

# Online Assessment of Sensing Performance in Experimental Spectrum Sensing Platforms

Stratos Keranidis<sup>⊕</sup>, Virgilios Passas<sup>⊕</sup>, Kostas Chouzos<sup>⊕</sup>, Wei Liu<sup>\*</sup>, Thanasis Korakis<sup>⊕</sup>, Iordanis Koutsopoulos<sup>✉</sup>, Ingrid Moerman<sup>\*</sup> and Leandros Tassiulas<sup>⊕</sup>

<sup>⊕</sup>Department of Computer and Communication Engineering, University of Thessaly, Greece

<sup>+</sup>Centre for Research and Technology Hellas, CERTH, Greece

<sup>\*</sup>Ghent University - iMinds, Ghent, Belgium

<sup>✉</sup>Department of Computer Science, Athens University of Economics and Business, Greece

{efkerani, vipassas, hounos, korakis, jordan, leandros}@uth.gr  
{wei.liu, ingrid.moerman}@intec.ugent.be, jordan@aueb.gr

## ABSTRACT

Dynamic Spectrum Access aims at exploiting underutilized frequency bands towards improving wireless network performance. In this context, spectrum sensing is employed, in order to monitor spectrum occupancy and drive appropriate adaptation decisions. Researchers in the field primarily evaluate proposed sensing approaches in terms of detection accuracy and efficiency of free spectrum utilization. In this work, we focus on online assessment of spectrum occupancy with respect to sensing delay and energy efficiency. Evaluation of spectrum sensing methods with respect to these two metrics is rather lagging in recent experimental developments. The first is related to the latency induced by the spectrum sensing process and its impact on sensing efficiency, which is tightly connected to the resulting performance of the cognitive solution. On the other hand, energy consumption is considered as a crucial issue in all types of wireless communications. Therefore, it is important to extend existing testbed experimentation tools and develop new ones, in order to equip cognitive testbeds with such advanced monitoring capabilities. To this aim, we integrated the proposed monitoring procedure with the experimentation tools of the CREW testbed federation. In order to demonstrate the applicability of our framework, we experimentally validate the performance of four different sensing platforms, as well as a real-time spectrum sensing engine that implements parallel processing on software-defined radios, in terms of the aforementioned metrics.

## Keywords

Spectrum Sensing; Energy Consumption; Sensing Delay; Experimentation; CREW project

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WiTECH'14, September 7, 2014, Maui, Hawaii, USA.

Copyright 2014 ACM 978-1-4503-3072-5/14/09 ...\$15.00.

<http://dx.doi.org/10.1145/2643230.2643232>.

## 1. INTRODUCTION

Cognitive Radio Networking is a rapidly evolving research thrust area in wireless communications nowadays, aspiring to create a major paradigm shift in the wireless landscape through Dynamic Spectrum Access and Management. The dynamic nature of the wireless medium, in accordance with the distributed frequency adaptation mechanisms, as applied in unlicensed bands, necessitate the development of spectrum-agile radios. In this context, spectrum occupancy monitoring is considered as the most important component towards enabling spectrum aware radio operation.

Theoretical developments [1, 2] in the area have proposed various methods for identifying the presence of signal transmissions and evaluating spectrum occupancy. In addition, several realistic experimentation platforms [3, 4, 5, 6] have emerged to enable the practical implementation of spectrum sensing functionality in real systems. Sound experimental validation of proposed spectrum sensing solutions requires experimentation under real world network scale and settings. To this aim, experimental cognitive radio testbeds [7, 8, 9, 10, 11, 12] have been deployed, while tools for orchestration of complex scenarios [13] and collection [14] of experimental data have also been developed. In order to extend cognitive testbeds with more advanced experimentation capabilities, existing tools have to be advanced and new frameworks have to be developed. In this work, we develop an evaluation framework that assesses the performance of cognitive platforms, in terms of two rather important metrics, namely sensing delay and energy efficiency. The former is directly related with the resulting performance of the cognitive solution, while the latter is considered as a crucial issue in all types of wireless communications, both due to restricted battery autonomy of mobile devices, as well as for moving towards “greener” solutions in telecommunications. Evaluation in terms of the aforementioned metrics is automated through a monitoring procedure that has been directly integrated in the experimentation tools of CREW [15] testbeds.

The remainder of this paper is organized as follows. In the following section we present the state-of-the-art spectrum sensing hardware and software platforms, referring also to well-established CR testbed deployments and the frameworks that are used to facilitate experimentation. Section 3 details the characteristics of the considered spectrum sensing devices, while also describing the experimental scenarios, under which performance evaluation of the various platforms

will be based on. In Section 4, we present the various parts that constitute the proposed evaluation framework, while in Section 5 we present extensive experiments that compare the performance of 4 sensing solutions in terms of the considered metrics. Finally, in Section 6 we assess the performance of an external use case that implements real-time spectrum sensing on SDRs and point out the conclusions reached through this work in Section 7.

## 2. RELATED WORK

In an effort to support realistic and large-scale experimentation with spectrum sensing platforms, several CR testbeds have recently been deployed. In the US, existing experimental CR deployments include the Virginia Tech CR Network (VT-CORNET) [7], the ORBIT [8] and Emulab [9] testbeds. In addition, NITOS [10], w-ilab.t [11] and Cortex-lab [12] are the main wireless experimentation testbeds in Europe that are equipped with heterogeneous Software-Defined Radio (SDR) platforms. Among the various developed SDR solutions, USRP [3] is one of the most popular commercial hardware, while more sophisticated solutions, such as the WARP [4] and SORA [5] have also been proposed. On top of the underlying hardware components, software radio architectures have been developed to specify the interconnection and facilitate configurability of components that implement reconfigurable radios, such as GNU Radio [16], the Software Communication Architecture [17] and IRIS [18].

As most experimental testbeds are developed separately from each other, an important issue that experimenters traditionally had to cope with is the scarcity of a common management system, as well as a common experiment description language. However, this issue was overcome with the introduction of OMF, which provides tools for the management and execution of experiments on testbed infrastructures. Nowadays, OMF has been deployed and used on multiple testbeds supporting many different types of technologies. Among the aforementioned CR deployments, ORBIT, NITOS and w-ilab.t testbeds have adopted OMF as their control and management framework. In our previous works, we developed the NITOS EMF [19] and CONCRETE [20] frameworks that extend the core OMF functionalities, by enabling the experimenter to experimentally evaluate the energy efficiency of proposed protocols and to validate the stability of experimental conditions accordingly. An important direction that we are also currently investigating is the direct integration of the IRIS software radio architecture with the OMF framework.

## 3. EVALUATION OF CREW TESTBED SENSING DEVICES

In this work, we exploit the advanced spectrum sensing platforms that are provided by the CREW federated testbed platform, which facilitates experimentally-driven research on advanced spectrum sensing, cognitive radio and cognitive networking strategies. Among the 5 individual wireless testbeds that constitute CREW, we decided to use the w-iLab.t indoor testbed, as it offers all the required hardware and software components for the development of the proposed framework. More specifically, w-iLab.t features several sensing devices that span from commercial sensor and Wi-Fi nodes, to SDR platforms and device prototypes. Another basic characteristic that motivated the usage of the

w-ilab.t testbed, was the adoption of OMF as its testbed control and management framework, which fact enabled us to build the proposed framework as a plug-in compatible with the well-adopted OMF. In the rest of this section, we describe in detail the capabilities of the considered w-ilab.t sensing devices, while we also describe the reference scenarios, under which performance evaluation of the various platforms would be based on.

### 3.1 Characteristics of Sensing Devices

In our experiments, we consider the well-established commercial SDR USRP N210 Networked Series [21], which is a commercial SDR platform that utilizes the general purpose processor of a connected host machine for measurement processing and due to its applicability has gained widespread usage. In addition, we consider the embedded version USRP E110 [22], which allows standalone operation without requiring the use of a host machine, thus allowing us to evaluate how its limited internal processing capabilities impact sensing performance. The next device under consideration is the prototype imec Sensing Engine (SE) [6], which is a high-end prototype reconfigurable radio that is designed to process samples real-time. Finally, we also decided to experiment with the Wi-Fi compliant Atheros AR9380 [23] chipset, as this device is able to operate as a spectrum scanner as well. By including the AR9380 device in our experiments, we can investigate how well the hardware assisted FFT processing capabilities of this chipset perform, contrary to the rest research oriented sensing devices. More details about the capabilities of the devices under investigation follow below.

#### 3.1.1 USRP N210

The USRP N210 consists of two parts, a fixed motherboard and a plug-in daughterboard. The motherboard contains ADC/DAC, an FPGA for digital down conversion with programmable decimation rate and a Gigabit Ethernet (GbE) interface for communication with the host machine. The USRP Hardware Driver (UHD) is used to control the USRP hardware and also to transmit and receive data over the GbE interface. We attach the USRP N210 with the XCVR 2450 daughterboard, which provides basic RF front-end functionality in the 2.4 GHz and 5.9 GHz bands. The channel switching delay for the XCVR 2450 daughterboard was measured and found to be 50 ms. Although the ADC is able to sample at the 100 MHz rate, the GbE interface limits the maximum streaming bandwidth to 25 MSps. In order to provide real-time sensing capability, the host machine should be able to achieve sufficient amount of parallel processing and thus this feature is host machine dependent.

#### 3.1.2 USRP E110

The USRP E110 [3] series combines a flexible RF frontend, FPGA and an OMAP 3, which includes an ARM Cortex-A8 and a C64 DSP. The extra features differentiate the E110 from the N210, as the former does not require the use of a host pc for measurement processing and thus allows standalone operation for embedded applications, or applications that do not require the full processing power of a commodity CPU. The USRP E110 is also programmed through the USRP Hardware Driver (UHD) and attached with the XCVR 2450 daughterboard is able to operate in the 2.4 GHz and 5.9 GHz bands. The channel switching delay for this setup also equals 50 ms. The attached ADC is

able to sample at the 64 MHz rate, but the FPGA interface provides for a maximum total bandwidth of 10 MSps. However, this does not guarantee that the embedded processor will be able to process samples at this rate. Our own tests have shown that up to the 5 MSps bandwidth configuration the embedded processor is able to provide stable operation. As a result the ability to provide for real-time sensing depends on the attached host machine.

### 3.1.3 *imec SE*

The prototype *imec SE* [4] is a reconfigurable radio and consists of two core components: an analogue RF front-end SCALDIO (SCALable raADIO) and a DIgital front-end for Sensing (DIFFS). Both these ICs are low-power and flexible and targeted towards implementing a cognitive radio as a mobile device. DIFFS features an FFT accelerator core, enabling the engine to perform spectral analysis. The receiver RF operating frequency is programmable from 0.1 to 6 GHz and the channel bandwidth is programmable between 1 and 40 MHz. The architecture of the SCALDIO chip allows us to already handle the reprogramming for the next frequency band during the current capture. Hence the reconfiguration time is reduced to zero and only the settling time of 50  $\mu$ s needs to be considered. Both the DIFFS processor and the ADC are running at 40 MHz. However, in our configuration the ADC is down-sampled to 20 MHz, resulting in the available channel bandwidth of 20 MHz. Hardware-wise the *imec SE* was designed to be able to process samples real-time. As the measurement collection and processing procedures can run in parallel, we just have to make sure that processing finishes before new data is available, in order to provide for real-time sensing. The *imec SE* firmware that is used in our experiments is able to perform all required calculations, before the next vector of samples that needs to be processed is available and thus is capable of real-time sensing.

### 3.1.4 *Atheros AR9380*

The *Atheros AR9380* is a single-chip, dual-band (2.4/5 GHz), 802.11n compatible chipset that supports up to 3x3 MIMO transmissions, offering both high throughput performance, along with low power consumption in every operational state. The special characteristic of AR9380 is its ability to support spectrum sensing capabilities. The Ath9k [24] driver enables the user to configure the card, in order to sense the spectrum over a specified set of channels and derive the Power Spectral Density with an FFT resolution of 56 bins. The 56 bins correspond to the 52 usable plus the 4 pilot sub-carriers that are supported by the 802.11n protocol in the case that the 20 MHz bandwidth configuration is applied. The FFT operation is executed on the chipset and scanning results that correspond to 4 $\mu$ s long snapshots of the spectrum are reported per frequency at the fixed period of 55 ms. The channel switching delay was measured at 1 ms for in-band and 2 ms for out-band transitions. The channel switching overhead of the *imec SE* and AR9380 is obtained through the device specifications that are provided by *imec* and *Atheros* accordingly, while such values are not officially reported for the USRP devices and are derived through our own measurements. We experimentally verified that maximum configurable sampling rate equals 100 KSps. The driver provides an interface to configure both the actual sampling rate through adaptation of the decimation level at 255 levels, as well as the number of result sets that will be

Sensing device	Host requirement	Channel bandwidth (MSps)	Channel switching delay	Real-time sensing
USRP N210	Yes	25	50 ms	Host-dependent
USRP E110	No	5	50 ms	Host-dependent
imec SE	No	20	50 $\mu$ s	Yes
AR9380	Yes	20	1-2 ms	No

**Table 1: Hardware characteristics of sensing devices**

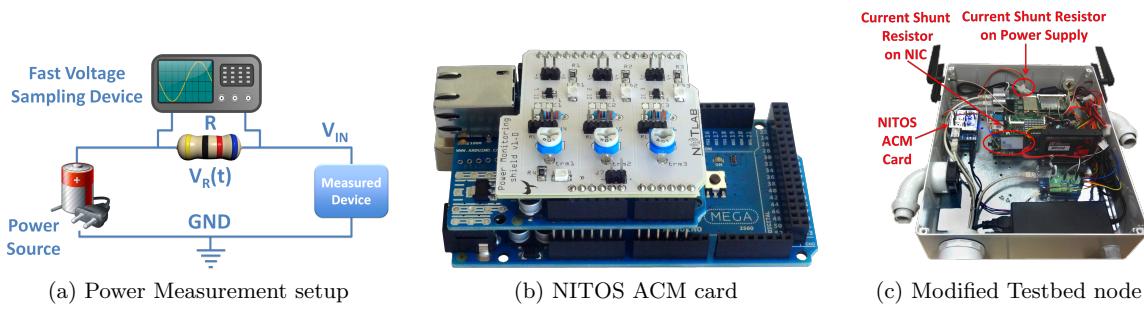
reported within the 55 ms interval. Although the AR9380 is able to sample at a sufficient sampling rate, this device is not able to provide for real-time sensing, as continuous sampling can only last up to 55 ms, upon which interval the device goes back to regular Wi-Fi operation.

## 3.2 Experimental Scenarios

The various hardware platforms are used in parallel to detect a signal generated, under various settings, by a pair of w-ilab.t testbed nodes. The test signal consists of a bunch of 802.11 frames, as this is the most common signal transmitted in the 2.4 GHz and 5GHz ISM bands that can be sensed by all the different spectrum sensing platforms. As the latest widely adopted version of the IEEE 802.11 standard is the 802.11n version, the test signal consists of 802.11n transmissions that occupy the medium for significantly lower duration compared to legacy 802.11 transmissions, due to the increased PHY-layer transmission rates that 802.11n offers. Considering transmissions of typical MPDU frames of 1534 bytes, frame transmissions duration approximates 227  $\mu$ s and 63  $\mu$ s at the highest PHY-layer rates of 54 Mbps and 195 Mbps that are supported by the 802.11a/g and 802.11n protocols at the bandwidth configuration of 20 MHz.

Having properly investigated the operation and capabilities of each different device, as listed in Table 1, we concluded that the only hardware dependent specifications that could restrict the design of the experimental scenarios would be related with the *imec SE* and the *Atheros 9380* devices. The USRP devices are fully configured through software, thus generating no restrictions. Considering the real-time sensing capability offered by default by the *imec SE*, we decided to build a proper scenario which would allow for a fair comparison of the other devices with the high-end *imec SE*. Based on the device specifications that were analyzed in the preceding section, we know that the only hardware dependent restrictions are the 20 MHz sampling rate of the *imec SE* and its 128 bin size of FFT calculations. When the *imec SE* is sampling at this rate, it requires 6.4  $\mu$ s, in order to gather 128 samples that are further fed to the on-chip FFT accelerator core for FFT processing that finally provides the PSD results. As the interval of 6.4  $\mu$ s is extremely short, compared to the shortest duration event of 63  $\mu$ s, we further decided to combine several FFT measurements, before arriving at a final result. We use Max-Hold filtering over 10 consecutive FFT operations, which results in the total sensing time of 64  $\mu$ s, within which interval we are able to decide about the presence of 802.11n transmissions.

Considering the fact that the aforementioned sensing devices feature varying sensing capabilities, we will test them under the reference scenario, to comparatively evaluate their performance in terms of sensing delay and energy efficiency. In order to provide for a proper evaluation and comparison setup, we decided to configure all the devices to sense for the same duration of 64  $\mu$ s, which we denote by Measurement Collection Duration (MCD). Having specified a com-



**Figure 1: Power consumption measurement methodology and introduced hardware**

Sensing device	FFT resolution	Result sets	Collected samples
USRP N210	1024	1	1600
USRP E110	256	1	320
imec SE	128	10	1280
AR9380	56	6	336

**Table 2: Sensing characteristics of sensing devices in the considered scenarios**

mon MCD, the different devices will sense the medium for the same duration and collect a different number of samples, as specified by their sampling rate configurations. In the next phase, a common measurement-processing program will process the different number of collected samples and report the time required for processing.

Moreover, we also consider a second scenario, where the targeted band is 80 MHz wide, so that all devices will have to perform channel switching to provide for proper monitoring of the whole band. In this second scenario, the total sensing delay of all devices will further vary due to the impact of the channel switching overhead. Various characteristics as configured for each device in the considered experimental scenarios are listed in Table 2.

## 4. EVALUATION FRAMEWORK

In order to provide for a fair performance evaluation, between the 4 considered devices, we configure them under their operational limits to sense the medium and assess the Power Spectral Density (PSD) on the 2.4 GHz band. In parallel with the spectrum sensing operation, we evaluate the power consumption and the distribution of sensing delay per device. In the rest of this section, we describe in detail the 3 steps that constitute our evaluation framework.

### 4.1 Power Spectral Density

Spectrum sensing efficiency evaluation can be based on the quality of PSD characterisation. The PSD characterises the distribution of energy variations and more specifically quantifies the amount of energy per certain frequency interval (dBm/Hz). PSD computations can be obtained directly through FFT transformation of power results that have been collected over time. Frequency resolution is directly dependent on the sampling frequency and the FFT size. More specifically, for a fixed sampling interval, the frequency resolution can be improved by increasing the FFT size. However, the maximum configurable FFT-size depends on the maximum sampling rate of each device and is even hardware dependent for devices that perform FFT calcula-

tions in hardware. To conclude, an inherent tradeoff exists between the configuration of PSD parameters and the obtained resolution, which we aim to evaluate jointly with the proposed metrics of Sensing Delay and Power Consumption.

### 4.2 Power Consumption

Next, we detail the measurement approach has been followed during the evaluation of the different sensing devices in terms of power consumption. In order to accurately measure the instantaneous power consumption, we follow a widely adopted power measurement procedure, which requires the placement of a high-precision, low impedance current-shunt resistor ( $R$ ) of a known resistance value, in series with the power source and the power supply pin of the device to be measured. The exact measurement setup described above is presented in Fig. 1(a). By consistently measuring the voltage ( $V_R(t)$ ) across the current-shunt resistor through proper voltage metering equipment, we are able to extract the instantaneous current draw of the device, based on Ohm's law. The instantaneous power consumption can be calculated as the product of the input voltage  $V_{IN}$  and the measured current draw:

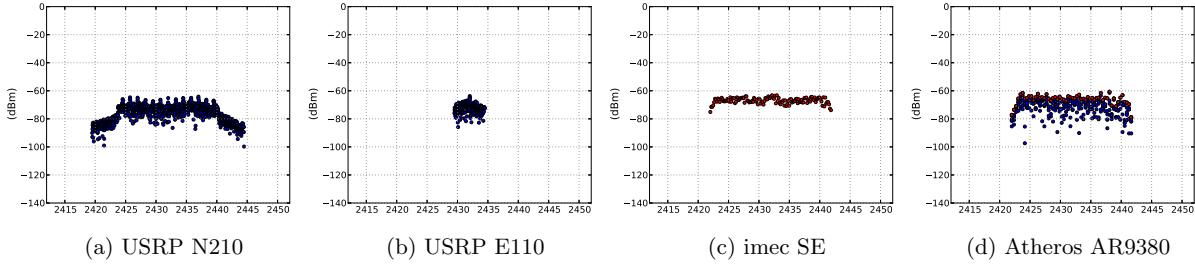
$$P(t) = V_{IN} \frac{V_R(t)}{R} \quad (1)$$

Estimation of the total energy consumption during a specific experiment, necessitates the accurate sampling of the instantaneous power consumption during the total experiment duration. Total energy consumption can be calculated as the integral of the power consumption over the specified duration ( $Dt = t_1 - t_0$ ), as follows:

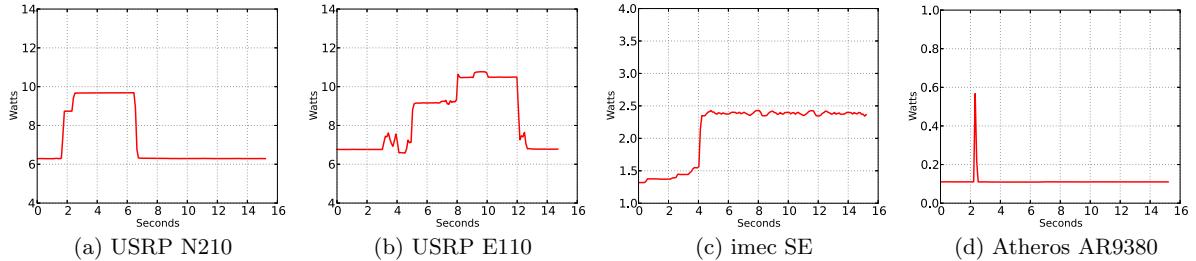
$$E(Dt) = \frac{V_{IN}}{R} \int_{t_0}^{t_1} V_R(t) dt \quad (2)$$

However it should be made clear that through the voltage sampling equipment, only a finite number of samples of  $V_R(\cdot)$  are acquired over  $[t_0, t_1]$  at discrete time instances.

In order to accurately measure the voltage drop across the resistors that are attached in series with the power supply of each device, we use the NITOS ACM card that was introduced in our previous work [19]. The developed card, which is presented in Fig. 1(b), supports the high sampling rate of 63 KHz and features up to three input channels, thus providing for online power consumption monitoring at both the sensing device and the attached host machine, in a joint way. In Fig. 1(c), we demonstrate the hardware setup used in measuring the consumption of the AR9380 and the host.



**Figure 2: Power Spectral Density Evaluation - Scenario 1**



**Figure 3: Power Consumption Evaluation - Scenario 1**

### 4.3 Sensing Delay Distribution

Overall sensing delay is composed of various parts that result due to different components of each device. As soon as all devices are configured properly, the first phase of actual sampling is executed. This duration does not directly correspond to the time of channel monitoring, but may also include time spent due to communication with specific device components, such as the transferring part. Third, as the measurements acquisition has been completed, the next step is to process the collected measurements. Duration of this task is mainly dependent on the processing capabilities of each sensing solution. Finally, in the case that the dimension of multiple channels is considered, the overall sensing delay may also be affected by the overhead induced by the channel switching process.

Ideally, the exact amount of time that is spent in each subprocess should be calculated for each device. We managed to derive the Sensing Delay Distribution for the USRP devices, by incorporating high precision timer functions at appropriate parts of the UHD source code. However, such low level measurements cannot be reported for the imec SE and the AR9380 devices, where the different subprocesses run in hardware. For these devices, we are only able to measure the total induced sensing delay and compare it with the duration of the actual sampling phase.

## 5. EXPERIMENTAL EVALUATION

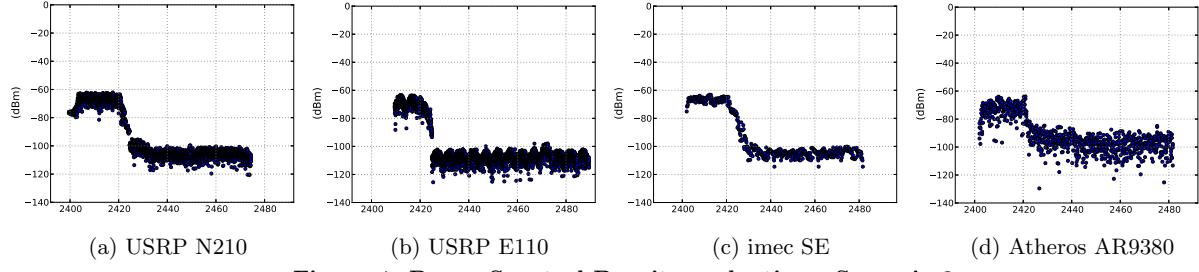
### 5.1 Power Spectral Density

Having configured the experimental scenarios and fixed the MCD interval, we next configure the transmitter 802.11n node to constantly transmit saturated traffic on channel 5 of the 2.4 GHz band and more specifically on 2432 MHz. In the first scenario, each device collects the fixed number of samples, as specified in Table 2. We then proceed by processing data produced by the USRP devices through FFT software, while the imec SE and AR9380 perform these calculations in dedicated hardware. For the USRPs, we use the number

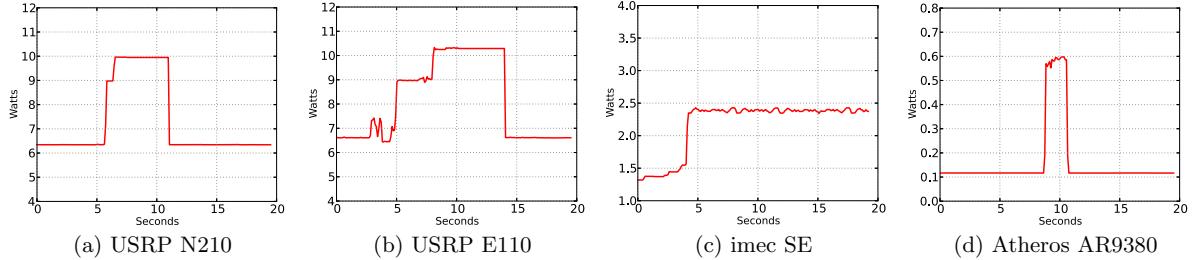
of samples that are closer to the next power of 2 and more specifically process 1024 samples to produce 1024-bin FFT results for the USRP N210 and 256-bin FFT for the USRP E110. As the number of samples that are collected from the imec SE and AR9380 are more than required for a single FFT calculation, the imec SE reports the maximum values that have been collected over 10 result sets, while for the AR9380 6 different result sets are generated.

Indicative screenshots representing the PSD evaluation of the channel as monitored by each different device for the first scenario are plotted in Figures 2(a), 2(b), 2(c) and 2(d) accordingly. We clearly observe how the maximum bandwidth and FFT-bin size specifications of each device affect the sensing efficiency. The imec SE along with the Atheros AR9380 device detect that the 20 MHz bandwidth are fully utilized, while the limited USRP E110 is able to monitor only 5 MHz of bandwidth. On the other hand, the USRP N210 is able to detect the drop at frequencies that exceed the central frequency by 10 MHz and thus is the only device able to characterize that the Wi-Fi transmission has the central frequency of 2432 MHz (channel 5). Moreover, taking into account the increased 1024 bin FFT resolution (24.41 kHz/bin) that is generated by the USRP N210 and plotted in Fig. 2(a), we observe that the PSD distribution is presented at the OFDM subcarrier level, among the 52 subcarriers that the 802.11n compatible transmission uses.

Figures 4(a), 4(b), 4(c) and 4(d) plot the PSD, as evaluated by each device in the second scenario. We observe that all devices detect channel 1 (freq. 2412 MHz) to be busy, while the rest part of the spectrum in the 2.4 GHz is characterized by Energy levels that are close to the Noise level (-100 dBm). In addition, the amount of channel switches that each device needs to perform in order to sense the whole 80 MHz band can also be extracted from the corresponding figures. Based on the bandwidth specification of each device, the USRP N210 (25 MHz) needs to switch its operational frequency 3 times, the USRP E110 (5 MHz) 15 times, the imec SE (20 MHz) 3 times and the AR9380 (20 MHz) 3 times, to sense the whole 80 MHz wide 2.4 GHz band.



**Figure 4: Power Spectral Density evaluation - Scenario 2**



**Figure 5: Power Consumption evaluation - Scenario 2**

## 5.2 Power Consumption

### 5.2.1 Consumption during Measurement Collection

In parallel with the PSD evaluation, we attach dedicated NITOS ACM cards to each one of the devices, in order to measure the energy consumption during the spectrum sensing procedure and plot the collected results in Figures 3 and 5 for each scenario. As soon as the devices have been properly configured and the actual sensing procedure takes place, the power consumption of each device reaches a stable level, which equals 9.68 W for the USRP N210, 10.54 W for the USRP E110, 2.39W for the imec SE and 0.57 W for the AR9380 accordingly. In general, we remark that the commercial Atheros AR9380 Wi-Fi chipset presents the best energy efficiency, while the embedded imec SE is also characterized by a low power consumption profile, in comparison with the energy harvesting research oriented USRP devices. In the second scenario the power consumption presents exactly the same results, however the duration of the sensing, transferring and processing procedures is prolonged and thus the energy consumption is increased.

### 5.2.2 Consumption during Measurement Processing

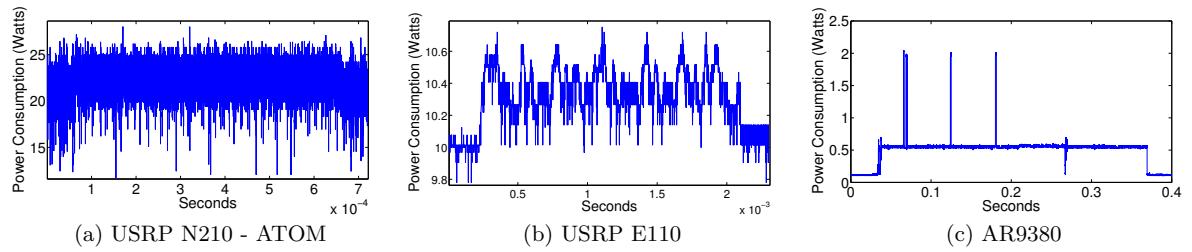
Both the imec SE and the AR9380 devices process measurements in hardware and as a result isolation of consumption due to processing cannot be obtained. Regarding the consumption of the USRP devices N210 and E110, we measure the consumption of the ATOM-based (Intel Atom D525, 1.8 GHz) host machine and the embedded processor, during measurements processing. Figures 6(a) and 6(b) depict the power consumption increase that the processing results in, which equals approximately 1.35 W for the ATOM and 0.4 W for the E110. In addition, Fig. 6(c) plots the power consumption of the AR9380 in the second scenario, where we observe spikes that correspond to the channel switching operation that lasts 1.2 ms and verifies the reported in the official product specifications channel switching delay.

## 5.3 Sensing Delay Distribution

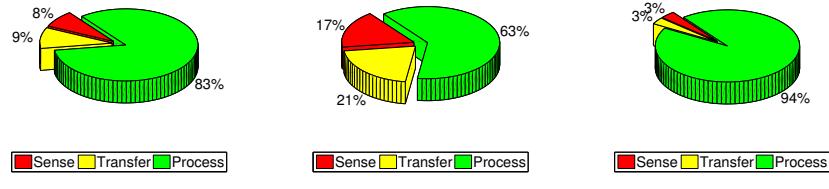
As we are not able to estimate the distribution of Sensing Delay for the imec SE and the AR9380 devices, here we present results that were obtained while experimenting with the USRP devices. Having configured the MCD interval of 64  $\mu$ s, the 1600 samples that have been collected by the USRP N210 require 79  $\mu$ s to be transferred over the GbE, while the USRP E110 requires 58  $\mu$ s to transfer the 320 samples from the FPGA to the RAM memory of the embedded machine. Regarding the processing duration, the 1024-bin FFT operation is executed in 701  $\mu$ s on the ATOM based host machine and 240  $\mu$ s to run on an i7-based (Intel i7-2600, 3.40 GHz). Finally, the ARM Cortex-A8 of the USRP E110 requires 1800  $\mu$ s to execute the 256-bin FFT operation. Figures 7(a), 7(b) and 7(c) plot the distribution of total Sensing Delay in the three different setups. We clearly notice that the processing process dominates the total delay, as it exceeds the duration of the other two phases and thus real-time sensing cannot be performed under these configurations. In USRP N210, data transfer happens in parallel with the sensing, as two different hardware modules inside the FPGA work in parallel. Hence, the reason that consists our solution incapable of performing real-time sensing is the high processing delay. However, as experimentation with longer MCD intervals, showed that the processing duration becomes significantly lower, we expect that real-time sensing can be performed in systems able to achieve significant amount of parallel processing.

## 6. EXTERNAL CASE EVALUATION

In this section, we assess the performance of an external use case that implements real-time spectrum sensing on SDRs, through a multi-threaded sensing engine software that achieves parallel processing. The sensing engine under evaluation was presented in [25] and is composed of two main threads, among which the first one is responsible for collecting samples from the USRP, while the second thread is responsible for processing the acquired samples. The sample-processing procedure again generates sev-



**Figure 6: Power Consumption during processing of measurements collected over the MCD interval**



**Figure 7: Sensing Delay distribution**

Sensing Delay Distribution	Channel Bandwidth			
	5 MHz	7 MHz	9 MHz	25 MHz
Collected Samples	125K	175K	225K	625K
Sensing	25 ms	25 ms	25 ms	25 ms
Transferring	4ms	5.6 ms	7.2 ms	20 ms
ATOM Processing	13.7 ms	18 ms	22.3 ms	-
i7 Processing	4.6 ms	6.2 ms	7.3 ms	17.3 ms

**Table 3: Sensing Delay Distribution across varying bandwidth configurations for the USRP N210 when attached with ATOM and i7-based setup**

eral sub-threads to process the incoming samples in parallel, by calculating the FFT-based PSD and the energy for specified channels. Once all sub-threads finish processing, they terminate and the original sample-processing thread outputs the result. We configure the sensing engine to use 16 processing threads that process in parallel the samples received from the sample-collecting thread at the highest configurable sampling rate of the USRP N210 (25 MSps). In cases that the sample-processing thread can no longer follow the sample-collecting thread, the software detects the overflow of samples and terminates.

The sensing engine under evaluation has been shown (in [25]) to sense in real-time when running on a hexacore server machine. In this experiment, we use the developed framework to evaluate its performance under the dual-core ATOM-based and the quad-core i7-based setups, trying to investigate whether these setups are able to provide for real-time spectrum sensing. We configure the sensing engine to sense Ch. 5 (2432 MHz) of the 2.4 GHz band and detect the presence of 802.11 Beacon frames that are transmitted every 100 ms, in order to validate the performance of the channel occupancy monitoring procedure. According to the IEEE 802.11 standard [26], Beacon frames are 134 bytes long and transmitted at 6 Mbps bitrate, thus their transmission requires approximately 179  $\mu$ s. We also configure the MCD interval at 25 ms and are able to take 4 decisions about channel occupancy within the 100 ms interval.

While experimenting with the ATOM-based setup, overflow of samples was detected when the channel bandwidth was configured above the 9 MHz. On the other hand, the

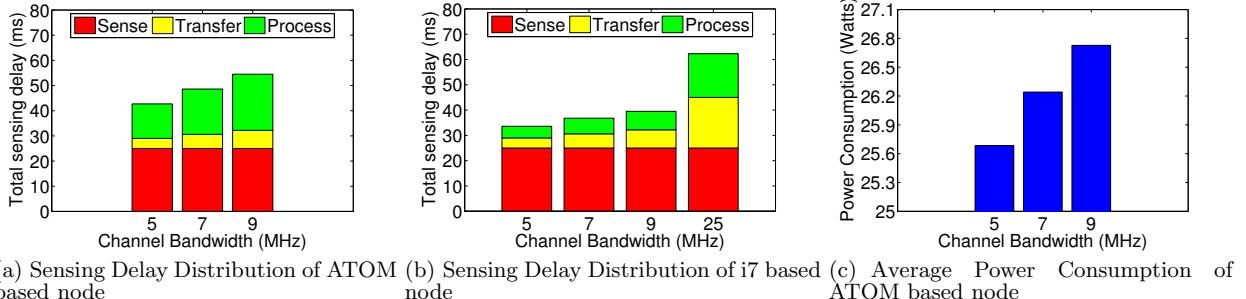
i7-based machine did not experience any overflows even in the 25 MHz bandwidth configuration. Figures 8(a) and 8(b) present the outcomes of our evaluation in terms of Sensing Delay Distribution, across varying channel bandwidth configurations. The obtained measurements are detailed in Table 3 and show that both systems are capable of sensing the spectrum in real time up to the 9 MHz and 25 MHz bandwidth configuration accordingly. Under this setup, the multithreaded system implementation is able to execute the processing phase in parallel with the sample acquisition phase, as long as the processing delay is constantly lower than the sensing delay. In Fig. 8(c), we also plot the average power consumption that the ATOM-based setup results in, under the three configured bandwidth values. We clearly observe that the increasing amount of data that are being processed impacts instantaneous power consumption and more specifically remark that the 5 MHz, 7 MHz and 9 MHz, resulted in the 25.684 W, 26.2413 W and 26.7276 W of average power consumption accordingly. In order to demonstrate the seamless capturing of 802.11 beacons, we plot in Figures 9(a), 9(b) and 9(c) spectrograms as resulting under each setup accordingly. During 700 ms, seven beacon signals are transmitted and they are all captured, clearly demonstrating the advantage of seamless capturing.

## 7. CONCLUSIONS

In this paper we demonstrate a framework that enables evaluation of cognitive devices, in terms of sensing delay and resulting energy consumption. We present a monitoring procedure that has been directly integrated in the experimentation tools of the CREW facilities and demonstrate how it aids in the online evaluation of 4 different cognitive platforms in terms of the aforementioned metrics. We also use the developed framework to experimentally validate a real-time spectrum sensing solution that implements parallel processing on software-defined radios.

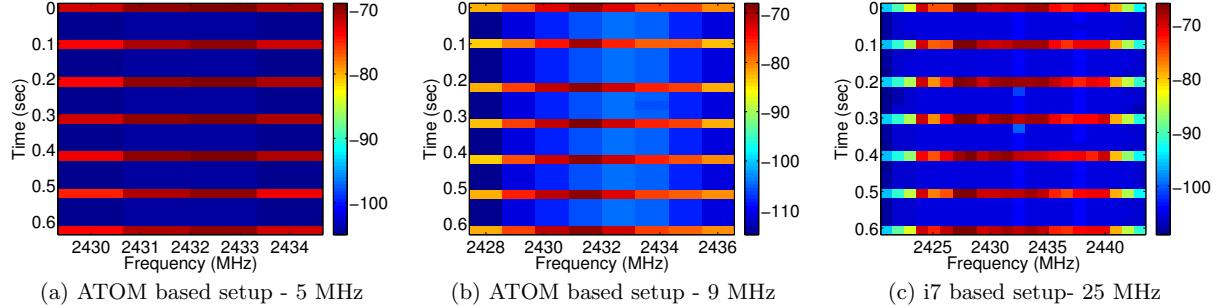
## 8. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme



(a) Sensing Delay Distribution of ATOM based node (b) Sensing Delay Distribution of i7 based node (c) Average Power Consumption of ATOM based node

Figure 8: External use case evaluation across varying bandwidth configurations



(a) ATOM based setup - 5 MHz (b) ATOM based setup - 9 MHz (c) i7 based setup- 25 MHz

Figure 9: Spectrogram of captured 802.11 Beacon frames across varying bandwidth configurations

(FP7/2007-2013) under grant agreement n. 258301 (CREW project).

## 9. REFERENCES

- [1] A. Ghasemi and E.S. Sousa. Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs. *Communications Magazine, IEEE*, 46(4):32–39, April 2008.
- [2] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys Tutorials, IEEE*, 11(1):116–130, First 2009.
- [3] “Universal Software Radio Peripheral (USRP)”, <http://www.ni.com/usrp/>.
- [4] “Wireless Open-Access Research Platform (WARP)”, <http://warp.rice.edu/>.
- [5] “Microsoft Research Software Radio (SORA)”, <http://goo.gl/KoYkhQ>.
- [6] S. Pollin, L. Hollevoet, P. Van Wesemael, M. Desmet, A. Bourdoux, E. Lopez, F. Naessens, P. Raghavan, V. Derudder, S. Dupont, and A. Dejonghe. An integrated reconfigurable engine for multi-purpose sensing up to 6 GHz. In *Proceedings of DySPAN*, May 2011.
- [7] “VT-CORNET”, <http://goo.gl/UTojEZ>.
- [8] “ORBIT Testbed”, <http://www.orbit-lab.org/>.
- [9] “Emulab Testbed”, <http://www.emulab.net>.
- [10] “NITOS Wireless Testbed”, <http://nitlab.inf.uth.gr>.
- [11] “w-ilab.t Testbed”, <http://www.crew-project.eu/wilabt>.
- [12] “CORTEX-lab”, <http://www.cortexlab.fr/>.
- [13] “OMF-cOntrol & Management Framework”, <http://omf.mytestbed.net/>.
- [14] “OML Measurement Library”, <http://mytestbed.net/projects/oml/wiki/>.
- [15] “CREW Project”, <http://www.crew-project.eu/>.
- [16] “GNURadio”, <http://gnuradio.org/>.
- [17] “Software Communications Architecture (SCA)”, <http://goo.gl/EjMIQ7>.
- [18] “Implementing Radio in Software (IRIS)”, <http://irissoftwareradio.wordpress.com/>.
- [19] S. Keranidis, G. Kazdaridis, V. Passas, T. Korakis, I. Koutsopoulos, and L. Tassiulas. Online Energy Consumption Monitoring of Wireless Testbed Infrastructure Through the NITOS EMF Framework. In *Proceedings of ACM WiNTECH*, 2013.
- [20] S. Keranidis, W. Liu, M. Mehari, P. Becue, S. Bouckaert, I. Moerman, T. Korakis, I. Koutsopoulos, and L. Tassiulas. CONCRETE: A benchmarking framework to CONtrol and Classify REpeatable Testbed Experiments. In *FIRE Engineering Workshop*, 2012.
- [21] “USRP N210 Networked Series”, <https://www.ettus.com/product/details/UN210-KIT>.
- [22] “USRP E110 Embedded Series”, <https://www.ettus.com/product/details/UE110-KIT>.
- [23] “Atheros AR9380 Chipset”, <http://goo.gl/51Mg7>.
- [24] “Ath9k wireless driver”, <http://wireless.kernel.org/en/users/Drivers/ath9k>.
- [25] Wei Liu, Daan Pareit, EliDe Poorter, and Ingrid Moerman. Advanced spectrum sensing with parallel processing based on software-defined radio. *EURASIP Journal on Wireless Communications and Networking*, 2013(1), 2013.
- [26] IEEE 802.11-2007 Wireless LAN Medium Access Control and Physical Layers Specifications. IEEE 802.11-2007 wireless lan medium access control and physical layers specifications, 2007.