. de Figueiredo et. al., *A Spectrum Sharing Framework for Intelligent Next-Generation Wireless Networks*, IEEE Access, vol. 6, pp. 60704-60735, December 2018.
23. Ettus Research LLC, *CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series only)*, [online]. Available: <https://www.ettus.com/product/details/CBX120>
24. Josef Weidendorfer, Markus Kowarschik, and Carsten Trinitis, *A Tool Suite for Simulation Based Analysis of Memory Access Behavior*, Proceedings of the 4th International Conference on Computational Science (ICCS), Krakow, Poland, June 2004.
25. Stefania Sesia, Issam Toufik, and Matthew Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*, John Wiley & Sons, 2011.

- 540 26. J. Neyman and E. S. Pearson, *On the problem of the most efficient tests of statistical hypotheses*, Philosophical
541 Transactions of the Royal Society of London, Series A. 231: 289-337, 1933.
- 542 27. Janne J. Lehtomaki, Markku Juntti, and Harri Saarnisaari, *CFAR Strategies for Channelized Radiometer*, IEEE
543 Signal Processing Letters, vol. 12, no. 1, January, 2005.
- 544 28. J. G. Proakis, *Digital Communications*, 5th Edition, McGraw-Hill, New York, 2001.
- 545 29. Papoulis, Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th Edition, McGraw-Hill Europe,
546 January 2002.
- 547 30. M. H. DeGroot, *Probability and Statistics*, 2nd Edition, Addison-Wesley, 1986.
- 548 **Sample Availability:** Samples of the compounds are available from the authors.