A Base-Band Wireless Spectrum Hypervisor for Multiplexing Concurrent OFDM signals

Felipe A. P. de Figueiredo*, Ruben Mennes‡, Irfan Jabandžić*, Xianjun Jiao*, and Ingrid Moerman*
*Ghent University - imec, IDLab, Department of Information Technology, Ghent, Belgium.
‡Department of Computer Science, University of Antwerp - imec, Antwerp, Belgium.
Email: *[felipe.pereira, irfan.jabandzic, xianjun.jiao, ingrid.moerman]@ugent.be, ‡ruben.mennes@uantwerpen.be

***Abstract*—The next generation of wireless and mobile networks will have to handle a significant increase in traffic load compared to the actual one. This situation calls for novel ways to increase the spectral efficiency. Therefore in this paper, we propose a wireless spectrum hypervisor architecture that abstracts a radio frequency (RF) front-end into a configurable number of virtual RF front-ends. The proposed architecture has the ability to enable flexible spectrum access in existing wireless and mobile networks, which is a challenging task due to the limited spectrum programmability, $i.e.$, the capability a system has to change the spectral properties of a given signal to fit an arbitrary frequency allocation. The main goal of the proposed approach is to improve spectral efficiency by efficiently using vacant gaps in congested spectrum-bandwidths or adopting network densification through infrastructure sharing. We demonstrate mathematically how our proposed approach works and present several simulation results proving its functionality and efficiency. Additionally, we designed and implemented an open-source and free proof of concept prototype of the proposed architecture, which can be used by researchers and developers to run experiments or extend the concept to other applications. We present several experimental results used to validate the proposed prototype. We demonstrate that the prototype can easily handle up to 12 concurrent physical layers.**

***Index Terms*—radio virtualization, software-defined radio, network densification, infrastructure sharing, multi-tenancy, cognitive radios.**

I. Introduction

According to [1], the average mobile connection speed in 2016 was 6.8 Mbps and it is forecast to grow at a compound annual growth rate (CAGR) of 24.4 percent, reaching nearly 20.4 Mbps by 2021. By 2020 it is forecast that there will be 11.6 billion mobile-connected devices, including Machine to Machine (M2M) modules [1]. Reports like [2] state that future mobile networks will need to support 1 million connections per square kilometer and up to a total of 100 billion connections in total.

In order to support the expected growth in capacity, the forecast for the future mobile networks, the area throughput, $i.e.$, bit/s/km^2, needs to be increased. Some ways to improve area throughput in cellular networks include densification of existing cellular networks using extra-small cells, provision of peer-to-peer (P2P) communications, multi-tier heterogeneous networks, full duplex communication, massive multiple-input-multiple-output (massive MIMO), millimeter-wave technologies, cognitive radios, beam division multiple access (BDMA), cloud-based radio access networks (CRAN) and wireless networks virtualization (WNV) [3]–[5].

One of the key features to achieve the forecast high data rates is the dense deployment of remote radio heads (RRHs). RRH is the name given to the RF front-end in mobile networks and encompasses the BS's RF circuitry plus analog-to-digital/digital-to-analog converters and up/down converters. The dense deployment of RRHs can be achieved, with a relatively lower cost when compared to the deployment of several physical RRHs, by multiplexing several different signals at a single RRH and transmitting them over several virtual RRHs. Carrier aggregation is one of the usage examples, where a base station (BS) increases its capacity by allocating more spectrum bandwidth. Infrastructure sharing is another usage example, where densification is achieved by sharing already deployed pieces of equipment, such as the RRH, among several wireless or mobile networks [6].

Infrastructure sharing has the advantage of reducing capital expenditure. It is well known that the deployment of cellular or wireless networks is expensive, and raising the capital for that effort is quite difficult as the operators always want to make the most out of the already deployed infrastructure. Therefore, in order to obtain a better return on the costs related to installation and maintenance of mobile or wireless infrastructure, it would be less expensive to share the already (or newly) deployed infrastructure with other operators than to build overlapping and concurrent infrastructure [7].

This way, RF front-end virtualization becomes very useful for flexible spectrum access, once it allows several physical layers (PHYs) to concurrently share a single physical wide-band RF front-end. Some direct consequences of RF front-end virtualization are the reduction in the required physical space for deployment (only a single wide-band RRH is required), energy consumption and price (reduction in redundancy) for the infrastructure provider. Figure 1 shows a multi-tenancy example where a single BS emulates several BSs where subscribers from different networks associate with their corresponding virtual BS.

Another way of achieving such demanding rates is through better utilization of the available and already underutilized spectrum bands. In general, due to the presence of primary and secondary users in shared licensed bands and of competing ($i.e.$, opportunistic) users in unlicensed bands, the available spectrum for users in a cognitive radio environment is fragmented and its use is intermittent, $i.e.$, the available spectrum

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

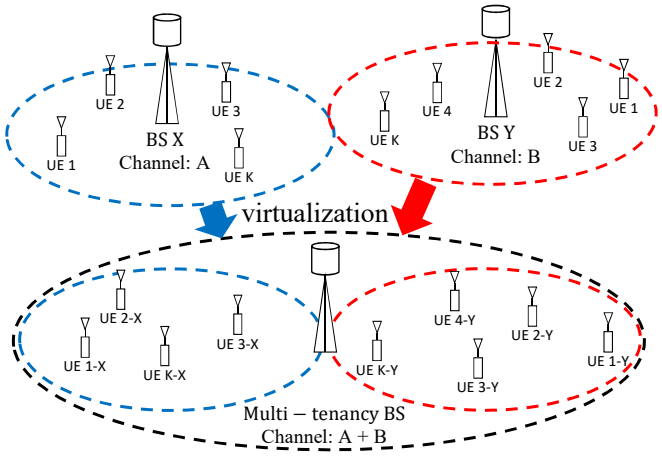

Fig. 1: A single base station emulating multiple virtual base stations. Subscribers from different networks associate with corresponding virtual BS although they use the same underlying infrastructure.

is split into non-contiguous chunks and is not used all the time. This intermittent and fragmented spectrum availability calls for a flexible and agile transmission scheme of the desired signals [8]. The concurrent transmission of several narrow orthogonal frequency-division multiplexing (OFDM)-based signals allows for the selective use of the available chunks of spectrum, which enhances the spectrum utilization, consequently improving the area throughput.

Therefore, in this work, we propose a non-intrusive and highly optimized wireless hypervisor architecture for software-defined radios (SDRs) that ensures coexistence, isolation, and programmability for multi-carrier-based systems ($i.e.$, OFDM) such as Wi-Fi, WiMAX, LTE, NB-IoT, etc. We focus on supporting OFDM-based systems once such waveform is still one of the most used and important ones being used even in 5G standards like the 5G New-Radio (NR) ones [9, 10]. Therefore, we understand that OFDM will still be widely adopted and employed for a long time.

We show that with the proposed architecture, spectrum programmability ($i.e.$, the ability to program/change frequency, bandwidth and gain settings) can be decoupled from physical layer (PHY) processing and delegated to a virtualization layer, ($i.e.$, the wireless hypervisor) which is added between a set of virtual PHYs (vPHYs) and the hardware/physical radio frequency (RF) front-end. The proposed solution can be applied to both of the approaches mentioned above of achieving higher area throughput.

We demonstrate that spectrum allocation can be decoupled from the PHY (or base-band processing) layer. In this way, the proposed hypervisor architecture supports flexible spectrum bandwidth allocation by creating a new layer located between several vPHYs and the physical RF front-end. The hypervisor dynamically maps the modulated signals of several vPHYs into configurable chunks of spectrum, before sending the resulting multiplexed signal of the vPHYs to the RF front-end. The hypervisor layer, which can be seen as a spectrum mapping/allocation layer, abstracts the underlying spectrum bandwidth dynamics and provides the vPHYs with a contiguous or non-contiguous (depending on the application of the proposed architecture) set of frequency subcarriers ($i.e.$, virtual spectrum bandwidth), where the desired spectrum bandwidth can be pre-defined (by operators for instance) or requested in an on-line basis by the vPHY itself.

Additionally, we present an open-source and free proof of concept prototype that was developed in order to assess the performance of proposed spectrum hypervisor architecture in real-world experiments. The source-code is available at [40] and can be used by researchers and developers to easily run experiments or extend the concept to other applications.

The remainder of the work is organised as follows. In section II, we present some aspects of hardware virtualization. In section III, we describe, mathematically, how we multiplex several OFDM signals. Next, in section IV, we list and discuss some important use cases where the proposed architecture can be employed. In section V we present and compare some related pieces of work on virtualization. Then, in section VI, we describe the proof of concept prototype developed in order to assess the feasibility of the proposed architecture. In section VII, some simulation results are presented showing the performance of the proposed architecture. Next, in section VIII, we present several experimental results obtained with the prototype. Finally, in section IX, we conclude our work with some conclusions and future work.

II. Hardware Virtualization

Hardware (in our case, the RF front-end) virtualization is achieved by means of hypervisors [11, 12]. A hypervisor is a hardware virtualization technique that abstracts, $i.e.$, isolates, multiple concurrent software radio protocol stacks ($e.g.$, LTE, NB-IoT, Wi-Fi, etc.), also known as software-defined radios (SDR), from the underlying radio hardware $i.e.$, RF front-end. It allows multiple radio stacks to run on top of a single piece of radio hardware at the same time, $i.e.$, concurrently, such that each protocol stack appears to have its own RF front-end (virtual RF front-end) that can be operated independently. One advantage is that wireless hypervisors help to maximize the effective use of the deployed infrastructure ($i.e.$, servers, RF front-ends, etc.).

The hypervisor is the mechanism allowing the seamless sharing of a particular resource by meeting three key requirements: abstraction, programmability, and isolation. The proposed solution addresses each of these requirements as shown in Figure 2 and briefly described next.

- **Abstraction**: this feature hides the underlying hardware characteristics and establishes simplified interfaces for accessing and sharing the hardware resources. This feature allows the clients of the hypervisor architecture to use it with no change to their upper layer radio stack, $i.e.$, from the MAC layer upwards. As shown in Figure 2, the proposed architecture provides concurrent access to multiple virtual RF front-ends, which are exposed to clients through several vPHYs. The proposed architecture can be supported by inexpensive RF front-ends ($e.g.$, commodity SDR equipment) or expensive wide-band RRHs. The wireless hypervisor ensures that multiple vPHYs can concurrently coexist on top of the same

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

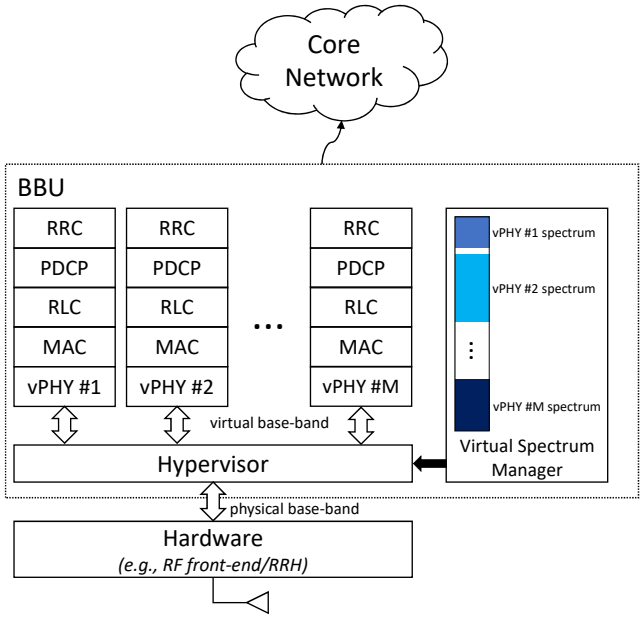

Fig. 2: High-level view of a hardware hypervisor in a 4G-like network.

physical RF front-end. The vPHYs are the point of access of the radio stacks to the virtual RF front-ends. Along with the hypervisor, the vPHYs provide a computationally efficient way of multiplexing several base-band signals into a wide-band signal, which consequently splits the spectrum bandwidth provided by the physical front-end. The proposed architecture supports the operation of multiple concurrent vPHYs, implementing totally different air-interfaces (as long as they are OFDM-based signals and have subcarrier spacings that are integer multiples of the smallest adopted spacing) with diverse processing constraints, channel bandwidths, medium access schemes, etc.

- **Programmability**: the proposed wireless hypervisor has to provide the multiple radio stacks running on top of it the same level of programmability (or configurability) provided by the single physical RF front-end. This feature is addressed by allowing the multiple radio stacks to configure different center frequencies, bandwidths and transmission/reception gains for individual virtual RF front-ends, $i.e.$, the virtual RF front-ends must provide the same set of functionalities provided by the physical RF front-end.
- **Isolation**: the proposed wireless hypervisor must make sure that any configuration or wrong configuration does not affect or cause interference to any other collocated radio stack. Isolation is the fundamental requirement that guarantees fault tolerance, security, and privacy to the multiple radio stacks running ($i.e.$, coexisting) on top of the same physical RF front-end [12]. This feature is enforced across multiple clients by providing them predefined access-points and bandwidths that do not overlap.

Figure 2 shows one out of several possible multi-tenancy uses for the proposed architecture, where it is employed in

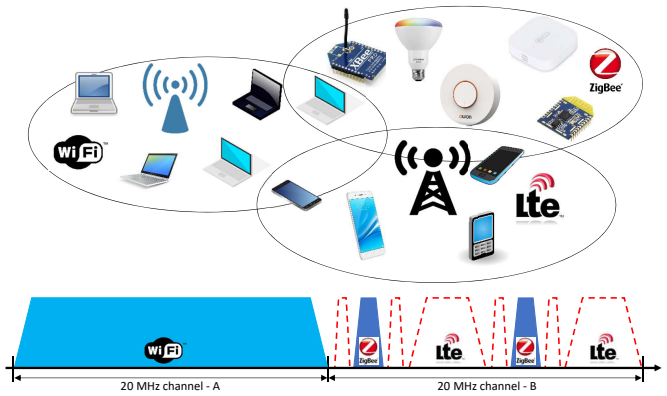

Fig. 3: Flexible and dynamic spectrum access example. Communications spectrum bandwidths can be flexibly and dynamically changed according to the availability of spectrum as well as application requirements.

4G-like mobile networks. However, it could be, for example, employed by Internet Service Providers (ISP) aiming at sharing their already deployed Wi-Fi access-points (or hot-spots) at locations such as airports, cafes and common shopping areas.

Figure 3 depicts yet another use case example regarding dynamic and flexible spectrum access. In the current wireless communications networks, channels have fixed central frequencies and bandwidths. Such static channel allocation causes spectrum fragmentation, lowering the spectrum utilization efficiency. Figure 3 shows a common coexistence scenario in industrial, scientific and medical (ISM) bands. In these bands it is common to have the coexistence of several different technologies, such as Wi-Fi, IEEE 802.15.4 ($e.g.$, ZigBee), LTE-U, etc. [13]. In the figure, two IEEE 802.15.4 networks occupy two channels that overlap with a 20 MHz-wide Wi-Fi channel. As can be seen, the two IEEE 802.15.4 networks may render the channel unusable for Wi-Fi communications due to the narrow-band interference caused by the two IEEE 802.15.4 networks. However, the remaining fragmented spectrum in that channel could be used by a flexible and dynamic system, as depicted by the red-dashed lines in Figure 3. As it is shown, the spectrum gaps created by the IEEE 802.15.4 transmissions could be used to establish LTE-like networks ($e.g.$, LTE-U, NB-IoT, etc.) or any other kind of OFDM-based network. This flexible and dynamic spectrum usage completely removes the concept of pre-defined and static channel allocations. Based on this approach, frequencies and bandwidths can be dynamically allocated based on the availability of spectrum and the requirements of the applications.

III. Multiplexing Concurrent OFDM Signals

In this section, we present a mathematical analysis of the working of the hypervisor proposed in this work and depicted in Figure 4.

For this analysis, we consider that complex modulated samples, s_i, carrying information are grouped into several narrow sub-bands, $B_k(i)$, which are composed of N_{vPHY} subcarriers. The complex samples, s_i, are symbols drawn from a digital modulation scheme such as BPSK, QPSK, 16QAM, etc. A maximum number of $M = \lfloor N/N_{\text{vPHY}} \rfloor$ narrow sub-bands, containing N_{vPHY} subcarriers can be used, where N is the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

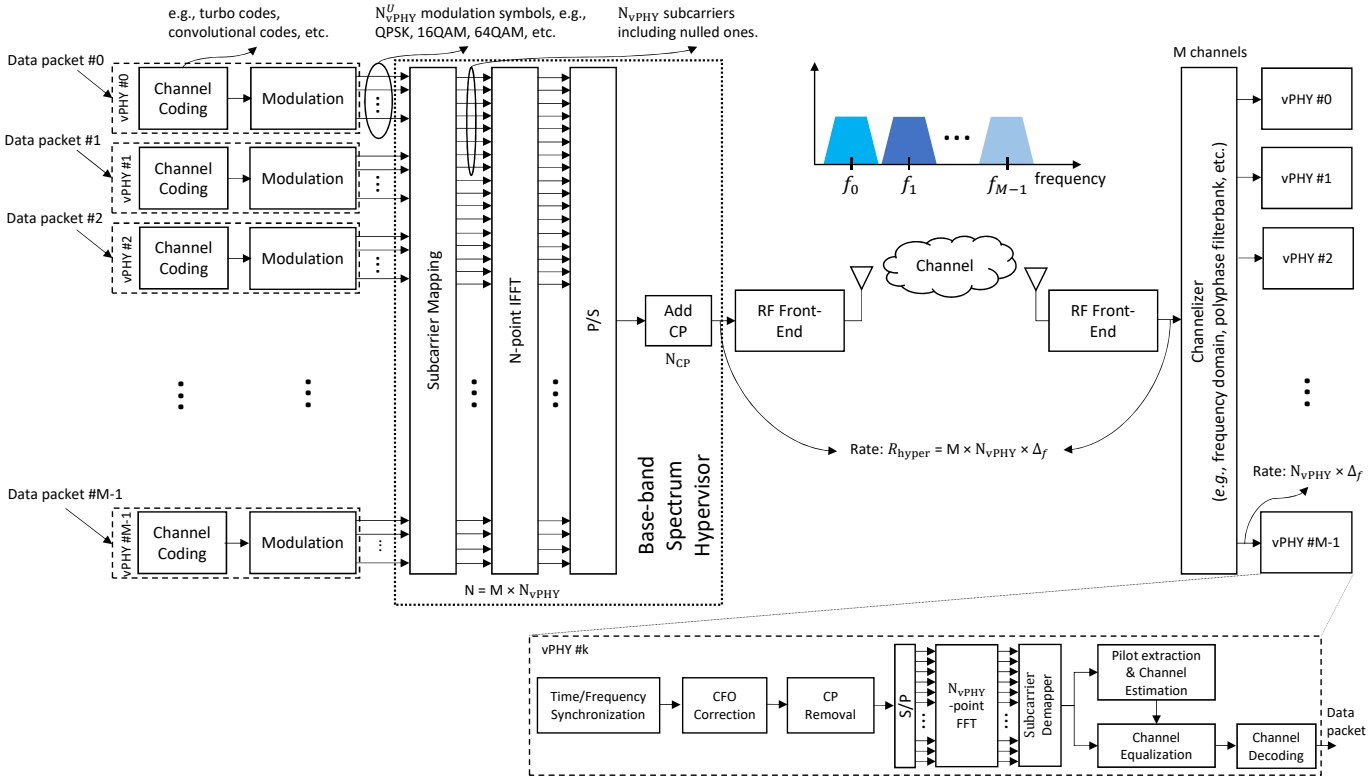

Fig. 4: Architecture of the proposed base-band wireless hypervisor.

length of the IFFT used in the hypervisor and $\lfloor c \rfloor$ is known as the floor operator and gives the largest integer value that is less than or equal to c. Out of the N_{vPHY} subcarriers, a narrow sub-band might have, N^U_{vPHY} carrying useful information, $i.e.$, the modulation symbols s_i, one Direct Current (DC) subcarrier, N^{DC}_{vPHY}, which is set to zero, and N^{Null}_{vPHY} subcarriers that are also set to zero. The nulled subcarriers can be used as frequency guard-bands at both edges of the narrow sub-band, allowing for the straightforward realization of anti-aliasing filters. On the other hand, the DC subcarrier allows the use of the simpler and cheaper direct-conversion, also known as zero intermediate-frequency (IF), RF front-ends. The narrow sub-band signal can be defined as

$$B_k(i) = \begin{cases} 0, & i = 0, \\ s_i, & 1 \leq i \leq \frac{N^U_{\text{vPHY}}}{2}, \\ 0, & (\frac{N^U_{\text{vPHY}}}{2}) + 1 \leq i \leq (\frac{N^U_{\text{vPHY}}}{2}) + N^{Null}_{\text{vPHY}}, \\ s_i, & (\frac{N^U_{\text{vPHY}}}{2}) + N^{Null}_{\text{vPHY}} + 1 \leq i \leq N_{\text{vPHY}} - 1. \end{cases} \tag{1}$$

For the sake of clarity, we consider that N is split into M sub-bands with N_{vPHY} subcarriers and that the subcarrier mapping block in the hypervisor maps complex samples into contiguous sub-bands. Note that the mapping of the M sub-bands, $B_k(i)$, into the IFFT subcarriers/bins corresponds to a frequency translation (mixing) of the sub-band signal, which is centered around 0 [Hz], to a frequency offset defined by the bin number times the subcarrier spacing, $\Delta_f = R_{\text{hyper}}/N$, where R_{hyper} is the sampling rate used by the hypervisor. The hypervisor's sampling rate, R_{hyper}, defines the amount of physical spectrum bandwidth that it can virtualize.

The set, $S(i)$, containing the concatenation of all the M narrow sub-bands, $B_k(i)$, and of length N is transformed into time domain using an IFFT of size N as

$$\begin{aligned} s(n) &= \frac{1}{N} \sum_{i=0}^{N-1} S(i) e^{-j2\pi in/N} \\ &= \frac{1}{N} \sum_{k=0}^{M-1} \sum_{i'=0}^{N_{\text{vPHY}}-1} B_k(i') e^{-j2\pi in/N}, \end{aligned} \tag{2}$$

where $i = i' + kN_{\text{vPHY}}$. Then, based on (2), the transmitted wide-band signal, $y_{\text{wBB}}(n)$, including a gain factor and the appending of a cyclic prefix can be represented by

$$y_{\text{wBB}}(n) = \frac{1}{N} \sum_{k=0}^{M-1} \sum_{i'=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i') e^{-j2\pi(i'+kN_{\text{vPHY}})n/N}, \quad -N_{CP} \leq n \leq N-1, \tag{3}$$

where ρ_k is the frequency amplification factor (or frequency gain), varying from 0 to 1, applied individually to each sub-band and N_{CP} is the cyclic-prefix (CP) length, which is an integer multiple of M. As can be noticed, $y_{\text{wBB}}(n)$ is a wide-band signal containing all the M multiplexed vPHY signals. As the final step in the transmission chain, the wide-band signal $y_{\text{wBB}}(n)$ is sent to the RF front-end, which will translate the signal into the desired pass-band frequency.

At the receiver side, the wide pass-band signal is translated into a wide base-band signal by the receiver RF front-end. Here we do not consider any channel influence, $i.e.$, the transmitted signal does not suffer from any impairment, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

consequently, it is perfectly received at the receiver side. Next, after the frequency translation, the received wide base-band signal $y^r_{\text{wBB}}(n) = y_{\text{wBB}}(n)$, which contains the M multiplexed sub-band (narrow-band) signals, is digitally translated/split into M narrow base-band signals (each one centered around 0 [Hz]), bandwidth restricted by low-pass digital Finite Impulse Response (FIR) filters, and finally, submitted to a sample rate reduction. These steps are depicted in Figure 5, which shows the architecture of a conventional channelizer as described in [14]. As shown in the figure, the output of the k-th channel, which is denoted by $y^{r_k}_{\text{wBB}}$, prior to the down-sampling, is a convolution as defined by

$$\begin{aligned} y^{r_k}_{\text{wBB}}(n) &= \left[y^r_{\text{wBB}}(n)e^{j\theta_k n}\right] \circledast h(n) \\ &= \sum_{l=0}^{N_{\text{FIR}}-1} y^r_{\text{wBB}}(n-l)e^{j\theta_k(n-l)}h(l), \\ 0 &\leq n < N + N_{\text{FIR}} - 1 \end{aligned} \tag{4}$$

where N_{FIR} is the length of the FIR filter, $\theta_k = \frac{2\pi\delta_k n}{N}$ is the angle corresponding to the digital frequency offset, and δ_k is the center-frequency of the k-th narrow sub-band, B_k, in number of subcarrier spacings, Δ_f. After substituting (3) into (4) we have

$$y^{r_k}_{\text{wBB}}(n) = \sum_{l=0}^{N_{\text{FIR}}-1} \frac{1}{N} \sum_{k=0}^{M-1} \sum_{i'=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i') e^{\frac{-j2\pi\Theta_{i',k}(n-l)}{N}} h(l), \quad -N_{CP} \leq n \leq N-1, \tag{5}$$

where $\Theta_{i',k} = (i' + kN_{\text{vPHY}} - \delta_k) \mod N$. The multiplication of $y^r_{\text{wBB}}(n)$ by $e^{j\theta_k n}$ centers the k-th narrow sub-band, B_k, around 0 [Hz] and the convolution with the low-pass filter makes sure that only the signal components belonging to the k-th narrow sub-band go through it while the other sub-bands are filtered out. This way, (5) can be re-written as

$$y^{r_k}_{\text{wBB}}(n) = \frac{1}{N} \sum_{i=0}^{N-1} \begin{cases} \rho_k B_k(i) e^{\frac{-j2\pi ni}{N}}, & 0 \leq i \leq \frac{N_{\text{vPHY}}}{2}, \\ \rho_k B_k(i) e^{\frac{-j2\pi ni}{N}}, & N - \frac{N_{\text{vPHY}}}{2} + 1 \leq i \leq N-1, \\ 0, & \text{otherwise}, \end{cases} \quad -N_{CP} \leq n \leq N-1. \tag{6}$$

As can be seen by inspecting (6), out of N subcarriers/bins, only the lower and upper $\frac{N_{\text{vPHY}}}{2}$ bins contain useful values, *i.e.*, values different from zero. We consider in (6), that the filters have a perfect window-shaped frequency response, which is not true in practice but can be approximated by well-designed high order low-pass filters [15].

Next, the wide base-band representation of the k-th narrow sub-band signal, $y^{r_k}_{\text{wBB}}(n)$, is subject to a down-sampling operation, as depicted in Figure 5, in order to have a narrow base-band representation of the sub-band signal, which can then be fed into a PHY for data decoding. After down-sampling, the signal expressed by (6), can be written as

$$y^{r_k}_{\text{wBB}}(nM) = \frac{1}{N} \sum_{i=0}^{N-1} F(i) e^{\frac{-j2\pi ni}{N/M}}, \quad -N_{CP} \leq nM \leq N-1. \tag{7}$$

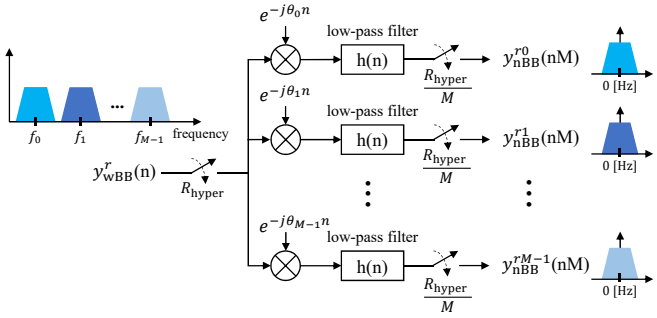

Fig. 5: Architecture of a conventional channelizer: frequency offset to base-band, low-pass filters, and down-samplers.

where the function $F(i)$ is defined as

$$F(i) = \begin{cases} \rho_k B_k(i), & 0 \leq i \leq \frac{N_{\text{vPHY}}}{2}, \\ \rho_k B_k(i), & N - \frac{N_{\text{vPHY}}}{2} + 1 \leq i \leq N-1, \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

Next, after the down-sampling operation the complex periodic signal $e^{\frac{-j2\pi ni}{N}}$ has its periodicity reduced from N to N/M samples, $e^{\frac{-j2\pi ni}{N/M}}$, and therefore, we can rewrite (7) as

$$\begin{aligned} y^{r_k}_{\text{wBB}}(nM) &= \frac{1}{N} \left[\sum_{i=0}^{N_{\text{vPHY}}-1} F(i) e^{\frac{-j2\pi ni}{N/M}} \right. \\ &\left. + \sum_{i=N_{\text{vPHY}}}^{2N_{\text{vPHY}}-1} F(i) e^{\frac{-j2\pi ni}{N/M}} + \cdots + \sum_{i=(M-1)N_{\text{vPHY}}}^{MN_{\text{vPHY}}-1} F(i) e^{\frac{-j2\pi ni}{N/M}} \right] \\ &= \frac{1}{N} \sum_{i=0}^{N_{\text{vPHY}}-1} \left[\sum_{m=0}^{M-1} F(i + mN_{\text{vPHY}}) \right] e^{\frac{-j2\pi ni}{N/M}}, \\ &\quad -N_{CP} \leq nM \leq N-1. \end{aligned} \tag{9}$$

From (8) we notice that only half of the first and last terms of the summation in (9) are different from 0, and therefore, it can be re-written as

$$\begin{aligned} y^{r_k}_{\text{wBB}}(nM) &= \frac{1}{N} \sum_{i=0}^{\frac{N_{\text{vPHY}}}{2}} F(i) e^{\frac{-j2\pi ni}{N/M}} \\ &+ \frac{1}{N} \sum_{i=\frac{N_{\text{vPHY}}}{2}+1}^{N_{\text{vPHY}}-1} F\left(i + (M-1)N_{\text{vPHY}}\right) e^{\frac{-j2\pi ni}{N/M}}, \\ &\quad -N_{CP} \leq nM \leq N-1. \end{aligned} \tag{10}$$

Finally, by using (8) and remembering the fact that $N/M = N_{\text{vPHY}}$, we show that the time-domain representation of the k-th narrow sub-band, $B_k(i)$, is recovered by the k-th branch of the channelizer (see Figure 5)

$$\begin{aligned} y^{r_k}_{\text{nBB}}(n) &= \frac{1}{N} \sum_{i=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i) e^{\frac{-j2\pi ni}{N_{\text{vPHY}}}}, \\ &\quad -N_{CP}^{\text{vPHY}} \leq n \leq N_{\text{vPHY}} - 1, \end{aligned} \tag{11}$$

where $N_{CP}^{\text{vPHY}} = N_{CP}/M$ is the cyclic prefix length of the vPHYs.

It is important to mention that this analysis is the same for any other narrow sub-band centered at any of the N

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

subcarriers of the hypervisor's IFFT module. Additionally, it is also important to notice that each one of the narrow sub-bands, which are multiplexed by the hypervisor, can have different widths. The only requirement is that the width of individual sub-bands is a divisible multiple of N. In this case, if a device is receiving more than one of the multiplexed narrow sub-band signals, then, it has to employ a non-uniform channelizer [16], as polyphase filter-bank channelizers only extract equally spaced spectrum chunks.

A. Discussion

In this section, we discuss the proposed architecture, its functionalities, features and possible uses.

As mathematically shown earlier, the proposed wireless hypervisor provides a virtual and discretized (in steps of $\Delta_f = R_{\text{hyper}}/N$) base-band spectrum abstraction layer to the several vPHY layers sitting on top of it. Its key function is the multiplexing, in the frequency domain, of several narrow base-band signals. It receives concurrent sets of modulated signals ($e.g.$, BPSK, QPSK, M-QAM, etc.) from several vPHYs and maps these sets into continuous or non-contiguous subcarriers ($i.e.$ spectrum) that will then be converted into time-domain by the N-point IFFT module and have a proper N_{CP} samples long CP added to it. The digital signal processing carried out by the spectrum hypervisor transforms the sets of modulated vPHY signals into a wide base-band waveform signal that is appropriate for transmission, while keeping the concurrent sets of vPHY signals unchanged and isolated from each other.

The hypervisor supports data flows from multiple concurrent vPHYs and provides each one of them a virtual RF front-end, which can have the following settings configured independently: frequency domain gain (ρ_k), frequency location (given by the mapping of the sub-band into the hypervisor's IFFT) and bandwidth of the vPHY (N_{vPHY}). Regarding the gain settings for each vPHY, as there is only one physical RF amplifier, the gain of the amplifier is set to a reasonable level (a level that avoids saturating the signal) and the independent vPHYs can set what we called frequency gain (ρ_k), which is a gain applied to the frequency domain signal and corresponds to a percentage ($i.e.$, 0 up to 100 %) of the RF amplifier's current gain.

Access to the wireless hypervisor is exposed through the vPHYs. The vPHYs work the same way as regular OFDM-based PHY layers ($e.g.$, Wi-Fi, WiMAX, LTE, etc.) being the only exception the removal of the OFDM modulation part. In the proposed architecture (see Figure 4), OFDM modulation is now executed by the wireless hypervisor, which efficiently multiplexes the base-band signal of several vPHYs into a single wide base-band signal. By removing and transferring the OFDM modulation of all PHYs to the hypervisor, we optimize the whole base-band processing performance by avoiding redundant/unnecessary, for example FFT and IFFT, operations that would be required to multiplex several concurrent regular PHY signals into a wide base-band signal [17]–[19]. The down-side of our approach is that it can only operate with multi-carrier-based ($i.e.$, OFDM) signals.

At the receiver side, different receiving approaches might be followed, depending on how the proposed hypervisor is employed to generate the wide base-band signal. The first approach refers to a communications connection between two devices ($i.e.$, a point-to-point connection). In this approach, as a first step, the received wide base-band signal coming from the RF front-end is split by a channelizer into equally or unequally ($i.e.$, non-uniform) spaced spectrum chunks and then, the down-converted and down-sampled narrow sub-band signals are fed into the narrow-band PHY receivers running on the device. The second approach is similar to the first one, where the only difference is that the signal coming from the RF receiver front-end is not split into narrow sub-bands. Here the wide-band received signal is fed directly into the OFDM demodulator of a corresponding wide-band vPHY. In the third approach, several independent, distributed and narrow-band devices have their RF receiver front-ends tuned to the center frequency and transmission bandwidth of each one of the transmitted vPHY signals. The fourth approach represents a mix of the previous three approaches, where there could be wide-band devices tuning to more than one transmitted narrow-band vPHY signal while other narrow-band devices would only be tuning to individual transmitted narrow-band vPHY signals. In all the approaches mentioned above, the signal multiplexing carried out at the transmitter side is transparent to the PHY receivers, meaning that the radio stack at the receivers does not need to be modified.

IV. Use Cases

In this section, we present and discuss some possible use cases for the proposed architecture.

A. Dynamic Spectrum Access

The Federal Communications Commission (FCC) has reported that licensed spectrum bandwidths are greatly under-used [20]. One of the examples mentioned in the report is the TV spectrum, which has one of the lowest utilization rates. That spectrum band is in most cases left totally unused in areas not so populated ($e.g.$, rural areas), and due to that, it is referred to as TV white-space. Based on this under-utilization of spectrum originally meant only for TV broadcasting, regulators are opening up this previously licensed spectrum for unlicensed use [21]. TV white-spaces for unlicensed use brought about a revolution to cognitive radio, spectrum sensing as well as to dynamic spectrum access [22]. TV white-spaces allow for the opportunistic use of vacant TV channels for data communications [23, 24].

Therefore, based on the opportunistic use of vacant spectrum, one of the use cases of the proposed wireless hypervisor is instantiating it as a Non-Contiguous Orthogonal Frequency Division Multiplexing (NC-OFDM) PHY, where one or several vPHYs could, based on spectrum availability/occupancy, have their data mapped into subcarriers that do not interfere/overlap with primary users in a Dynamic Spectrum Access (DSA) scheme, and thereby enabling efficient use of the spectrum [8]. For this use case, an NC-OFDM PHY node would employ the proposed hypervisor architecture as a wide-band OFDM modulator, where the subcarrier mapping module depicted in Figure 4 maps the modulated symbols only into subcarriers

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

that correspond to vacant spectrum, *i.e.*, not being used by the primary or other users. At the receiver side, the signal is demodulated and decoded as a wide-band NC-OFDM signal where some subcarriers had their values set to zero. The PHY receiver for this signal is depicted in the bottom right corner of Figure 4, and is nothing but a generic OFDM demodulator/decoder where the subcarrier demapper module needs to know beforehand which subcarriers are active during a given transmission interval [8].

Figure 6 depicts a possible snapshot of the spectrum usage in the TV white-space when the proposed architecture is used as a dynamic packet-based PHY layer with allocated bandwidth being dynamically changed according to vacant spectrum. In this example, the PHY layer establishes concurrent/simultaneous communications with several nodes. As can be seen, the channel's center frequency and allocated transmission bandwidth are changed throughout time, depending on spectrum availability and/or traffic load.

Another use case for the proposed architecture is in the implementation of a spectrum sharing scheme between wireless operators and Fixed Satellite Services (FSS) in the Citizens Broadband Radio Service (CBRS) band. FSS stations must share spectrum with new entrant wireless operations, while the entrant networks must ensure that the interference that they introduce to the incumbent FSS remains below a specified threshold [26, 27]. The aim with the launch of CBRS in the USA is that wireless systems should dynamically share the spectrum among them [27]. In this use case, the proposed architecture could easily adapt its transmission bandwidth to the available (either continuous or discontinuous) spectrum at any specific time. This use case is similar to the one described by Figure 6. Mainly aligned with this use case, 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 smarter/more intelligent coexistence and collaboration techniques of heterogeneous networks in the same wireless spectrum bands [54, 55]. One of the examples they have been advocating for is the adoption of such spectrum-sharing technologies in the CBRS band [56].

B. Network Densification

The capacity of an additive white Gaussian noise (AWGN) channel is given by the following equation [6]

$$R < C = m\left(\frac{W}{n}\right)\log_2\left(1+\frac{S}{I+N}\right), \qquad (12)$$

where W is the BS allocated bandwidth, n denotes the BS load factor (*i.e.*, the number of users sharing the given BS), m is the spatial multiplexing factor (*i.e.*, it denotes the number of spatial data streams connecting the BS and devices), S gives the signal power and I and N represent the interference and power noise, respectively, experienced at the receiver side. After analyzing the equation, it is possible to see that the capacity can be increased by decreasing the BS load factor, which can be attained through cell splitting. Cell splitting involves deploying a larger number of BSs and making sure that the user traffic is evenly distributed among all the deployed BSs [6]. A possible consequence of cell splitting/densification is that it might improve the signal power, as the users could now be closer to the new deployed BS and consequently experiencing a reduced path-loss.

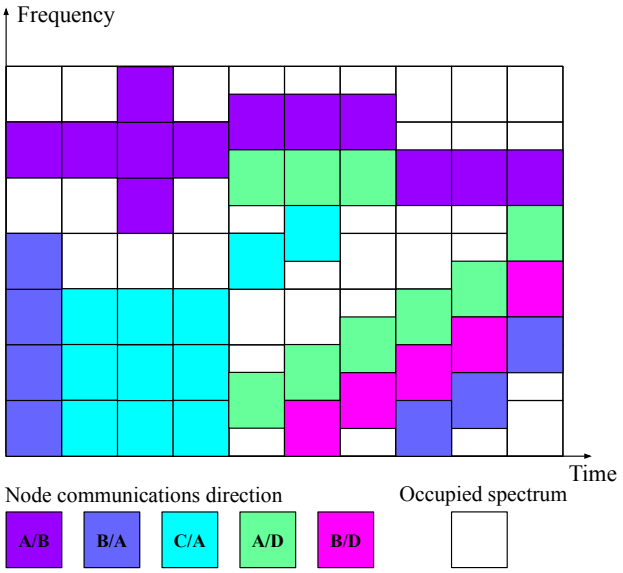

Fig. 6: Snapshot of the spectrum usage in the TV white-space.

The dense deployment of infrastructure is a precondition for cutting the BS load-factor n down in (12). However, the deployment of additional BSs entails significant costs and detailed site survey/planning [25]. Therefore, another use case, which represents a very important application for the proposed architecture, is instantiating the proposed architecture as an RF front-end multiplexer, where several vPHYs share a single physical RF front-end. This could be employed in multi-tenancy cases [17], where a cellular network infrastructure provider shares their owned infrastructure and/or spectrum with mobile virtual network operators (MVNOs) or vertical markets such as energy, automotive, city management, food and agriculture, health-care, government, public transportation, manufacturing, etc. In this case, the infrastructure provider shares its already deployed infrastructure, which ranges from base-band processing units (BBUs) to the RF front-ends (also known as Remote Radio Head - RRH). By employing multi-tenancy schemes, operators, MVNOs and verticals can decrease their capital expenditure (CAPEX) and operational expenditure (OPEX). Multi-tenancy makes the deployed infrastructure more energy efficient/greener by allowing a reduced number of BS sites, and, therefore, largely reducing the power consumption of air conditioning and other site support pieces of equipment.

The infrastructure provider can, for example, split its owned spectrum band into smaller chunks and lease it to MNVOs or verticals. Another possible example is instead of splitting its own spectrum into smaller chunks, is the possibility of providing other operators access to their own spectrum bands due to the wide-band capabilities of the current RRHs, ranging from 10 to 250 MHz of useful instantaneous bandwidth [28]–[30]. The virtualization of the physical RRHs is an important

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

TABLE I: COMPARISON OF RELATED WORKS.

	Related Work										
Comparison Metric	This work	[19]	[35]	[36]	[33]	[37]	[23]	[38]	[39]	[17]	[34]
Multiplexing domain	Frequency	Frequency	Frequency	Frequency	Time	Frequency	Frequency	Frequency	Frequency	Frequency	Time
Virtual resource type	Spectrum	Spectrum	Spectrum	RB	AP in time	RB	Spectrum	RB	RB	Spectrum	AP in time
RAT agnostic	Only MC[10]	Yes	Yes	Only WiMax	Only Wi-Fi	Only LTE	Yes	Only WiMax/LTE	Only LTE	Yes	Only Wi-Fi
Spectrum flexibility[11]	Yes	Yes	Yes	Yes	FC/FB[12]	Yes	Yes	Yes	Yes	Yes	FC/FB[12]
Operation mode	S/PB	S	S	S	PB	S	S	S	S	S	PB
Simulation validated	Yes	No	No	Yes	No	No	No	No	Yes	No	No
Experimentally validated	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Implementation	SW	SW	SW	SW	SW	SW	SW	SW	SW	SW	SW
Create self-interference	No[1]	Yes	Yes	No[1]	No[2]	No	Yes	No	No	Yes	No[2]
Create OOB emissions	Yes	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes
Individual gain control	Yes	Yes	Yes	NA	No	No	Yes	NA	NA	No	No
Open-source prototype	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
Dynamic allocation[3]	Yes	No	Yes	NA	Yes	Yes	No	Yes	Yes	No	Yes
Radio resource isolation[4]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual radio independence[5]	Yes	No	Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Concurrent Tx/Rx at different channels	Yes	Yes	Yes	Yes	No	No[6]	Yes	No[6]	No[6]	Yes	No
Number of VR instances[7]	12	2	2	2 slices[9]	2[8]	3 slices[9]	3	NA	4 slices[9]	4	2[8]

Legend: RB: resource-block - MC: multi-carrier - AP: access-point - SW: software - S: streaming - PB: packet-based - NA: information not-available - VR: virtual-radio - FC: fixed channel center frequency - FB: fixed channel bandwidth.

[1] OFDM based systems do not create self-interference as the subcarriers are mutually orthogonal.
[2] Once it time-multiplexes the access-point.
[3] Allow on demand destruction and creation of VRs without interrupting the operation of the spectrum hypervisor or other VRs.
[4] Allocate non-overlapping sub-bands to different VRs and prevent interference among VRs, *e.g.*, guard bands.
[5] Ensure that VRs cannot interfere with the operation and performance of other VRs, even in the case of a malfunctioning or misbehaving VRs.
[6] Use the concept of virtual RBs, therefore, concurrent transmissions only happen at different RBs within the same channel.
[7] The maximum number of VRs instantiated during the experiments.
[8] A single AP PHY that is time-shared between two VRs.
[9] A single PHY layer that has its RBs split into slices, creating VRs.
[10] Optimized for multi-carrier-based wave-forms, *e.g.*, OFDM.
[11] Flexibility in setting different channel center frequencies and bandwidths to different concurrent VRs.
[12] The approach only works with fixed channel center frequencies and bandwidths as it time-multiplexes a Wi-Fi AP.

step in the direction of multi-tenancy networks as being studied by the 3GPP [31].

This possible application/instantiation of the proposed architecture enables the Radio Access Network (RAN) to be made available as a service (RANaaS) to MVNOs and verticals. Virtual PHYs create new ways for infrastructure providers to monetize their owned spectrum bandwidth and deployed infrastructure. In this way, infrastructure providers can offer virtual PHYs as a service (vPHYaaS) in order to provide isolated and independent virtual networks to MNVOs and/or verticals sitting on the top of a shared physical infrastructure [32]. For this use case, a BBU, providing vPHYs as a service to MNVOs/verticals/operators, would have the proposed hypervisor architecture multiplexing the signal of several vPHY at downlink side while a channelizer would be deployed at uplink side in order to provide each one of the vPHYs with a signal corresponding to its allocated bandwidth, just as depicted in Figure 4. This use case is aligned with standardization efforts made by 3GPP that consider a BS serving both LTE and NB-IoT users [32].

V. RELATED WORK

In this section, we describe and discuss some related pieces of work on virtualization.

The related works on virtualization can be split into two main categories, time- or frequency-multiplexing. The time-multiplexing approaches achieve virtualization by splitting the access time to a common Wi-Fi access point PHY layer [33, 34]. This is possible due to the fact that Wi-Fi is a packet-based wireless network, where access points do not transmit data all the time, and therefore, being able to allow virtual Wi-Fi radio stacks to use its idle time. The frequency-multiplexing approaches can be further divided into two sub-categories based on the type of the virtualized resource, which can be the spectrum bandwidth through the virtualization of the RF front-end [17]–[19, 23, 35] or the time-frequency resource grid through the virtualization of the resource blocks (RB) provided by Orthogonal Frequency-Division Multiple Access (OFDMA)-based radio technologies such as LTE and WiMax [36]–[39].

Out of all the compared related works, only a few are technology agnostic, *i.e.*, can provide virtual radios to any radio access technology (RAT) [17]–[19, 23, 35]. However, the issue with these works is that they exchange optimized performance for being generalized so that they can operate with different RATs. In most of these works [17]–[19, 35], in order to be agnostic, the hypervisor layer re-implements operations (*e.g.*, FFT, IFFT, Sub-carrier Mapper, etc.) that are already performed by the PHY layers of the individual RATs, creating an extra overhead that decreases the performance of the solution.

An important comparison point is the dynamic creation and destruction of virtual radios without interfering with or stopping the hypervisor or other already instantiated virtual radios. Out of all works, only a few do not support such feature [17]–[19, 23, 36], where the number of instantiated virtual radios and their respective bandwidth allocations must be configured before running the hypervisor.

Another interesting point of discussion is the maximum possible number of instantiated virtual radios concurrently running during the experiments. The compared works were experimentally tested with the number of concurrently running virtual radios ranging from 1 to 4 virtual radios, however, on the other hand, our prototype has been experimentally tested

with 12 concurrently running vPHYs.

Finally, it is also important to add that independent gain configuration is an important feature to be exposed to the virtual radios once, for example, devices might be at totally different locations, however, it seems that independent gain configuration is not a major concern to most of the related works, once only a very few mention its support [19, 23, 35].

Differently from the other compared works, the virtualization architecture proposed in this work was designed to be highly optimized for sharing the same underlying physical RF front-end among several concurrent multi-carrier based virtual radios ($i.e.$, vPHYs). It provides dynamic access to several concurrent and configurable virtual RF front-ends ($e.g$, frequency gain, frequency location, and bandwidth) that are accessed through multiple vPHYs. The vPHYs can be instantiated in real-time without interfering with running vPHYs and without the necessity to stop the hypervisor.

The majority of the compared related works do not make their source code available [23, 33, 34, 36]–[39], however, we believe that research on this field can only progress if the different implementations are made available for comparison and a better understanding of their functionalities and features. Therefore, we make our proposed architecture prototype available at GitHub [40]. The source code includes some examples to measure the prototype's performance, a Graphical User Interface (GUI) for visualizing the transmitted spectrum and a channel emulator that can be used to run experiments without the necessity of having a dedicated piece of physical RF front-end. It emulates Additive White Gaussian Noise (AWGN), Rayleigh and Multi-path channels with several different Signal-to-Noise (SNR) values. The prototype makes extensive use of Single Instruction Multiple Data (SIMD) functions, including the FFT and IFFT implementations [41], making it even more optimized.

Table I presents a comprehensive comparison of related virtualization and hypervision works.

VI. Proof of Concept Prototype

In this section we describe a proof of concept prototype developed in order to verify the performance of proposed spectrum hypervisor architecture in real-world experiments.

Figure 7 depicts the high-level architecture of the implementation of the proposed architecture. As can be seen, the prototype is composed of several modules, namely, PHY communicator control, M vPHYs, and the hypervisor control module, which is in turn, composed of the modules hypervisor Tx and Rx.

The prototype is an open-source software-defined PHY layer designed to multiplex and receive ($i.e.$, demultiplex) the signal of several vPHYs [40]. It is implemented based on the srsLTE library [42]. srsLTE is an open-source and free LTE software-based library [42]. The prototype can run on top of several Ettus software-defined radio (SDR) devices such as the Ettus USRP X family or National Instruments' (NI) RIO SDR devices [43, 44] by using the Universal Software Radio Peripheral (USRP) Hardware Driver (UHD) software Application Programming Interface (API) [13, 45]. Therefore, the prototype accesses the SDR device through the UHD driver and its APIs [46].

TABLE II: List of messages and their respective parameters.

vPHY Message	Parameter	Type	Unit	Range
Tx control	vPHY ID	uint32	-	0-1
	Tx gain	float	%	0-100 %
	Tx vPHY BW	uint8	MHz	1-6 [1]
	Tx channel	uint32	-	≥ 0
	Data struct	uchar[]	-	uchar range
	MCS	uint8	-	0-28
	# of resource blocks	uint8	-	1-100
	User Data length	uint32	-	> 0
	User data	uchar[]	-	uchar range
Rx control	vPHY ID	uint32	-	0-1
	Rx vPHY BW	uint8	MHz	1-6 [1]
	Rx channel	uint32	-	≥ 0
Rx statistics	vPHY ID	uint32	-	0-1
	CQI	uint8	-	0-15
	RSSI	float	dBW	float range
	Noise	float	dBW	float range
	Decoded MCS	uint8	-	0-28
	CRC error counter	uint32	-	≥ 0
	Data length	uint32	-	≥ 0
	Received data	uchar[]	-	uchar range

The communication between the prototype and the upper layers is carried out through a set of three well-defined messages, which are exchanged over a ZeroMQ bus [48], $i.e.$, the prototype and the upper layers are interconnected through a publish-subscribe messaging system known as ZeroMQ [48]. ZeroMQ is a high-performance asynchronous messaging library, designed to be used in distributed or concurrent applications [48]. The set of vPHY messages is designed with Google's Protocol Buffers (protobuf) [47]. Protobuf is used for data serialization and works perfectly with the 0MQ messaging library [48].

The first two vPHY messages, called, Tx and Rx Control, are used to control and configure the transmission and reception of user data, respectively. The parameters in these two control vPHY messages should be configured and sent to the individual vPHYs by the upper layers before the transmission of every new subframe. Each vPHY control message, as the name suggests, controls the operation of only one vPHY. The remaining message, called Rx statistics vPHY message, is used to give upper layers feedback on the operation of each individual vPHY.

Tx control messages transport the user data to be transmitted ($i.e.$, TBs) and transmission parameters such as vPHY ID, vPHY Tx BW, MCS, Tx gain, Tx channel, number of resource blocks used by that user, data length, and data. The vPHY ID field is used in all messages to specify to which one of the vPHYs a control message is meant to or received from. **Rx control** messages are used to configure reception parameters such as vPHY ID, vPHY Rx BW, and Rx channel.

Rx statistics messages carry the vPHY ID, received decoded user data, and reception statistics such as Received Signal Strength Indication (RSSI), Channel Quality Indicator (CQI), decoded MCS, cyclic redundancy check (CRC) error counter, etc. The vPHY messages and their parameters are summarized in Table II.

[1]These numbers correspond to the following LTE bandwidths: 1.4, 3, 5, 10, 15 and 20 MHz respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

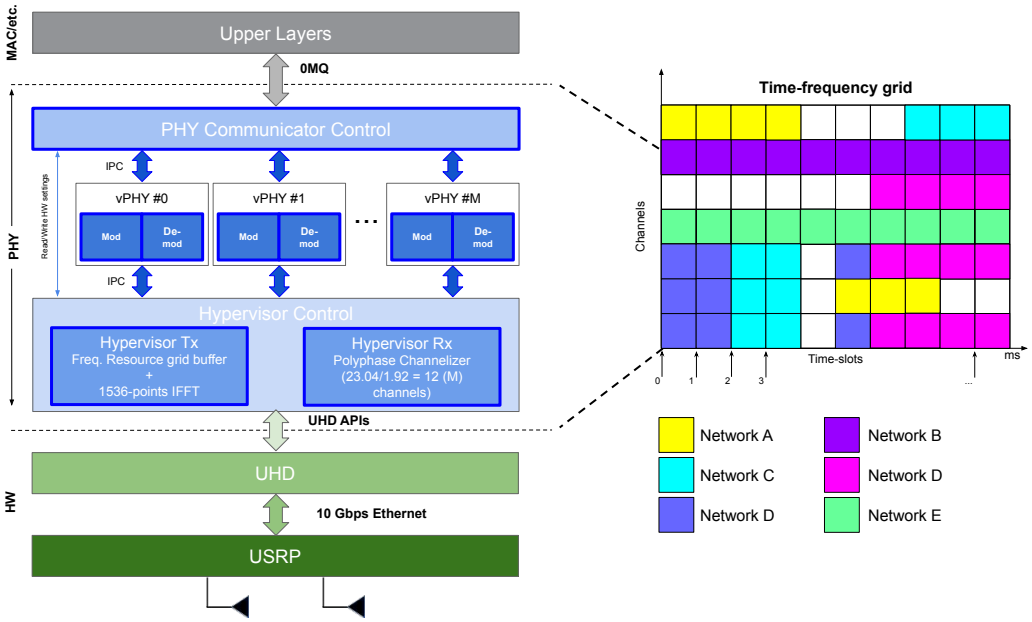

Fig. 7: High-level architecture of the implemented prototype.

Next, we describe each one of the modules composing the prototype.

- **PHY Communicator Control**: this module is responsible for the exchange of messages with several and possibly independent upper layers, *e.g.*, the MAC layers from different users or operators. This module works on a subframe basis, meaning that the connected upper layers always send/receive in one control message to/from the module the content of a subframe as the minimum unit of data exchange. The received control messages, carrying user data, are then relayed to the respective vPHY by using the vPHY ID in the control message. The decoded user data is sent to the respective upper layer also by using the vPHY ID.
- **vPHY**: modulates and demodulates the user data. After modulating the data, each vPHY maps, according to the channel configured in the Tx control message, its N^U_{vPHY} modulated symbols into a memory buffer, called, *resource grid buffer*, which is a 1 ms (*i.e.*, the duration of a subframe) representation of the frequency-domain spectrum band multiplexed by the PHY prototype. Each vPHY only has to map the N^U_{vPHY} data symbols (*i.e.*, useful symbols), while the remaining positions or subcarriers of the buffer have their values already set to zero before every new transmission. In the current implementation of the prototype, $N^U_{\text{vPHY}} = 72$ and $N_{\text{vPHY}} = 128$ subcarriers. The *resource grid buffer* is a discretized, in number of subcarriers or IFFT points, representation of the spectrum for the duration of 1 ms. In this version of the prototype, we used a 1536-point IFFT. The *resource grid buffer* is a memory buffer that is shared by all vPHYs. In the demodulation case, the IQ samples that are output by the Hypervisor Rx module are decoded accordingly by the respective vPHY. In the current implementation of the prototype 12 vPHYs can be instantiated and concurrently transmit/receive their data. This number of vPHYs is obtained by dividing the number of IFFT points, $N = 1536$, by the total number of vPHY subcarriers including the null ones, $N_{\text{vPHY}} = 128$.
- **Hypervisor Tx**: applies a 1536-point IFFT to the *resource grid buffer*, adds CP and transfers the IQ samples to the USRP for transmission over the air. The internal architecture of the Hypervisor Tx module is depicted in Figure 8. As showed in the figure, the Hypervisor Tx can be seen as an OFDM modulator where each OFDM symbol is created by reading and processing the consecutive 1536 data symbols stored at the *resource grid buffer*. As showed in Figure 8, the *resource grid buffer* stores data of 12 channels $\times$ 14 OFDM symbols, totalling 1 ms of data.
- **Hypervisor Rx**: applies a FIR polyphase filter-bank channelizer to the IQ samples received from the USRP and outputs the M down-converted channels to the vPHYs for data demodulation and decoding. The channels output by the channelizer are centered at 0, $N_{\text{vPHY}} \times \Delta_f$, $2 \times N_{\text{vPHY}} \times \Delta_f$, $3 \times N_{\text{vPHY}} \times \Delta_f$, ..., $M-1 \times N_{\text{vPHY}} \times \Delta_f$, respectively, where $N_{\text{vPHY}} \times \Delta_f = 1.92$ MHz

For improved processing performance, each one of the just described modules runs on an exclusive thread. As the prototype works on a 1 ms basis, it is possible to have a mix of streaming and bursty-based transmissions as shown in Figure 7.

VII. Simulation Results

In this section, we present some simulation results carried out in order to validate and assess the functionality of the proposed architecture.

For the first validation, we want to check the average Mean Squared Error (MSE) related to the multiplexing of several vPHY modulated signals by the hypervisor. In this simulation,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

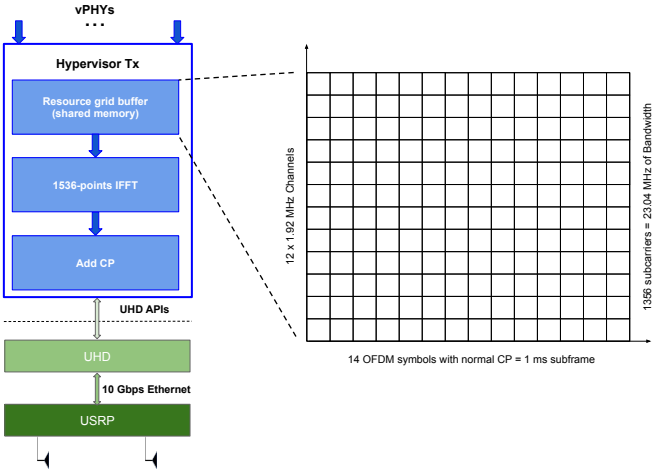

Fig. 8: Internal architecture of the Hypervisor Tx module.

we measure the error between OFDM symbols, y_{PHY}, created with an 128-point IFFT, N_{PHY}, plus 9-sample long CP, N_{PHY}^{CP}, and the OFDM symbols received from the hypervisor after down-conversion and channelization, y_{vPHY}. Each (v)PHY signal modulates 72 consecutive subcarriers, which translates into a useful transmission bandwidth of 1.08 [MHz] when a subcarrier spacing, Δ_f, of 15 [KHz] is used. We average the MSE error over 10^5 iterations, where at each iteration, we have the single PHY and all the vPHYs modulated with the same randomly generated data. For this simulation, we use 12 vPHYs, $i.e.$, $M = 12$, where each vPHY has its signals mapped into 128 consecutive subcarriers, N_{vPHY}, totalling 1536 subcarriers, which is the number of points used by the IFFT block in the hypervisor, N. At the receiver side, we use a polyphase FFT analysis filter-bank [14] to split the wide-band signal into multiple uniformly spaced narrow sub-bands. It has a 180 [dB] stop-band attenuation and 512 filter coefficients per sub-band. The frequency response of the polyphase filter-bank

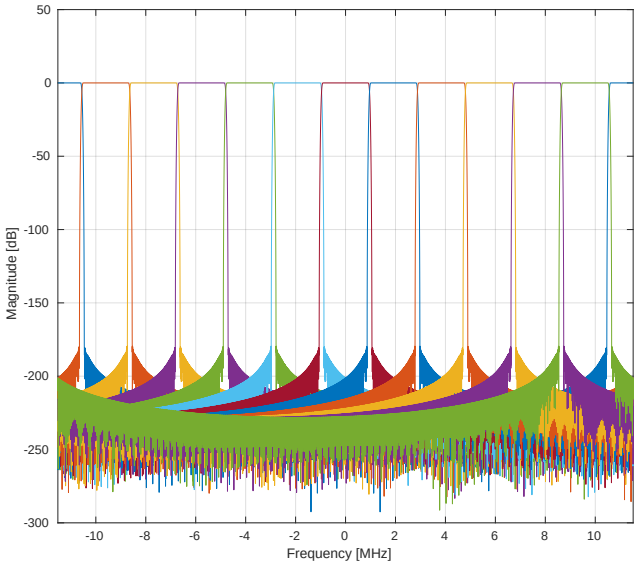

Fig. 9: Frequency response of a polyphase filter-bank with stop-band attenuation of 180 [dB] and 512 filter coefficients per sub-band.

used in all the simulations presented in this section is depicted in Figure 9. As can be seen in the figure, the physical spectrum band, R_{hyper}, which is provided by the RF front-end, is split into M equal-bandwidth equally spaced sub-bands of 1.92 [MHz] ($i.e.$, $\frac{N \times \Delta_f}{M}$). It is important to mention that the stop-band attenuation and filter order parameters play an important role in the fidelity of the multiplexed signals to the single PHY one [14]. The MSE for the k-th vPHY is calculated as defined by (13) below.

$$\text{MSE}_k = \mathbb{E}\left[\frac{1}{(N_{\text{PHY}} + N_{\text{PHY}}^{CP})} \times \sum_{n=0}^{(N_{\text{PHY}}+N_{\text{PHY}}^{CP})-1} |y_{\text{PHY}}(n) - y_{\text{vPHY}}(n)|^2\right]. \quad (13)$$

We additionally, we have also calculated the average Modulation Error Ratio (MER) as a way to compare the error introduced by the multiplexing to the vPHY modulated signal. The MER compares the error between the modulated data symbols ($i.e.$, the BPSK, QPSK, etc. symbols used to modulate the OFDM subcarriers) and the demodulated data symbols at the receiver side after all the vPHY, hypervisor and channelizer processing. Here we also take the MER average over 10^5 iterations. The MER for the k-th vPHY is calculated as follows

$$\text{MER}_k = \mathbb{E}\left[10\log_{10}\left(\frac{\sum_{n=0}^{N_{\text{vPHY}}^U-1}(I_k^2+Q_k^2)}{\sum_{k=0}^{N_{\text{vPHY}}^U-1}(I_k-\tilde{I}_k)^2+(Q_k-\tilde{Q}_k)^2}\right)\right], \quad (14)$$

where N_{vPHY}^U represents the number of useful subcarriers ($i.e.$, the subcarriers that are modulated with the data symbols), I_k is the In-phase value corresponding to the k-th reference symbol, Q_k is the Quadrature phase value corresponding to the k-th reference symbol, $\tilde{I}_k$ is the In-phase value corresponding to the k-th received symbol, and $\tilde{Q}_k$ is the Quadrature phase value corresponding to the k-th received symbol. The MER can be seen as a signal-to-noise ratio (SNR) measurements, where it calculates the distortion/interference caused by the multiplexing operation performed by the hypervisor.

In Table III we show the MSE and MER averages for several different modulation schemes. The MSE estimation values shown in the table were calculated averaging the error between PHY and vPHY OFDM symbols over 1×10^5 iterations. The MER averaging was also executed over the same number of iterations. For each iteration, the same set of randomly picked data bits modulates both the single PHY and the 12 vPHYs. The table shows that the MSE is quite low and almost the same for all modulation schemes and that the MER is high and the same for all schemes, which means that both MSE and MER are independent of the employed modulation scheme. Additionally, it is worth mentioning that as all vPHYs are fed with the same set of bits, their resulting MSE and MER are the same, and due to that, we only present one value per modulation scheme in Table 12.

In Table IV we show the comparison of MSE and MER for several different filter order values. As we noticed with the results in Table III, the MSE and MER values are almost the same for all modulation schemes, and therefore, in this

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

TABLE III: MSE AND MER FOR SEVERAL DIFFERENT MODULATION SCHEMES.

	Modulation Order					
	BPSK	QPSK	16QAM	64QAM	128QAM	256QAM
MSE	$3.8517e^{-10}$	$3.8521e^{-10}$	$3.8533e^{-10}$	$3.8526e^{-10}$	$3.8514e^{-10}$	$3.8526e^{-10}$
MER [dB]	70.572	70.572	70.572	70.572	70.572	70.572

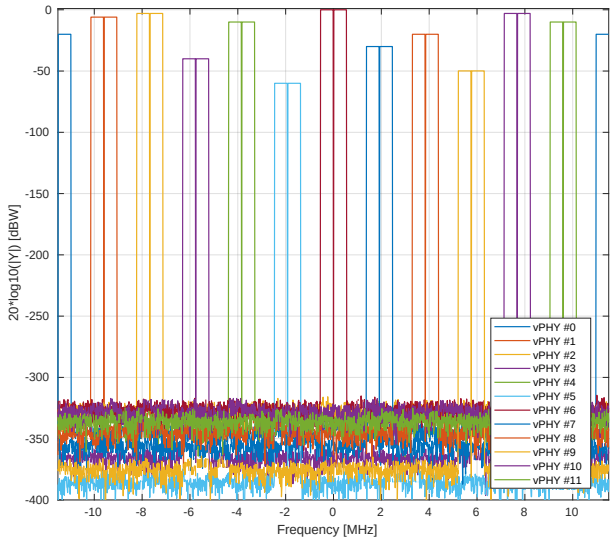

Fig. 10: Frequency-domain representation of the wide base-band signal generated by the hypervisor.

simulation, we use the same modulation scheme, QPSK, for all trials. As mentioned before, the filter order is important to guarantee a good signal fidelity. This is due to the fact that, the higher the filter order, the closer it is to the perfect window filter frequency response, which does not impose any distortion to the filtered signal [49].

In Figure 12, depicts the comparison between OFDM symbols generated by a plain ($i.e.$, single) PHY and 2 vPHYs having their output signals multiplexed by the hypervisor. Here, for the sake of comparison, the single PHY and the 2 vPHYs are fed with the same set of data bits. As can be seen, the vPHY OFDM symbols are quite similar to the OFDM symbol generated by the single PHY.

In Figure 10, we show the frequency-domain representation of the wide base-band signal generated by the proposed wireless hypervisor. Here, in this figure, the hypervisor multiplexes, in the frequency domain, the signals of 12 vPHYs, M, where each vPHY uses 128 subcarriers, N_{vPHY}, totalling 1536 subcarriers, which is the number of points used by the IFFT block in the hypervisor, N. Each vPHY only modulates 72 subcarriers and leaves the remaining subcarriers along its edges as guard-bands. The subcarrier at the center of each vPHY channel is set to 0, which is used to allow receivers to employ simpler/cheap direct-conversion ($i.e.$, zero intermediate-frequency) RF front-end receivers. In the example, shown in the figure, 64QAM

TABLE IV: MSE AND MER FOR SEVERAL DIFFERENT FILTER ORDERS.

	Filter Order					
	16	64	128	256	512	1024
MSE	1.3317e-03	1.9853e-08	3.6452e-09	3.2091e-09	3.8513e-10	2.3038e-10
MER [dB]	5.1897	53.4555	60.8164	61.3699	70.5720	72.8092

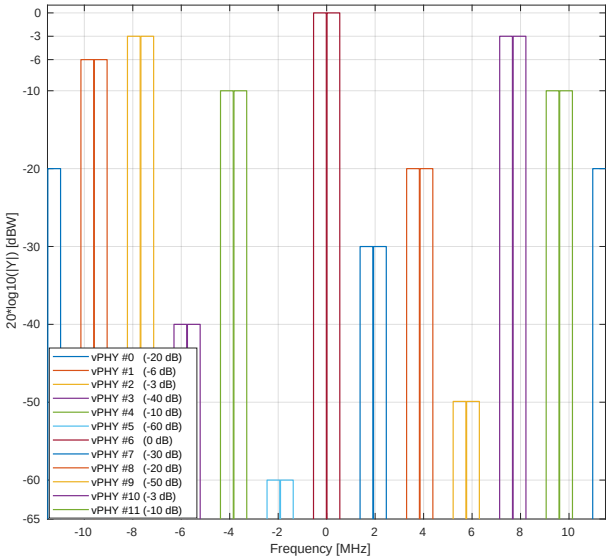

Fig. 11: Closer look at the frequency-domain boost factor.

modulation is used to modulate the data signal. As can be seen, there are (i) a guard-band between consecutive vPHY signals, (ii) a null-subcarrier exactly at the center of each one of the vPHY transmissions, (iii) and different transmission levels for the vPHYs. As shown in the figure, it is possible to give independent gains to each vPHY, which is accomplished by multiplying the N^U_{vPHY} useful modulation symbols by a multiplication factor varying from 0 (no transmission power at all) to 1 (maximum transmission power used by the physical RF front-end). Figure 11 gives a closer look at the boost factor, showing that it is possible to vary the transmitted power of each individual vPHY just by changing the factor used to multiple the useful subcarriers, N^U_{vPHY}. The legend on the figure shows the attenuation given to the default transmission

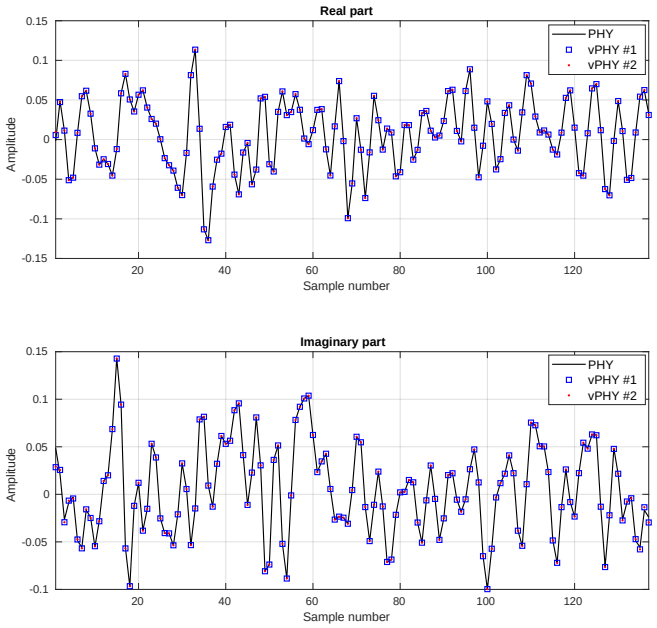

Fig. 12: Comparison between OFDM symbols generated by a plain/single PHY and 2 vPHYs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

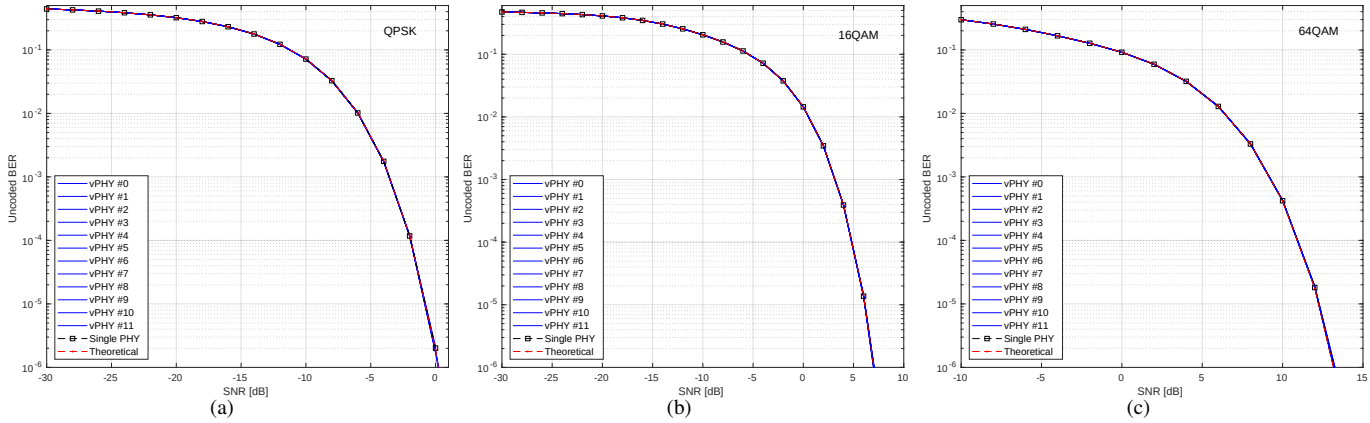

Fig. 13: BER curves for a setup where 12 vPHYs concurrently have their signals multiplexed by the hypervisor and use: (a) QPSK, (b) 16QAM and (c) 64QAM modulation schemes.

power of a vPHY, which is around 0 [dBW] as shown by the vPHY centered around 0 [Hz].

Figure 13 presents uncoded BER results for a setup where 12 vPHYs concurrently have their signals multiplexed by the hypervisor and employ QPSK, 16QAM, and 64QAM modulation schemes. In each one of the sub-figures, we compare the BER for the 12 vPHYs against the BER achieved by a single PHY, *i.e.*, there is no other signal being transmitted along with that of the PHY under test. In this simulation, we consider an Additive White Gaussian Noise (AWGN) channel. The single PHY and each one of the 12 vPHYs modulate 72 (*i.e.*, $\mathrm{N}^{U}_{\mathrm{vPHY}}$) out of 128 (*i.e.*, $\mathrm{N}_{\mathrm{vPHY}}$) consecutive subcarriers by using a 64QAM modulation scheme. The hypervisor's IFFT length, N, is set to 1536 and the cyclic prefix length, N_{CP}, is set to 108 samples. For each SNR point, 1×10^6 iterations were run, where the total number of wrongly decoded bits and transmitted bits were calculated for the BER calculation. As can be seen, the vPHY BER curves exactly match the single PHY BER curve (dashed black curve with squares along it), meaning that there is no interference between the current transmissions. Additionally, it is also worth mentioning that all curves match the theoretical BER curves (red-dashed curve with dots along it), which can be approximated by (15) [50].

$$P_b \approx \frac{2(\sqrt{\gamma}-1)}{\sqrt{\gamma}\log_2(\gamma)}\mathrm{erfc}\left(\sqrt{\frac{3\log_2(\gamma)(E_b/N_0)}{2(\gamma-1)}}\right), \qquad (15)$$

where γ is the modulation order and E_b/N_0 is the bit energy over the power spectrum density. The result presented in Figure 13(c), is very important as it proves that the proposed architecture provides perfect isolation among all the signals being multiplexed. The perfect isolation is due to the orthogonality provided by the IFFT processing in the hypervisor, which guarantees that every single subcarrier, spaced of Δ_f [Hz], is mutually orthogonal to all other ones.

VIII. Experimental Evaluation

In this section, we present some experimental results carried out in order to validate and assess the functionality of the proof of concept implementation of the proposed architecture. All the experiments presented in this section were carried out with the prototype running on a desktop with an Intel Xeon E5-2650 v4 CPU (@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 a x310 USRP with 10 Gigabit Ethernet link, and equipped with CBX-120 RF daughterboards [51]. These RF daughterboards operate from 1200 up to 6000 MHz with a bandwidth of 120 MHz [51].

For all the experiments presented in this section, each vPHY has a useful transmitting BW of 1.08 MHz, which is equivalent to 6 LTE RBs, and a guard-band of 420 KHz at each side of the transmitted spectrum, totalling 1.92 MHz of used BW (*i.e.*, useful-band plus guard-band sections) per vPHY, totalling $12\times$ 1.92 MHz/vPHY $=$ 23.04 MHz of occupied BW when we have 12 vPHYs operating concurrently.

Figure 14 shows the spectrum of 12 vPHYs transmitting concurrently. This figure was collected with an Anritsu MS2690A Signal Analyser. The RF front-end center frequency was set to 2.4 GHz and Tx gain set to 3 [dB] with the USRP Tx output connected to the signal analyser through a cable with 20 [dB] of attenuation. As can be seen, the total transmission BW spans over 23.04 MHz, which is equivalent to having 12 1.92 MHz wide vPHYs with their center frequencies located at 1.92 MHz apart from each other. Additionally, we can also see that the two vPHYs transmitting at the right and left edges suffer from attenuation, which is caused by cascaded integrator-comb (CIC) filter roll-off. CIC filter are implemented in the USRP to provide decimation by an arbitrary programmable integer decimation factor, however, they present a very significant pass-band roll-off, which are often called *spectral droop* or *CIC rolloff* [52].

Figure 15 shows the spectrogram (time versus frequency) for the same experiment setup used to capture Figure 14. In this experiment all 12 vPHYs transmit in streaming mode, *i.e.*, each vPHY transmits subframes all the time with no gap between subsequent subframes. The transmitted signal was captured for a period of 100 ms. It can be noticed that all 12 vPHYs transmit at the same time throughout the whole analysis interval, without any gap between consecutive subframes.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

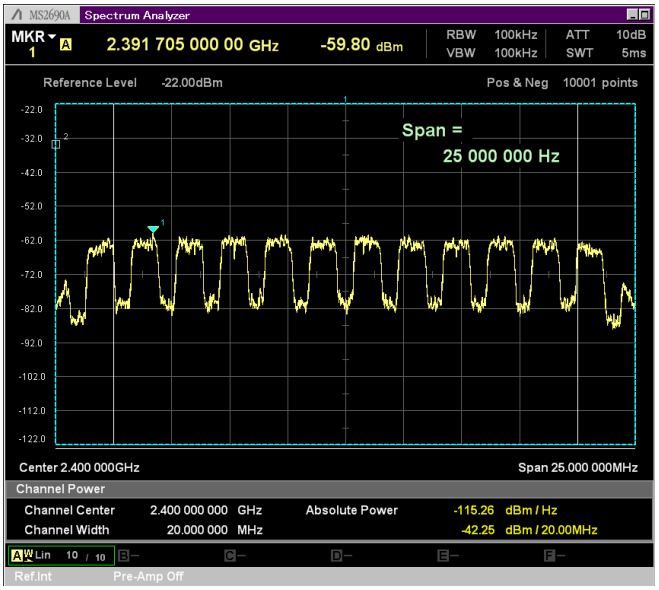

Fig. 14: Spectrum of 12 vPHYs concurrently transmitting at a center frequency of 2.4 GHz.

Fig. 15: Spectrogram of 12 vPHYs concurrently transmitting at a center frequency of 2.4 GHz.

Figure 16 shows that the proof of concept prototype of the proposed hypervisor is able to handle discontinuous transmissions and to apply independent gains to each vPHY, as described in Section III. Again, we used the same experiment setup used to capture Figure 14. In this experiment all 12 vPHYs transmit in discontinuous (bursty) mode with random number of subframes transmitted in a row, channel number and frequency amplification factor. The number of subframes, channel number and frequency amplification factor of each vPHY are randomly selected between the ranges $0-11$, $0-5$, and $0-100$ % respectively. The transmitted signal was captured for a period of 100 ms. As can be seen, the prototype is also able to work on bursty mode with independent frequency amplification factor for each vPHY. This result also shows

Fig. 16: Spectrogram of 12 vPHYs with discontinuous transmissions and independent frequency gains at a center frequency of 2.4 GHz.

that the prototype also supports run-time configuration of the number of transmitting vPHYs as we see that not all vPHYs might be transmitting during a period.

The following experiments are executed by adding a channel emulator between the Tx and Rx sides of the prototype. At the Tx side, the generated multiplexed signal, instead of being sent to the USRP HW is sent to an abstraction layer that emulates the HW and adds additive white Gaussian noise (AWGN) to the transmitted signal, next, the abstraction layer transfers the noisy signal to the receiving side of the prototype.

Figure 17 shows the throughput measurements taken with the proposed architecture prototype for several MCS values and a Duty Cycle (DC) of 95.24%. In this experiment, the prototype works in full-duplex mode ($i.e.$, it is simultaneously transmitting and receiving) with 1 single vPHY (upper part of the figure) and 12 vPHYs (lower part of the figure) working at the same time. We adopt a full-duplex mode in order to check if this mode somehow impacts the measured throughout, as in full-duplex mode the prototype is being fully utilised. The measurements were taken with transmissions of 20 ms ($i.e.$, 20 subframes) and a gap of 1 ms between subsequent transmissions, and therefore, a DC of 95.24%. The throughput is calculated as an average over 10 measurement intervals of 10 seconds each. During one measurement interval ($i.e.$, 10 s) the total number of received bits from all vPHYs is counted and then divided by the interval to produce the throughput measured during that interval. As in this experiment we are only interested in the maximum throughput that can be achieved, the SNR on the link was set to 30 [dB] so that the packet reception rate for all MCS values was equal to 1. For the sake of comparison, the theoretical maximum throughput achieved by the *Streaming* mode ($i.e.$, transmissions with a DC equal to 100%) is added to the figure. The theoretical maximum throughput is calculated by dividing the size in bits of a LTE transport block for each MCS value [53] by

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

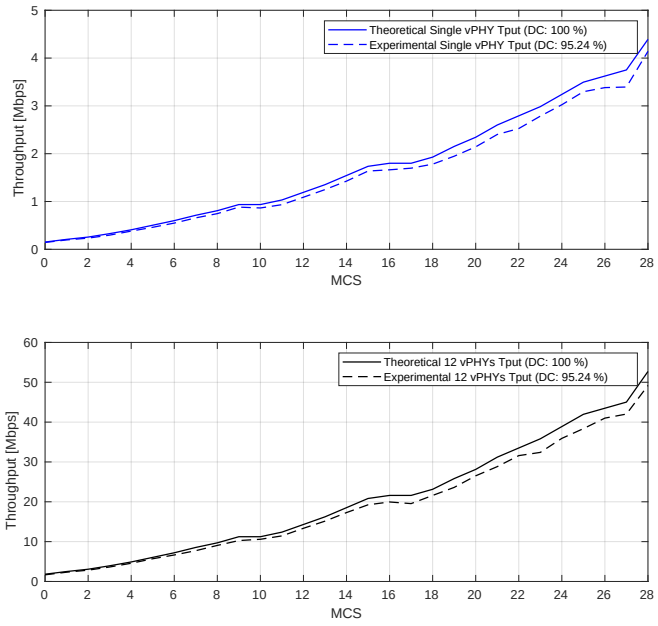

Fig. 17: Prototype's throughput for single and 12 vPHYs over different MCS values.

1 ms. As can be seen, the measured prototype's throughput approaches the theoretical maximum throughput for all MCS values, yielding more than 4.14 Mbps in the single vPHY case and more than 49.2 Mbps in the 12 vPHYs case for MCS 28. Additionally, as can be also noticed, the operation in full-duplex mode has no visible impact on the achieved throughput. This is due to the powerful server, with 12 CPU cores, used to run the prototype.

Next, in Figure 18 we compare the data packet reception ratio (PRR) of a single PHY against the data PRR of 12 vPHYs concurrently transmitting over a range of SNR values. In this experiment, 12 vPHYs concurrently have their signals multiplexed, transmitted and received by the prototype. The experiment uses 3 different MCS values so that all 3 modulation schemes employed by LTE standard ($i.e.$, QPSK, 16QAM, and 64QAM) are tested. The PRR is calculated as the average over 10^5 Monte Carlo trials, where at each trial, the Tx side of the prototype sends either a single PHY signal or the multiplexed signal of 12 vPHYs. In each one of the sub-figures, we compare the PRR of the 12 vPHYs against the PRR obtained with a single PHY, $i.e.$, there is no other signal being transmitted along with that of the PHY under test. As theorised earlier, the PRR curve of the 12 concurrent transmitting vPHYs match the PRR of the single PHY ($i.e.$, the dashed black curve with squares along it). This means that there is no interference between the current vPHY transmissions. These results prove that the prototype of the proposed architecture also, as showed before with the simulation results, provides isolation among all the signals being multiplexed by the prototype. The achieved isolation is due to the orthogonality provided by the IFFT processing implemented by the prototype.

Figure 19 depicts the CPU and memory utilization of the prototype for several MCS values. These results compare CPU and memory utilization when the prototype has to multiplex the signal of 1 and 12 vPHYs, respectively. The results in the figure were calculated by averaging CPU and memory usage values sampled every 200 ms during the duration of the experiment, which was set to 60 seconds. Each one of the vPHYs transmit 20 subframes in a row with a 1 ms gap between consecutive transmissions.

As can be observed, the CPU utilization increases as the MCS increases, however, there is no CPU starvation issue. The increase in CPU utilization is mainly 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 consequently demand a lot more of CPU for data processing. For a MCS equal to 28 and 12 vPHYs concurrently transmitting, the CPU utilization is of approximately 450 %, meaning that the processing power of fewer than 5 cores is being employed, leaving the other cores in the idle state for large periods. On the other hand, we see that the memory utilization is practically constant for all MCS values and goes from around 0.5% to 2.2% for 1 and 12 vPHYs, respectively. Therefore, memory utilization is practically independent of the MCS value being used. This is an expected result as all memory being used by the prototype is pre-allocated during its initialization. Therefore, based on the results presented in Figure 19, it can be concluded that the prototype does not exhaust CPU or memory resources as the MCS value increases. These are important results, once they show that given the current server configuration, the prototype can be scaled to support even more vPHYs without exhausting CPU or memory resources.

Next, in Figure 20, we present the assessment of the CPU consumption of independent processing tasks making up the architecture prototype. For this assessment we employ the *callgrind* tool, which is part of the *valgrind* profiler. *Callgrind* is a profiling tool that keeps the call history among functions in a program's run as a call-graph through the use of runtime instrumentation [13, 57]. The figure presents the functions with the highest CPU processing load ($i.e.$, the most representative CPU consumers) for 3 different MCS values and the cases where 1 and 12 vPHYs are instantiated. The setup used for this experiment is the same as the one used during the experiment for CPU and memory profiling.

As can be noticed, channelization presents the highest CPU consumption throughout all test cases. Channelization is a quite heavy processing once it keeps always processing a bandwidth equivalent to the maximum number of configured vPHYs, which in this case is equal to 12, no matter the number of actually instantiated vPHYs. For 1 instantiated vPHY, inverse FFT is the second most consuming task, however, its CPU consumption remains constant for all considered MCS values as it does not depend on the MCS used. In the case that 12 vPHYs are instantiated, we see that memory copy, *memcpy*, increases its load from 15.35% for MCS 0 to approximately 20% for MCS 28. Compared to the 1 vPHY case, it is a drastic increase once it has a maximum CPU load of 7.41% for MCS 28. For the 12 vPHYs case, the inverse FFT is the third most consuming task, ranging from 20.04% for a MCS equal to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2019 doi:10.20944/preprints201911.0180.v1

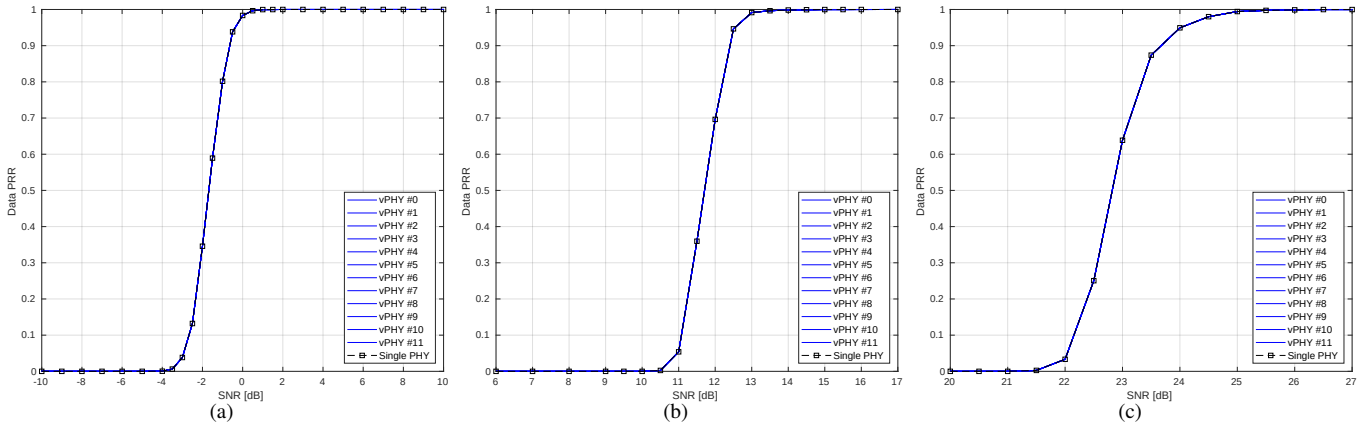

Fig. 18: Data PRR curves for a setup with 12 vPHYs having their signals multiplexed by the prototype and using: (a) MCS0 (QPSK), (b) MCS16 (16QAM), and (c) MCS28 (64QAM).

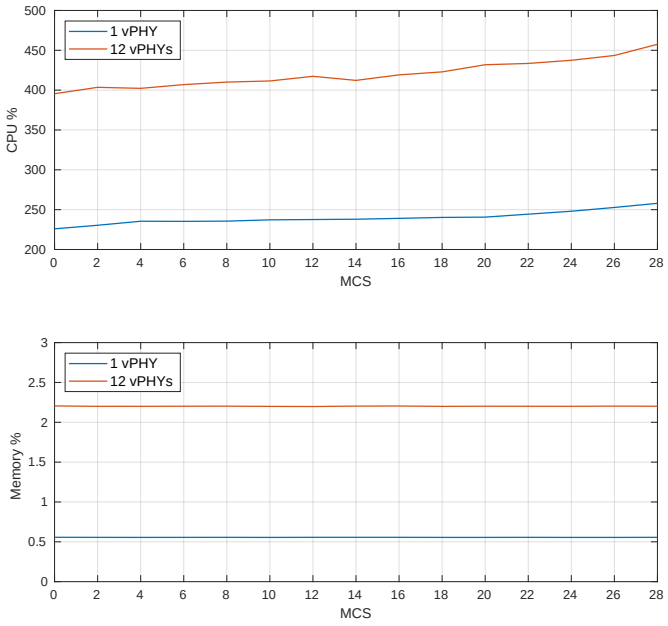

Fig. 19: CPU and memory utilization of the architecture prototype for 1 and 12 vPHYs.

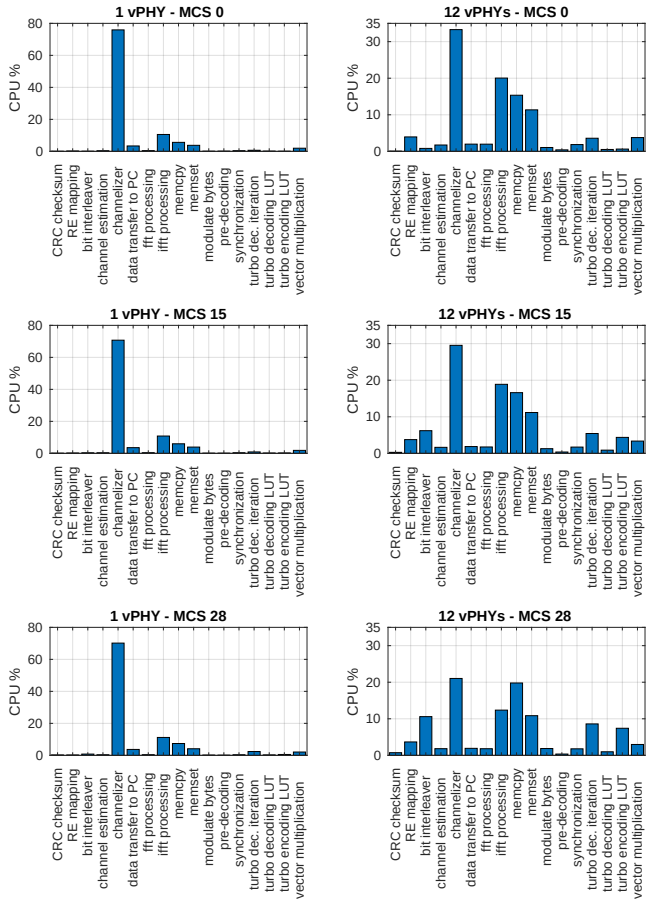

Fig. 20: CPU profiling of individual components of the architecture prototype for 1 and 12 vPHYs.

0 to 12.37% for a MCS equal to 28. We see that, in the second column, the one for 12 instantiated vPHYs, the iFFT processing load gradually decreases while *memcpy* gradually increases its CPU load. On the other hand, we also see that the CPU processing load of the memory set operation, *memset*, remains constant throughout the evaluated MCS values. It is also important to highlight that bit interleaving processing gets heavier as the MCS value increases, consuming approximately 0.81% of CPU time for a MCS equal to 0 and going to 10.59% when the MCS value is made equal to 28.

IX. Conclusion

In this paper, we proposed a wireless spectrum hypervisor architecture that is able to abstracts a radio frequency (RF) front-end into a configurable number of virtual RF front-ends. Our approach was proposed in order to improve spectral efficiency by efficiently using vacant gaps in congested spectrum-bandwidths or employing network densification through infrastructure sharing. We provided a mathematical demonstration on how the proposed approach works and presented several simulation results proving its functionality and efficiency. Additionally, we presented an open-source and free proof of concept prototype of the proposed architecture and several experimental results validating its functionality and showing its performance.

As future work, we highlight the following improvements to the proposed architecture prototype. As we could see, channelization and iFFT ($i.e.$, OFDM modulation) processing tasks consume a lot of CPU time when 12 vPHYs are instantiated, therefore, offloading these tasks to the FPGA can increase both real-time and processing performance. Additionally, memory copy also consumes a lot of CPU time, and therefore, smarter ways of carrying out these copies should be investigated. The proposed architecture supports only multi-carrier based waveforms, however, it would be interesting to add support to other kinds of waveforms, and therefore, another direction would be adding the support to non-multi-carrier waveforms. Finally, the integration of the proposed architecture's prototype with LTE or 5G-like upper layers is a direct sequence of the work presented here and would serve as a demonstration of what can be achieved with the proposed architecture in terms of real deployments.

ACKNOWLEDGEMENT

The project leading to this publication has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732174 (ORCA project).

REFERENCES

[1] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015â"2020*, White Paper, February 2016.

[2] IMT-2020 (5G) Promotion Group, *5G Vision and Requirements*, Technical Report, [Available online] http://www.imt-2020.org.cn/en/documents/download/3, December 2015.

[3] E. Hossain and M. Hasan, *5G cellular: Key enabling technologies and research challenges*, IEEE Transactions on Instrumentation and Measurement, vol. 18, no. 3, pp. 11-21, June 2015.

[4] Emil Bjornson, Jakob Hoydis and Luca Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*, Foundations and Trends in Signal Processing: vol. 11, no. 3-4, pp. 154-655. DOI: 10.1561/2000000093, 2017.

[5] Lopez-Perez, D., M. Ding, H. Claussen, and A. H. Jafari, *Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments*, IEEE Commun. Surveys Tuts. vol. 17, no. 4, pp. 2078-2101, June 2015.

[6] Naga Bhushan, et. al., *Network Densification: The Dominant Theme for Wireless Evolution into 5G*, IEEE Communications Magazine, vol. 52, no. 2, pp. 82-89, February 2014.

[7] Bernard Aboba, *Virtual Access Points*, IEEE document IEEE 802.11-03/154r1, May 2003.

[8] Ratnesh Kumbhkar et. al., *Design and implementation of an underlay control channel for NC-OFDM-based networks*, Conference on Information Science and Systems (CISS), Princeton, NJ, USA, April 2016.

[9] 3GPP, *TS 38.211 - NR; Physical channels and modulation*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network.

[10] Florian Kaltenberger, Guy de Souza, Raymond Knopp, and Hongzhi Wang, *The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap*, Workshop on Smart Antennas (WSA), April 2019.

[11] Ian F. Akyildiz, Pu Wang, and Shih-Chun Lina, *SoftAir: A software-defined networking architecture for 5G wireless systems*, Journal of Computer Networks, vol. 85, pp. 1-18, July 2015.

[12] C. Liang, and R. Yu, *Wireless network virtualization: A survey, some research issues and challenges*, IEEE Communications Surveys Tutorials, vol. 17, no. 1, August 2014.

[13] Felipe Augusto P. de Figueiredo, Xianjun Jiao, Wei Liu, Ruben Mennes, Irfan Jabandzic, and Ingrid Moerman, *A spectrum sharing framework for intelligent next generation wireless networks*, IEEE Access, vol. 6, pp. 60704-60735, October 2018.

[14] Fredric J. Harris, Chris Dick, and Michael Rice, *Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications*, IEEE Transactions on Microwave Theory and Techniques, vol. 51, no. 4, April 2003.

[15] L. Litwin, *FIR and IIR digital filters*, IEEE Potentials, vol. 19, no. 4, October/November 2000.

[16] Gereon Such, Matthias Schraml, and Andreas Knopp, *Frequency Domain Channelizer: A computationally efficient method for non-uniform channelization*, Proceedings of the GNU Radio Conference, vol. 3, no. 1, September 2018.

[17] Jose Mendes, Xianjun Jiao, Andres Garcia-Saavedra, Felipe Huici, and Ingrid Moerman, *Cellular Access Multi-Tenancy through Small Cell Virtualization and Common RF Front-End Sharing*, Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation Characterization (WiNTECH), October 2017.

[18] Maicon Kist, Juergen Rochol, and Luiz A. DaSilva, and Cristiano Bonato Both, *HyDRA: A hypervisor for software-defined radios to enable radio virtualization in mobile networks*, IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), May 2017.

[19] Maicon Kist, Juergen Rochol, Luiz A. DaSilva, and Cristiano Bonato Both, *SDR Virtualization in Future Mobile Networks: Enabling Multi-Programmable Air-Interfaces*, IEEE International Conference on Communications (ICC), January 2018.

[20] FCC, *Report of the Spectrum Efficiency Working Group*, Federal Communications Commission, Report, [Online]. Available: https://transition.fcc.gov/sptf/files/SEWGFinalReport_1.pdf, November 2002.

[21] Ghufran Baig, Dan Alistarh, Thomas Karagiannis, Bozidar Radunovic, Matthew Balkwill, and Lili Qiu, *Towards unlicensed cellular networks in TV white spaces*, Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT'17), December 2017.

[22] Hena Maloku, Zana Limani Fazliu, and Mimoza Ibrani, *A Survey on Coexistence in Heterogeneous Wireless Networks in TV White Spaces*, Wireless Communications and Mobile Computing, vol. 2018, Article ID 7256835, 14 pages, July 2018.

[23] K. Tan, H. Shen, J. Zhang, and Y. Zhang, *Enable flexible spectrum access with spectrum virtualization*, IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 47-58, October 2012.

[24] Felipe Augusto P. de Figueiredo, Nathalia F. T. Aniceto, Jorge Seki, Ingrid Moerman, and Gustavo Fraidenraich, *Comparing f-OFDM and OFDM Performance for MIMO Systems Considering a 5G Scenario*, Preprints 2019, 2019050307 (doi: 10.20944/preprints201905.0307.v2), May 2019.

[25] Pushpendu Kar, and Bhasker Dappuri, *Site Survey and Radio Frequency Planning for the Deployment of Next Generation WLAN*, International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), November 2018.

[26] Amendment of the Commissions Rules with Regard to Commercial Operations in the 3550-3650 MHz Band, Report and Order and Second Further Notice of Proposed Rulemaking, Federal Communications Commission (FCC)Released: April 21, 2015.

[27] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, *Spectrum access system for the citizen broadband radio service*, IEEE Communications Magazine, vol. 53, no. 7, pp. 18-25, July 2015.

[28] Nokia Networks, *Flexi Multiradio BTS RF Module and Remote Radio Head Description- Doc. Num. DN0951745*, November 2014.

[29] CommScope, *RRH-LTE-2600 Wildcat - Remote Radio Head- LTE*, Product Specification, September 2013.

[30] Nutaq, *TitanMIMO-X Technology*, Product Specification, [online available at] https://www.nutaq.com/products/titanmimo/titanmimo-x/technology, accessed on June 2019.

[31] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, *From network sharing to multi-tenancy: The 5G network slice broker*, IEEE Communications Magazine, vol. 54, no. 7, pp. 32-39, July 2016.

[32] 3GPP, *TR 28.801: Study on Management and Orchestration of Network Slicing for Next Generation Network*, Technical Report, 3rd Generation Partnership Project, May 2017.

[33] Lei Xia, Sanjay Kumar, Xue Yang and Praveen Gopalakrishnan, York Liu, Sebastian Schoenberg, and Xingang Guo, *Virtual Wi-Fi: Bring Virtualization from Wired to Wireless*, ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, vol. 46, no. 7, pp. 181-192, March 2011.

[34] Gautam Bhanage, Dipti Vete, and Ivan Seskar, *SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs*,

IEEE Global Telecommunications Conference (GLOBECOM), December 2010.

[35] Joao F. Santos, Maicon Kist, Jonathan van de Belt, Juergen Rochol, and Luiz A. DaSilva, *Towards Enabling RAN as a Service - The Extensible Virtualisation Layer*, IEEE International Conference on Communications (ICC), May 2019.

[36] Ravi Kokku, Rajesh Mahindra, Honghai Zhang, and Sampath Rangarajan, *NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks*, IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1333-1346, October 2012.

[37] Xenofon Foukas, Mahesh K. Marina, and Kimon Kontovasilis, *Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture*, ACM International Conference on Mobile Computing and Networking (MobiCom), pp. 127-140, October 2017.

[38] Joachim Sachs and Stephan Baucke, *Virtual radio: a framework for configurable radio networks*, International Conference on Wireless Internet, pp. 1-7, November 2008.

[39] Yasir Zaki, Liang Zhao, Carmelita Goerg, and Andreas Timm-Giel, *LTE wireless virtualization and spectrum management*, Wireless and Mobile Networking Conference (WMNC), pp. 1-6, October 2010.

[40] Felipe Augusto P. de Figueiredo, *Wireless-Spectrum-Hypervisor*, GitHub Repository, [available online at]: https://github.com/zz4fap/Wireless-Spectrum-Hypervisor, July 2019.

[41] Matteo Frigo and Steven G. Johnson, *The Design and Implementation of FFTW3*, Proceedings of the IEEE Special Issue on Program Generation, Optimization, and Platform Adaptation, vol. 93, no. 2, pp. 216-231, February 2005.

[42] I. Gomez-Miguelez, 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.

[43] Ettus Research LLC, *USRP X Series*, [online]. Available: https://www.ettus.com/product/category/USRP-X-Series

[44] 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/

[45] Ettus Research LLC, *USRP Hardware Driver*, [online]. Available: http://files.ettus.com/manual/

[46] Ettus Research LLC, *Products*, [online]. Available: https://www.ettus.com/product

[47] Google, *protobuf - Protocol Buffers - Google's data interchange format*, [online]. Available: http://code.google.com/p/protobuf/, 2011.

[48] P. Hintjens, *ZeroMQ Messaging for Many Applications*, O'Reilly Media, March 2013.

[49] Javad Abdoli, Ming Jia, and Jianglei Ma, *Filtered OFDM: A new waveform for future wireless systems*, IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), July 2015.

[50] Kyongkuk Cho, Dongweon Yoon, *On the general BER expression of one- and two-dimensional amplitude modulations*, IEEE Transactions on Communications, vol. 50, no. 7, July 2002.

[51] Ettus Research LLC, *CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series only)*, [online]. Available: https://www.ettus.com/product/details/CBX120

[52] Eugene B. Hogenauer, *An economical class of digital filters for decimation and interpolation*, IEEE Transactions on Acoustics, Speech, and Signal Processing. vol. 29, no. 2, pp. 155–162, April 1981.

[53] 3GPP, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 13.0.0 Release 13)*, May 2016.

[54] 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, pp. 45818-45830, April 2019.

[55] 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.

[56] DARPA, *The world's first collaborative machine-intelligence competition to overcome spectrum scarcity*, [online available] https://www.spectrumcollaborationchallenge.com/, 2016.

[57] 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.