CorteXlab: A Cognitive Radio Testbed for Reproducible Experiments

Leonardo S. Cardoso, Abdelbassat Massouri, Benjamin Guillon, Florin Hutu, Guillaume Villemaud, Tanguy Risset, Jean-Marie Gorce

CorteXlab: A Cognitive Radio Testbed for Reproducible Experiments

Leonardo S. Cardoso, Abdelbassat Massouri, Benjamin Guillon Florin Hutu, Guillaume Villemaud, Tanguy Risset, Jean-Marie Gorce
Université de Lyon, INRIA,
INSA-Lyon, CITI-INRIA, F-69621, Villeurbanne, France
Email: leonardo.cardoso@insa-lyon.fr

***Abstract*—The efficiency and potential gain of cognitive radio and more generally opportunistic cooperative communications have been already demonstrated from a theoretical point of view and supported by various simulation results. Beyond these promising results, several questions remain open from a practical point of view. Addressing these issues is not straightforward because deploying complex heterogeneous systems for cooperative scenarios is tedious, time consuming and hardly reproducible. We propose to make a step in this direction by offering a new experimental facility, called CorteXlab, that allows complex multi-node cognitive radio scenarios deployment from anywhere in the world. Our objective is neither to design new software defined radio (SDR) nodes nor to propose a new software framework, but rather to provide a comprehensive access to a large set of high performance SDR nodes. The CorteXlab facility offers a 167 m^2 electromagnetically (EM) shielded room and integrates a set of 24 universal software radio peripherals (USRPs) from National Instruments, 18 PicoSDR nodes from Nutaq and 42 IoT-Lab wireless sensor nodes from Hikob. CorteXlab is built upon the foundations of the SensLAB testbed and also exploits the free and open-source toolkit GNU Radio. Automation in scenario deployment, experiment start, stop and results collection is performed by an experiment controller, called Minus. CorteXlab is in its final stages of development and is already capable of running specific test scenarios. In this contribution, we show that CorteXlab is able to easily cope with the usual issues faced by other testbeds providing a reproducible experiment environment for CR experimentation.**

I. Introduction

Cognitive radio (CR) has a potential to achieve high spectral efficiencies through a better re-use of radio resources. CR associate wide-band and dynamic RF front-ends, opportunistic and distributed schedulers and software radio programming to exploit free bands on the fly. The efficiency of these systems may be further increased by providing these nodes with further capabilities to cooperate. Cooperation may take various ways such as mutual packet relaying, cooperative sensing, distributed antennas beamforming, interference alignment,...

However, many of these scenarios when evaluated from a theoretical point of view rely on some questionable assumptions such as perfect synchronisation, perfect timing, zero-delay processing, zero-delay reconfiguration,... Several of such technical issues are easier to be dealt experimentally, rather than from an analytical or simulation point of view. Experimental approaches allow to enhance where are the true problems and what are the real performance of different techniques. In the last ten years, experimenting with SDRs has been made possible by the commoditization of radio platforms and the availability of software radio toolkits, such as the universal software radio peripheral (USRP) and GNU Radio, respectively. They have popularised – and even democratised – the experimentation on radio, allowing researchers, engineers and amateurs to build and test radio transceiver technologies, ranging from the most simple (e.g. [1]) to the state-of-the-art in radio communications (e.g. [2]). However, while the deployment of an experimental scenario over a couple of radio nodes is manageable, it becomes a daunting task when many nodes are involved. This becomes especially hard when multiple types of radio nodes, with different capabilities and characteristics integrate a heterogeneous testbed. Indeed, the testbed operator must sequentially log on each host computer to deploy the code that implements the radio processing, start the experiment and, when finished, collect the results possibly spread over several computers. If any kind of synchronised or timed start procedure is needed, then it becomes impossible for a single operator. Another common issue of traditional testbeds is the lack of reproducibility of experiments. The scientific value of testing and comparing cutting edge radio techniques depends on the ability to reproduce experiments. Unmanaged environments offer unpredictable propagation and interference fluctuations, that are usually disregarded in experimental results. Due to the random nature of both of these impairments, it is very unlikely that the same conditions will be offered when re-executing the experiment, which will probably affect the assessment of the results obtained.

To address all of these issues we introduce *CorteXlab* [3], a testbed composed of 84 heterogeneous and high performance radio nodes deployed in an electromagnetically isolated room, for cutting edge radio experimentation. With CorteXlab, our main objective is to provide a unified and comprehensive access to a large set of heterogeneous nodes in a reproducible environment, in the aim to foster experimental development of future radio techniques. CorteXlab offers 24 USRPs 2932 from National Instrument, 18 PicoSDRs from Nutaq and 42 IoT-Lab wireless sensor nodes from Hikob. It makes available the full potential of SDR, through the widely accepted GNU Radio toolkit as well as high performance real-time field programable gate array (FPGA) development. The SDR hardware available in CorteXlab ranges from simplistic wireless sensor network

This work has been partially submitted to IEEE Crowncom 2014

(WSN) nodes to full blown $4x4$ MIMO SDR nodes with agile radio capabilities. Through a carefully designed backbone network, those radio nodes are capable of cooperating to emulate complex radio technologies such as network multiple input – multiple output (MIMO) [4], interference alignment (IA) [5] and physical layer based relay networks [6].

CorteXlab [3] is developed, along with 8 other testbeds, under the framework of a nationwide French program Future Internet of Things (FIT) [7]. FIT aims to develop an experimental facility, a federated and competitive infrastructure with international visibility and a broad panel of users. It will provide this facility with a set of complementary components that enable experimentation on innovative services for academic and industrial users. The project will give a means to experiment on mobile wireless communications at the network and application layers thereby accelerating the design of advanced networking technologies for the future internet. In this work, CorteXlab is introduced, detailing its structure, inner workings and main differences with respect to the currently available testbeds.

The remainder of this paper is organised as follows. The next section provides a brief review of the state-of-the-art related to experimental testbed for CR. Section III gives a full description of CorteXlab. Section IV details scenarios currently under investigation for implementation in CorteXlab. Finally, conclusions and perspectives are drawn in section V.

II. State of the Art of Experimental Testbeds

Large-scale CR testbeds are mandatory to develop and evaluate the performance of upcoming physical (PHY) and medium access control (MAC) layers and future CR techniques. Unfortunately, testing new algorithms on real testbeds is complex and time-consuming. Several research groups already obtained interesting results from their own testbeds, but comparing fairly all results and methods with a good reproducibility is currently not possible.

Whereas numerous testbeds are available for specific wireless network standards (sensor or 802.11-oriented), only a few large-scale testbeds have been developed having full SDR and CR capabilities. Appart from on-going projects such as CREW [8] or TRIAL [9] and some other small testbeds involving less than 10 nodes, we found only two testbeds developed respectively at Rutgers University, Orbit [10] and at Virginia Tech., Cornet [11], where USRPs have been adopted.

The Orbit testbed [12] counts with an impressive number of nodes, a total of 400, dispatched on the ceilings of an experiment room. It also uses a clever scheduling and deployment system, called ORBIT Management Framework (OMF) [13], to manage experiments on those 400 nodes. For CR experiments, however, only 28 nodes are of the USRP kind, since the main focus of Orbit was initially on higher layer experiments. The remaining nodes implement a fixed 802.11 PHY/MAC layer. A portal website is used to control all user interaction to the platform.

The Cornet testbed, unlike Orbit, was created from the ground up for PHY/MAC layer research and supports CR experimentation. It counts with a total of 48 nodes, distributed over 4 levels of an academic building in the Virginia Tech campus. The USRP radios use a custom developed radio frequency (RF) board which supports several non ISM frequencies. Those frequencies were made available by the federal communications commission for Cornet's exclusive use. The USRPs are connected to a host computer that performs the signal processing in software, and which are accessible to the user via secure shell (SSH) connection. In spite of the impressive node count and the possibility to transmit freely on non ISM bands, users of Cornet are still faced with a frugal calendar based reservation system and one-by-one access to the nodes, which hinders large scale deployments. Furthermore, since the nodes are on different floors, they have limited radio accessibility to nodes on different floors. Last but not least, after creating a user account anyone can have access to any node at any time, and consequently to the code installed by other users. This intrinsic insecurity can scare security sensitive users away from the platform.

On both cases, the registered users can remotely program and run experiments on the USRPs. Still, the USRPs adopted therein are not the bleeding edge hardware available.We remark that both facilities have their nodes deployed in a conventional environment, which means that the system may suffer from external interference while itself may produce interference on any external system.

III. The CorteXlab Facility

Using as our starting point the current state of these front line testbeds, we have developed CorteXlab, a new facility for large scale radio experimentation. The main objective of CorteXlab is to enable users to run real-time communications with customised application (APP) (with traffic generation), transport control (TCP), network (NET), MAC and PHY layers, implementing current (WIFI, Zigbee, LTE, LTE-A) and upcoming (5G) standards. Users all over the world will be able to schedule experiments and access CorteXlab through a custom built web portal and a standard secure shell (SSH) remote connection. CorteXlab deploys an heterogeneous set of nodes, including WSN and SDR nodes in an electromagnetically (EM) shielded room. By providing access to different nodes technologies, we broaden the scenario possibilities, hopefully appealing to a larger audience of research and development engineers.

Our testbed is complementary to Cornet and Orbit in the following aspects:

- A shielded experimentation room reinforces the reproducibility and the control of the interference environment;
- Heterogeneous high performance nodes are adopted, fostering agility, flexibility and real-time performance;
- An experimentation control plane, called Minus is introduced to control the execution of the experiment.

Each of these aspects are further detailed in the following.

A. Experimentation Room

CorteXlab counts with a completely shielded room to ensure EM isolation from outside. This allows the use of any (restricted or open) frequency in the range $[300MHz - 5GHz]$,

Fig. 1. The experimentation room.

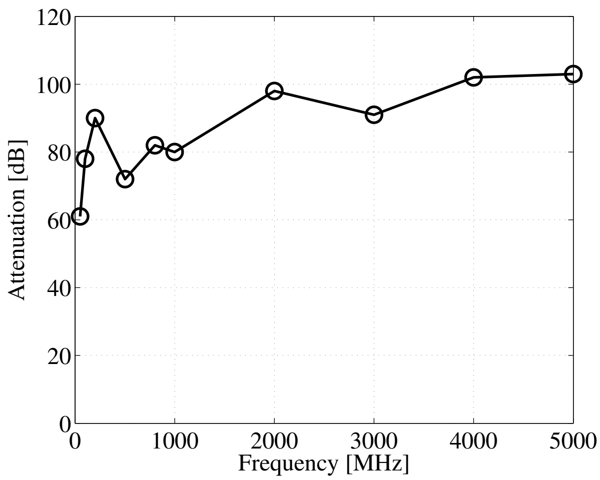

Fig. 2. Signal attenuation in the experimentation room with respect to outside.

which might appeal to researchers looking to test specific frequency or standard related techniques. It also improves reproducibility since all sources of interference and channel impairments are confined to the interior of the room. To avoid excessive reflections, EM wave absorbing foams cover all walls and roof, leaving room from some reflections to occur. A glimpse of the experimentation room can be seen in Figure 1. To validate the EM isolation feature, a measurement campaign has been conducted putting wide-band sources outside and measuring the signal strength inside the room. The results exhibit more than $80dB$ of attenuation in the band $[1000MHz - 5GHz]$ (see Figure 2), with at least $60dB$ throughout the whole target band. We are well aware that by shielding the room, a risk is taken on producing an unrealistic propagation environment. To properly assess the radio environment with the EM shielding, two complementary approaches are currently under study. The first consist in installing reflectors and other materials in the room to create some propagation diversity while avoiding strong corridor effects. An extensive campaign coupling ray-tracing simulations and extensive measurements is under process. However we cannot expect a strong frequency diversity in this way, and the frequency response between different nodes may suffer from being flat. To obtain additional spectral variations, we develop a specific node model which may be host in any CR node. This node is just programmed to generate random signals with a low coherence bandwidth. By adjusting the time variations and the positioning of these nodes over the room, we can generate an environment with a high frequency diversity over the room.

B. Radio Platforms

As previously stated, we chose a mix of three types of nodes in the CorteXlab: low power WSN nodes, general-purpose SDR nodes and real-time high performance SDR nodes.

1) WSN platforms: The WSN nodes are powered by a Cortex A8 processor and count with an off-the-shelf CC2420 802.15.4 radio interface operating at 2.4 GHz, built by a start-up called Hikob. These nodes embed a complete Linux environment and can be programmed to implement different MAC, NET and TCP layers. They will be used to test techniques such as low consumption routing for internet of things and distributed calculation for wireless sensor networks. They can also be used as medium sensing nodes (in the ISM band) that cooperate with SDR nodes to form an intelligent distributed spectrum sensing tier.

2) General purpose SDR platforms: Represented by the National Instruments USRP 2932, the general-purpose SDR nodes will use (but are not limited to) the GNU Radio toolkit for rapid prototyping of transmission techniques mostly reliant on the general purpose processor (GPP) of the host PC. The USRP 2932 is a high end radio platform, counting with a 400 MHz – 4.4 GHz RF board, data rates of up to 40 MHz (with reduced dynamic range, nominal band of 20 MHz), a precise OCXO clock source and a 1 gigabit ethernet (GigE) connection to the host PC. The host PC is based on a Linux environment and will allow users to test not only PHY layer techniques, but also MAC, NET, TRA and APP, by deploying custom kernel modules that can coexist with the stock implementations easily.

Both the use of Linux and GNU Radio will facilitate the development and test at the user's own computer before bringing the experiment over to CorteXlab. Furthermore, the GNU Radio and Linux communities have progressed a lot in the last decade, offering good and free support to users as well as a multitude of open and accessible examples, that can be downloaded off the internet and used without additional charges.

3) High performance SDR platforms: The high end SDR nodes, are composed by the Nutaq PicoSDR radio platforms. In CorteXlab they will come in two flavours: 2x2 and 4x4 MIMO. The 2x2 PicoSDRs possess two Radio420x mezzanine cards on top of a Perseus board, that counts with a powerful Virtex6 FPGA. It will connect to the host PC through a GigE. The 4x4 MIMO PicoSDRs possess four Radio420x mezzanine cards and two Virtex6 FPGAs. It can connect to the host PC through either a GigE or a 4x PCI express (PCIe), the latter offering high bandwidth for full MIMO operation. All Radio420x can tune in the 300 MHz to 3.8 GHz range and can step up to 28 MHz in bandwidth. As with the USRP case, the host PCs are Linux based and will allow users to test PHY and above layers via custom kernel modules.

Initially, two programming modes will be supported by the PicoSDRs:

- **GNU Radio**: this method will work more or less the same way as it does for the USRPs, with the added possibility of 2x2 and 4x4 MIMO. In this mode the user will be able to choose either a passthrough FPGA design, giving direct access to the radio interface, or any one of our custom PHY layer intellectual properties (IP)s, including OFDM and 802.14.5. GNU Radio blocks interfacing to either of these FPGA IPs will be available.
- **Board software development kit (BSDK)**: this is the typical Nutaq method for programming the PicoSDRs and consists of part VHDL FPGA and part C code. It brings full flexibility, allowing a user to bring his own IP to the PicoSDR FPGAs, through ready made bitstreams that will be flashed to the FPGA at the beginning of the experiment. The C code, in the form of a Linux executable, will interface to the FPGA inputs and outputs through a readily available driver that uses either the GigE or PCIe interfaces, and can contain not only PHY layer signal processing but also MAC and above implementations. We know that this method is more challenging than the simple GNU Radio one, and thus we will offer the possibility to use one of our custom PHY IPs (OFDM and 802.14.5) as a starting point. It should be noted that, to use the BSDK method, users will have to own a license of the Xilinx development suite.

Finally, every SDR node is co-located with a single WSN node. They are laid out in the experimentation room in an uniform pattern, as shown in Figure 3, where the node positions (marked with a small "+" symbol) are shown in the floor plan of the room. The nodes are attached to the ceiling and are given a 1.8 m clearance is between them.

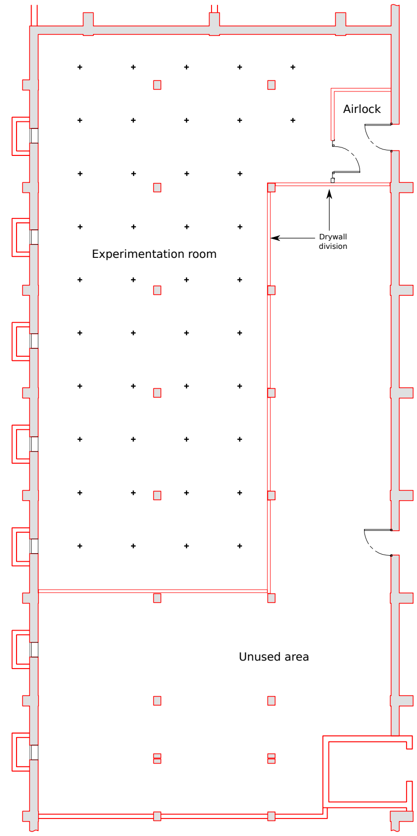

Fig. 3. Experimentation room floor plan. Nodes are positioned at the "+" symbols.

C. Experimentation Control Plane

In order to promote the automated control of all nodes during an experiment, an experimentation control plane was created, called *Minus*. It, allows for the scheduling, deployment, start, finish and results collection of user experiments. Each experiment is organised as a Minus *task*, and contains:

- **Scenario description**: a textual file with the list of nodes, their roles and their parameters;
- **Scenario roles**: the actual firmware, GNU Radio Python scripts, pre-compiled libraries, pre-compiled C code and FPGA bitstreams that define the node behaviour during the experiment;
- **Role parameters**: the configuration values that refine the behaviour of each role.

To better understand the task organisation, lets consider the following spectrum sensing example. Suppose the user wants to create a cooperative spectrum sensing system with two tiers. He chooses a set of USRPs to be small cell base stations (SCBSs), which he further divides into those operating at frequency f_1 with band b_1 (first tier) and those operating at frequency f_2 with band b_2 (second tier). Then he selects another set of USRPs to be user equipments (UEs) that switch automatically between tiers, (f_1, b_1) and (f_2, b_2), according to the estimated probability of missed detection. In this case, there are only two roles: SCBSs and UEs which can be defined by two GNU Radio scripts, say "scbs.py" and "ue.py". There are also only two parameter sets: "-f1 2.412G -W1 1M" and "-f2 2.490G -W2 2M" for each tier involved.

Minus is divided into a server and a client side. At the server side, based on the scenario description file, Minus is able to turn on the related nodes, transfer the task files, start the experiment simultaneously on all nodes (or with a pre-determined delay between them), capture the end of the task, turn nodes off and collect results. The results are later on copied to a predefined disk space accessible only by the concerned user. Along with the results, standard output and standard error files are generated to aid the user in finding problems in the task execution if any.

The Minus client side is composed of a set of command line executables that enable the user to create a task file, submit a task, query the server status of the server and abort a task. The user can access to the client side of Minus through a special machine, called "Airlock" responsible for hosting all user-side services. Finally, through Minus, no direct user access is allowed on the nodes themselves, guaranteeing the security and longevity of the testbed as a whole. Even in the case of node issues, Minus provides a remotely operated reinstall, to restore the node normal operation.

D. System Architecture

Tu support Minus, a complex network infrastructure was built. To enforce security, all networking is divided into virtual LANs and filtered through firewalls. A *services* server hosts all virtualised core services, including Minus. A *storage* server automatically backs up all important data to avoid loss in case of failure.

The CorteXlab network architecture is shown in Figure 4. All SDR nodes possess two network interfaces. The management interface (black connections) is hidden from the user and allows Minus to control the nodes, whereas the data interface is open to the user, and allows communication between nodes. The latter interface can be used to implement cooperative schemes, where nodes can exchange information. These interfaces can also be used to allow one node to control several others, for example in a distributed MIMO approach.

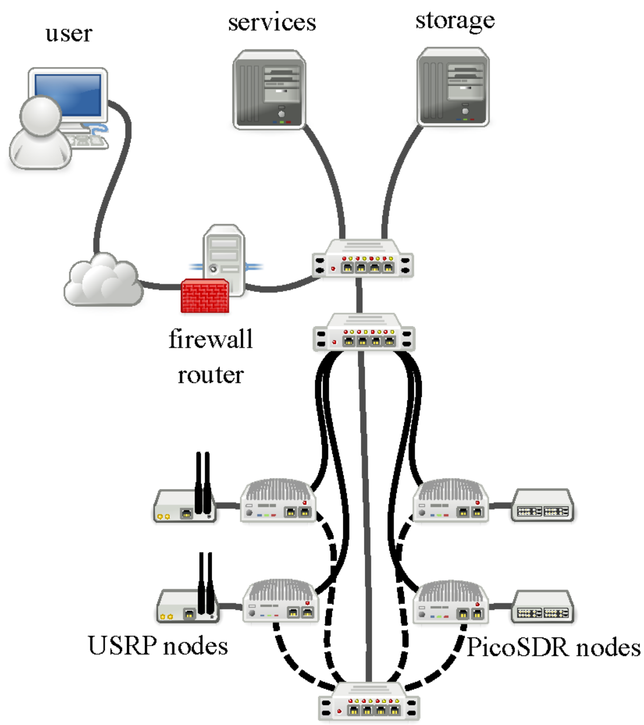

Fig. 4. CorteXlab network architecture.

IV. Experiments and scenarios

All the scenarios presented below are not yet operational. The idea is to illustrate the usage of the platform.

A. Remote point to point transmission with spectrum control

The simplest scenario possible with CorteXlab is the point-to-point transmission. In this scenario, a USRP is configured as an OFDM transmitter, and streams data formatted into packets to an OFDM receiver. Another USRP, configured as an OFDM receiver, tries to decode the incoming packets and, if successful, extracts a packet serial number. The outcome of the whole decoding process is stored in a file for later analysis. A third USRP is configured as a spectrum analyser to snoop on the ongoing transmission between the OFDM transmitter and the OFDM receiver. It streams the band power usage to a visual interface, located at the end-user's workstation. A typical spectrum output is as seen in Fig. 5.

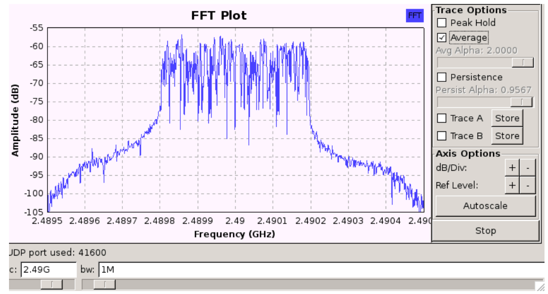

Fig. 5. A typical spectrum usage of OFDM, measured in CorteXlab and streamed to the user interface.

The whole scenario configuration is described in the scenario file, including the spectrum analyser's parameters. The OFDM transmitter and receiver is given by a python script that defines the base-band processing using the standard GNU Radio OFDM blocks. The spectrum analyser is also developed in GNU Radio and is made available as part of the standard CorteXlab toolset. The scenario file required to start the scenario is given by:

```
# Node list

# Spectrum analyser on node3
node3:
    entry node_sa.py
    params --antenna="TX/RX" --rx-gain=25
                -W 2M -f 2.49G -P 41600
    passive true

# Receiver deployed on node2
node2:
    entry benchmark_rx.py
    params --antenna="TX/RX" --rx-gain=25
                -v -W 2M -f 2.49G

# Transmitter deployed on node1
node1:
    entry benchmark_tx.py
    params --antenna="TX/RX" --tx-ampli
                tude=0.2 -v -W 2M -f 2.49G
```

Finally, this scenario aims especially at introducing novel users to the inner workings of CorteXlab. It will be offered in a tutorial form in the future.

B. Interference avoidance

This scenario deals with an avoiding-interference use case where two PicoSDR's are communicating using IEEE ZigBee PHY layer while a cognitive MIMO-OFDM transceiver running on one PicoSDR must avoid interference with them as depicted in Fig 6 [14]. Moreover, a PicoSDR is used as a GNU Radio based remote spectrum analyser to forward the spectrum state to the end user.

This scenario illustrate the capability of the testbed to work on different technologies. This scenario indeed includes GNU radio code adapted for PicoSDR nodes while the $IEEE802.15.4$ transceivers have been designed on the FPGA. All codes are uploaded simultaneously on the nodes through Minus.

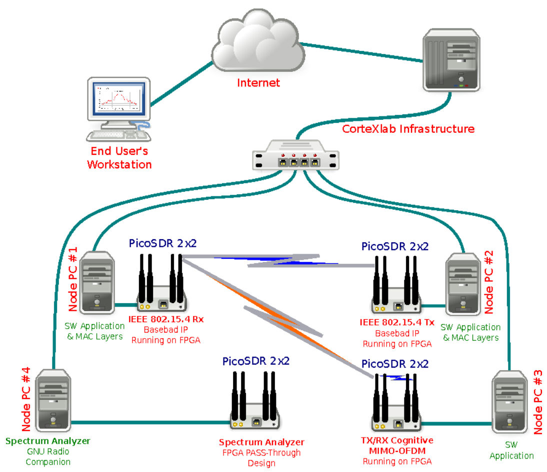

Fig. 6. An Interference Avoidance scenario based on IEEE ZigBee and cognitive MIMO-OFDM transceivers.

C. Interference alignment

In the current literature, IA refers to a large range of signal processing techniques that aim at using the dimensionality of signals transferred over the network as to reduce the space spanned by unwanted signals at the receiver, thereby reducing or completely cancelling interference. One of the key results from the application of IA ideas to networks is that, under specific conditions, dense and high-power wireless networks are not fundamentally interference limited. As an example, under idealised assumptions, using IA in the setting of an interference channel formed by K transmitter-receiver pairs interfering between one another allows each pair to achieve a data rate equal to half of his interference-free channel, regardless of K [15].

Strong efforts in the research community have been done to extend IA far beyond the initial K-user interference channel. The recent review [16] highlights the different technical challenges to be solved before envisioning a practical application among which implementing accurate feedback loops is probably the most important challenge. But beyond the practical implementation of IA solutions in a network, the actual model of the network tends to be complex and involve a large number of hypothesis. These assumptions, or lack thereof, are needed and will play a significant role in the design of IA schemes. These IA schemes are in return heavily tuned to the specific hypotheses made and may not adapt to all cellular configurations, thereby justifying the need to develop an experimental evaluation of these techniques. A downlink cellular network is basically an *interfering broadcast channel*, where BSs transmit towards a number of users and interfere with each other.

However, several tentatives to extend this approach in the context of cellular networks revealed some limiting improvement [17] for the following reasons:

- The direct extension of the IC model to cellular networks relies on defining first the association of each mobile to a given subset of resources. Thus in each cell, the BS decides without coordination which set of resources (e.g. sub-block of frequencies) is given to each mobile. To have a significant gain, IA should be performed for users mutually suffering from interference. The probability of having such situation is relatively low.
- Many works rely on a clustering approach [18]: BSs belonging to a given cluster try to align their precoders for their mobiles sharing the same resource. In this case however, users located at cluster edges cannot benefit from any improvement and are still subject to strong interference.
- Using IA only on space alignment suffer from a lack of degrees of freedom. Note that frequency space may enlarge the space size.
- Signalling requirements reduce the theoretical gain of the system.

More recently Suh and Tse [19] proposed an extension of IA for downlink channels by considering a scenario where each BS uses a reduced space for its own transmission, preserving a given free subspace for other cells. Their solution allows users in a cell to cancel their dominant interferer as well as the intra-cell interference for users in the same sub-band, achieving 1 d.o.f., without any communication between the BSs. Each mobile measures and feedbacks its own free subspace to its BS. Then, the BS tries to form precoders for its global users to align inter-cell interference to out-of-cells interference feedback received from the mobiles. Basically, a certain cooperation between cells is not explicit but exists through the feedback channels. This scheme is more appealing and may be implemented with several options: e.g. the mobile may feedbacks only the best direction in its subspace in which it wants to receive its signal or a larger $N-$subspace in which it expects the signal. The second option offers more capability at the BS side to adapt to the different constraints.

The performance of such scheme relies on many parameters such as synchronisation, feedback capabilities, precoders choice... This why we are working on an implementation of a scenario that would allow to test different algorithms in a reproducible and multi-nodes environment. This scenario is illustrated in Fig.7.

V. Conclusion

Cognitive radio is a paradigm that refers to dynamic radio resource sharing among heterogeneous wireless systems. It is clearly expected to play a fundamental role in a near future, allowing coexistence of multiple radios in a unique frequency band. Cognitive radio will only unlock the high spectral efficiencies foreseen if software defined radio technologies are correctly exploited. In this paper we have introduced CorteXlab, a facility for experimentation with cognitive radio at the PHY layer. At the best of our knowledge, CorteXlab is

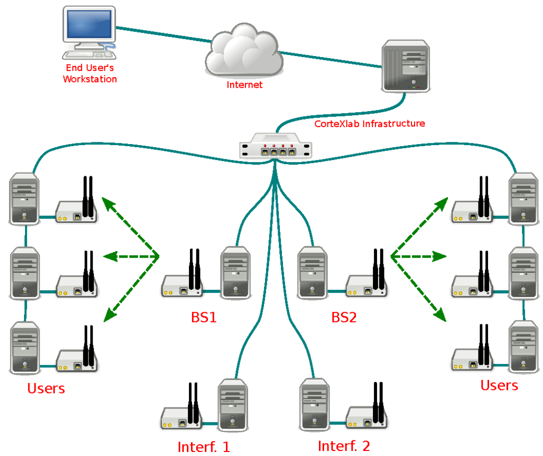

Fig. 7. An Interference Alignment scenario based on cognitive MIMO-OFDM transceivers.

the first facility offered to the research community allowing to remotely test cognitive radio scenarios in a completely reproducible environment. Compared to existing facilities, CorteXlab provides three important new features: coexistence of heterogeneous technologies, a shielded room and the automatic scenario deployment, through Minus. Last but not least, the co-existence of simple low power wireless sensor networks nodes together with complex software radio nodes will open the door to new original and complex scenarios. In the final stages of its development, CorteXlab is currently open only to restricted partners.

Acknowledgment

This work is funded by the project FIT (Future Internet of Things) from the program 'equipement d'excellence' of the French government.

Prof. Jean-Marie Gorce has also been funded by a grant from the Rhone-Alpes region, France.

References

[1] S. Katz and J. Flynn, "Using software defined radio (sdr) to demonstrate concepts in communications and signal processing courses," in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. IEEE, 2009, pp. 1–6.

[2] J. W. Massey, J. Starr, S. Lee, D. Lee, A. Gerstlauer, and R. W. Heath, "Implementation of a real-time wireless interference alignment network," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*. IEEE, 2012, pp. 104–108.

[3] (2012) Cortexlab: Cognitive radio testbed. [Online]. Available: http://www.cortexlab.fr

[4] S. Venkatesan, A. Lozano, and R. Valenzuela, "Network mimo: Overcoming intercell interference in indoor wireless systems," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*. IEEE, 2007, pp. 83–87.

[5] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the k user interference channel," *arXiv preprint arXiv:0707.0323*, 2007.

[6] R. H. Y. Louie, Y. Li, and B. Vucetic, "Practical physical layer network coding for two-way relay channels: performance analysis and comparison," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 2, pp. 764–777, February 2010.

[7] (2011) Equipex fit: Future internet of things. [Online]. Available: http://fit-equipex.fr

[8] (2009) Cognitive radio experimentation world. [Online]. Available: http://www.crew-project.eu/

[9] Trial environment for cognitive radio and networks. [Online]. Available: http://www.tekes.fi/programmes/trial

[10] (2003) Orbit - wireless network testbed. [Online]. Available: http://www.winlab.rutgers.edu/docs/focus/ORBIT.html

[11] (2010) Virginia tech cognitive radio network testbed. [Online]. Available: http://cornet.wireless.vt.edu

[12] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3. IEEE, 2005, pp. 1664–1669.

[13] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "Omf: a control and management framework for networking testbeds," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 54–59, 2010.

[14] A. Massouri, L. Cardoso, B. Guillon, F. Doru-Hutu, G. Villemaud, T. Risset, and J.-M. Gorce, "Cortexlab: An open fpga-based facility for testing sdr and cognitive radio networks in a reproducible environment," in *IEEE Infocom*. IEEE, April 2014.

[15] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," vol. 54, no. 8, pp. 3425–3441, Aug 2008.

[16] O. El Ayach, S. W. Peters, and R. W. Heath, "The practical challenges of interference alignment," vol. 20, no. 1, pp. 35–42, Feb 2013.

[17] D. Aziz, F. Boccardi, and A. Weber, "System-level performance study of interference alignment in cellular systems with base-station coordination," in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2012, pp. 1155–1160.

[18] R. Tresch, M. Guillaud, and E. Riegler, "On the achievability of interference alignment in the k-user constant mimo interference channel," in *IEEE Workshop on Statistical Signal Processing*. IEEE, 2009, pp. 277–280.

[19] C. Suh, M. Ho, and D. N. C. Tse, "Downlink interference alignment," vol. 59, no. 9, pp. 2616–2626, 2011.