A Comparative Study of Cognitive Radio Platforms

Moshe Timothy Masonta
CSIR Meraka and TUT
P.O. Box 395
Pretoria 0001, South Africa
mmasonta@csir.co.za

Mjumo Mzyece
Tshwane University of Technology (TUT)
Pretoria, South Africa
mzyecem@tut.ac.za

Fisseha Mekuria
Council for Scientific and Industrial Research (CSIR)
Pretoria 0001, South Africa
fmekuria@csir.co.za

ABSTRACT

Cognitive radio (CR) technology has become one of the buzzwords within the wireless communications community over the past 12 years. Its ability to learn, decide and adapt to the external environment made CR attractive to regulators, researchers, academia, politicians and the industry. CR promises to bring a paradigm shift in spectrum management policies from command-and-control regime to dynamic and opportunistic spectrum access. Despite more than a decade of research in the CR area, there are too little CR systems ready for the market. This lack of ready CR systems may reflect an overemphasis in the CR literature on theory and simulations with less work done in experimental-based-research and publications. In order to fast-track the real-life deployments of CR systems, the research community is now focusing on the development of CR platforms. With different software defined radio (SDR) packages and hardware available, it is confusing to decide which one to build or use. The objective of this paper is to study the design of CR platforms making use available SDR software packages and hardware. Our conclusion is that CR research should now focus on experimental-based results using real-life CR platforms in order to realize market-ready CR systems.

Categories and Subject Descriptors

C.4 [**Computer-Communication Networks**]: Performance of Systems; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*wireless communication*

General Terms

Performance, Experimentation

Keywords

Cognitive Radio, GNU Radio, Platform, Software Defined Radio, Testbed, USRP.

1. INTRODUCTION

Cognitive radio (CR) technology has become one of the hot research topics in wireless communications network over the past twelve years. CR is an intelligent wireless communication system capable of changing its transceiver parameters based on interaction with its external environment [16]. CR is attractive due to its frequency agility which promises to address the inefficiency and scarcity of radio frequency (RF) problems. The frequency agile CR allows dynamic spectrum access (DSA) by secondary users to coexist with licensed users without causing interference. Today, high volume of literature published on CR topics has produced theoretical and simulation results with little work done on experimental and deployment results. However, the industry, business and regulators are now expecting the realization of real-life deployments of CR systems which are ready for the market. To meet this demand, it is important for the research and academic community to speed up the research and development of CR technologies from simulation based to practical implementations and experimentations. .

Significant number of software defined radios (SDRs) and CR platforms are developed globally to support experimental-based-research and to fast-track the deployment of CR systems. Some of these platforms are built on open source SDR software packages and cost effective hardware, and they are available for free (software only). Knowledge of these packages will assist students and researchers to decide the appropriate and relevant software and hardware to use/purchase when building their own CR platforms or to access the already available online CR platforms. The remote accessibility of most platforms means it may not be necessary for every institution researching on CR to build their platforms, instead, they can use the existing platforms remotely. This paper shows that there are several existing software and hardware packages available assist researchers and students to increase the experimental-based-research results and publications

The remainder of this paper is presented as follows. Section 2 presents the SDR concept and motivates the need for CR platforms. Section 3 presents different SDR software and hardware packages used to build CR platforms. Section 4 briefly outline the R&D efforts at CSIR, and Section 5 concludes the paper.

2. THE NEED FOR CR PLATFORMS

The cognitive radio (CR) concept was first introduced by Joseph Mitola over a decade ago (in 1999) [16]. Since its introduction, there have been numerous research work focused

Table 1: Results Evaluation Techniques

Simulations	Emulation	Platforms
Based on computer modelling and assumptions	Uses modelled and real parts	Based on real & practical systems
High abstraction & complex to duplicate	Near practical & can be duplicated	Practical, deployable and duplicable
Near-realistic results	Improved results reliability	Over-the-air and realistic results
Broader choice of options, faster & parallel running	Uses real measured data	Low computational complexity

on addressing the key CR functions theoretically. However, it was almost five years after its introduction where we started observing the implementations and experimentations of CR technology on testbeds and platforms [17]. Before that, research on CR produced high volumes of simulated results and this has contributed towards the delayed real-life realization of the CR technology. In this section we motivate the need for working on platforms as opposed to theoretical and simulation-based research. We also presents the evolution of CR from software defined radio (SDR) to intelligent radios.

2.1 The Need for CR Platforms

In wireless networking research, academics and researchers are expected to perform precise evaluation and validation of their results to ensure accurate deployments of new network protocols, applications and services. The three commonly used evaluation techniques in the study of communications systems are simulations, emulations, and testbeds or platforms. Simulation is preferably used when modelling the behaviour of the system. The use of emulation requires that at least one piece of the system is a real component. And testbed is a scaled version of the targeted system which consists of the hardware and related software to carry out the evaluation [2]. Table 1 summarizes these three evaluation techniques. Simulation is based on high level of abstraction [2] and does not always produce realistic results [11]. Wireless networks simulation is also weak in modelling complex channels with shadowing, small scale fading and frequency selectivity [12]. Furthermore, high quality network simulation tools are very expensive and require annual license renewals. However, once the license costs are handled, simulations can cheaper, faster, portable, and can explore broader choice of options. On the other hand emulation provides a performance evaluation environment that contains both real and modelled parts [2], but it lacks high degree of realism. The solution is to build wireless research infrastructure platforms in order to produce realistic results. Platforms are becoming necessary to wireless engineers or students in the evaluation and validation of networking results which proved to produce real-life deployments of new technologies, applications and services [2].

2.2 Evolution of CR Platforms

CR can be viewed as an evolution of software defined radio (SDR), which was also introduced by Mitola in 19995 [15]. SDRs have the ability to tune to any frequency band and receive different modulations across a large frequency spectrum by means of programmable hardware which is controlled by software. While SDRs perform the majority of signal processing in the personal computers (PCs), a minimal hardware for radio frequency (RF) front-end is still required for the radio to interact with the external environment. Therefore a complete CR device would consist of SDR software with appropriate RF front-end hardware.

Building CR platforms and conducting research on these platforms is crucial in advancing wireless communications networks towards the future intelligent networks. Chen *et al.* [4] presents three reasons why it is important to build CR network (CRN) testbeds: i) it enables a real-time demonstration of different CR concepts; ii) algorithms and protocols for CRN can be further tested in real-time; and iii) unexpected or practical problems can be discovered earlier for future research. Typical CR applications that can be evaluated in CR platforms include dynamic spectrum management, spectrum sensing, TV white space exploitation (such as in IEEE 802.22), energy efficiency, MAC protocols, wireless distributed computing and green radios [17]. CR platforms have evolved over the years from a basic radio performing spectrum sensing to advanced and intelligent radios capable of making informed decisions, DSA and other functions. The next section present a selected number of these platforms based on their software and hardware packages.

3. DESIGNING CR PLATFORMS

The design and deployment of CR platform requires flexible hardware and software packages. A selected number of common SDR software and supporting RF front-end hardware are discussed in this section. Hybrid platforms are also discussed in this section. The criteria used to select these platforms was mainly based on the wide and common usage of the platforms or packages. Some of these packages are low cost, open source and have a wider community of users and support. The section is concluded by a comparative table for easy of reference.

3.1 SDR Software Packages

Different SDR packages are available for the implementation of CR platforms (see [8]). Four of these SDR packages are discussed in this sub-section: GNU Radio [6], Iris [19], ASGARD [3], and Open Source SCA Implementation::Embedded (OSSIE) [7].

3.1.1 GNU Radio Software

GNU Radio [6] is a free, open source and widely used SDR development toolkit. It provides the signal processing runtime and processing blocks to implement SDRs using external RF hardware. GNU Radio was born in 2001 as a spin-off of the Massachusetts Institute of Technology's PSpectra code and it was rewritten in 2004 [17]. GNU Radio uses the flow-graph technique to develop SDRs. In flow-graphs, the vertices are signal processing blocks and the edges represent the data flow between the blocks. Most signal processing blocks are written in C++, while the signal flow-graphs and graphic user interface (GUI) are written in Python. Python wrapping allows C++ blocks to run internally and convert the input/output in a format used by Python language.

To increase GNU Radio flexibility, new features have been added: event based scheduler replaced the old synchronous polling scheduler, Volk has been added to increase computational speed and the smart-tags for event triggering [6]. To use GNU Radio, one does not require high knowledge of C++ programming because there are many working signal processing blocks provided with the software package. Basic Python programming skills are enough to build great GNU Radio applications. GNU Radio remains one of the official GNU projects having strong support from the international development community.

3.1.2 *IRIS*

IRIS is a dynamically reconfigurable software radio testbed for rapid prototyping and deployment of CR system, developed by the University of Dublin [6]. It is written in C++, and features a configuration method based on Extensible Markup Language (XML). XML is used to specify the signal chain construction and characteristics which can be dynamically reconfigured to meet communications criteria. It is based on flow-graph architecture concept, and it is more flexible and efficient when compared to GNU Radio architecture. IRIS can operate in conjunction with virtually any RF hardware front-end and on a wide variety of operating systems.

3.1.3 *ASGARD*

ASGARD is developed by Aalborg University in Denmark, written in C++ language, and runs as user space application [3]. The focus of the ASGARD project is to tackle the challenge of next generation telecommunication systems using multi user MIMO and aggregated spectrum techniques. ASGARD software architecture is based on Application Programmable Interfaces (APIs), and unlike GNU Radio, it does not have data flow-graphs. The system is designed for Linux OS and exploits the capabilities of a number of open-source libraries. ASGARD is processing-oriented, and allows the developer to manage its application architecture. This makes it more flexible, and easily optimized for each specific processing need. The APIs consist of several components and modules which can start and stop the executions. The platform was demonstrated in different conferences, but its usage is not as widespread as the GNU Radio.

3.1.4 *OSSIE*

Open-Source Software Communication Architecture Implementation - Embedded (OSSIE) is a Virginia Tech open source SDR project based on the Software Communication Architecture (SCA) development framework [7]. The SCA is an open SDR architecture associated with U.S Department of Defense's Joint Tactical Radio System (JTRS) and has been used to implement commercial communications standards. The OSSIE core framework is developed in the C++ language and has been embedded in several digital signal processing (DSP) platforms. OSSIE comes with readily mastered tools, processing blocks, device interface code, and documentation that enable basic tasks of SDR development. OSSIE comes with several tools such as Eclipse integration plug-in and can be integrated with the GNU Radio. Table 2 depicts a comparative summary of the four SDR packages discussed in this subsection.

Table 2: Summary of SDR Software Packages

	Language	Embedded System	Architecture	Cost	Comments
GNU Radio	C++ & Python	Partly supported	Component Based	Free, GPL License	Open Source, large user community
IRIS	C++	Fully Supported	Component Based	Free to members only	Developed by University of Dublin
ASGARD	C++	Partly Supported	Component Based	Free, signing agreements	Hosted: Aalborg University, Denmark
OSSIE	C++	Fully Supported	Component Based	Free, GPL License	Supported by US Dep. of Defence

3.2 SDR Hardware

There are limited number of SDR hardware providing the RF front-end to support the development of CR platforms. In this subsection we look at the hardware platforms commonly used to build CR platforms.

3.2.1 *Universal Software Radio Peripheral (USRP)*

The USRP is developed by Ettus Research [13] and is one of the most widely used hardware for CR platform development. It composes of a motherboard and at least one selectable RF daughterboard. The major computing power on the motherboard comes from the field-programmable gate arrays (FPGA). USRP hardware is available in three different categories: Gigabit Ethernet networked (USRP N2xx), USB bus (USRP B1xx) and Embedded (USRP E1xx) series. The USRP N2xx networked series offers high-bandwidth, high-dynamic range processing capability, larger FPGA, and plug-and-play MIMO capability. The USRP B1xx bus series is a new member of the USRP family, and offers low price for cost-sensitive software radio applications. It can stream up to 8 MHz of RF bandwidth through the USB 2.0 host interface. The USRP E1xx series combines the flexibility of the USRP product family with a Texas Instrument embedded processor [13]. USRP E1xx offers flexibility for embedded applications, and it is ideal for use in mobile applications, such as unmanned systems and standalone base stations. USRPs are very easy to use and cost-effective. However, its weak points are poor RF performance, narrowband operation, its built-in FPGA is difficult to introduce modifications in the firmware unless you are good FPGA programmer and want to write the firmware from the scratch.

3.2.2 *eFalcon*

eFalcon is a flexible hardware-based CR platform developed in 2011 by University of Duisburg-Essen in Germany [10]. To address the FPGA's weakness for debugging the implemented signal processing algorithms, eFalcon's core components include triple-core digital signal processor (DSP) (TMS320C6474 running at 1 GHz system clock) in combination with an FPGA [10]. The DSP has 256 MB of double data rate (DDR2) random access memory (RAM) to enable the implementation of memory-intensive applications. eFalcon consists of multiple serial high-speed interfaces such as Gigabit Ethernet link, three full duplex open base station architecture initiative lanes, commonly referred to as antenna interface, and a Serial Rapid Input Output (SRIO) link. Although not widely used, eFalcon merges the advantage of both digital processors running software and programmable logic allowing massive parallelization to guarantee real-time operation.

3.2.3 *COBRA*

Cognitive Baseband reconfigurable Radio (COBRA) is the second generation of IMEC's reconfigurable platform featuring digital baseband front-end implementing the packet detection using a synchronization processor [9]. COBRA is developed under the auspices of IMEC's Green Radio program which includes four industrial partners: Panasonic, Renesas, Samsung and IMEC spin-off. The COBRA architecture is intended for future mobile handsets, all kinds of battery-powered wireless connectivity devices, and for base-stations for small cells. The associated tools enable C-based compilation, as well as assisted parallelization over multiple cores and threads. This speeds up the design leading to shorter time-to-market and more energy-efficient radios. Cobra architecture can be customized to meet the requirements for many standards including wireless IEEE 802.11n to 802.11a, cellular, and broadcast digital television standards [9].

3.3 Hybrid Platforms

There are CR platforms developed from the scratch to operate with dedicated SDR software and hardware. Such platforms are discussed in this subsection.

3.3.1 WARP

Wireless open-Access Research Platform (WARP) is a scalable, programmable and extensible SDR platform developed by the Rice University, Houston in Texas [1]. One of its goals was to open both hardware and software needed to research, build and prototype next-generation of wireless networks. This is enabling a community of researchers to pool their ideas in undertaking clean-slate prototype networks. WARP's hardware is very similar to the USRP approach. The hardware uses FPGAs for DSP-optimized development, where a number of FPGAs can be scaled as necessary to deliver increased computational power as needed. .

3.3.2 SORA

SORA is a fully programmable SDR platform developed by Microsoft Research Asia [20]. SORA uses both hardware and software techniques to address the challenges of using personal computer (PC) architectures for high-speed SDR. It comes with inexpensive radio control board (RCB) with a radio front-end for transmission and reception. This RCB bridges an RF front-end with PC memory over the high-speed and low-latency PCIe bus, and it can support throughput of up to 16.7 Gbps with sub-microsecond latency [20]. At the physical layer (PHY), SORA takes advantage of lookup tables (LUTs) to reduce the computational requirements of PHY processing. There is an academic kit available to academia for research purposes. The kit contains drivers, sample codes, various software development tools for baseband processing and common Microsoft software suite.

3.3.3 COgnitive RAdio Learning (CORAL)

CORAL is a CR platform developed by the Canadian Communications Research Centre for research and commercial use [18]. CORAL is based on the IEEE 802.11 standard and operates in the License-Exempt ISM bands. It uses a scheduled time division multiplexed (TDM) architecture to assign time-slots to client radios and access points (called WiFi-CR). Each WiFi-CR node is equipped with the GPS to determine position location and timing. The WiFi-CR is capable of supporting beam forming that can be scheduled and synchronized with the TDM slot assignments. CORAL allows the study of such attributes as sensing, adaptation, and collaboration in a field environment.

3.3.4 Berkeley Emulation Engine (BEE2)

BEE2 is a generic multipurpose platform that supports up to 500 Giga-operations per second by distributing the load among its multiple FPGAs and was developed by the University of California at Berkeley Wireless Research Center (http://bee2.eecs.berkeley.edu/). It provides a unique multiuser environment, much like that in a conventional PC cluster, where many users can share a common pool of computing resources with guaranteed computational throughput [5]. The BEE2 hardware is completely built out of commercial common-off-the-shelf (COTS) components. BEE2 is a demonstrator and not a mass production machine; therefore it is not available for sale, hence it is not included in a comparative Table 3.

3.3.5 Lyrtech's small form factor (SFF) SDR

SFF SDR is one of the high end commercial SDR platforms for development of CR. SFF SDRs are designed around DSP and FPGA technology as low-cost, off the-shelf, integrated hardware and software development solutions [21]. SFF SDR advantages include the flexibility, commercial software support and small form factor. Common applications of the SFF SDR platform include the military, public safety, and commercial markets. The SFF SDR development platform is divided into three distinct modules: the digital processing module, data conversion module and RF module [21]. This platform is very expensive and its development is too complex for purely research support.

The SDR packages and hardware discussed in this section are not the only ones available, but they provide a bigger picture on possible resources and toolkits available to design and build CR platform for research and academic purpose. Table 3 depicts a comparative summary of the SDR hardware discussed in this subsection.

4. R&D AT CSIR WIRELESS RESEARCH

The wireless research group within the CSIR is involved in CR and TV white space (TVWS) research and development (R&D) [14]. We are also performing trials with TVWS technology for rural broadband connectivity in South Africa. The group is using the GNU Radio and USRP-2 based platform for performing R&D in spectrum and energy efficient wireless networks. We are also supporting the national regulatory authority on dynamic spectrum access and management for future wireless network regulation.

5. CONCLUSIONS

In order to realize promised theoretical CR capabilities, it is important for researchers to focus on experimenting, evaluating and validating different CR concepts in real-life platforms and testbeds. This paper studies different design and development approaches for CR platform development. Different SDR software and hardware packages are discussed and compared. While all the platforms presented here claims to be CR, we conclude by saying that there is still lot of work that needs to be done to realize true CRs which can perform all theoretically defined CR functions, which includes wide-band sensing and learning from the external environment.

6. REFERENCES

[1] Wireless open-access research platform (warp) website.

Table 3: Summary of SDR Hardware

	Interface	Co-Processor	Operating Frequency	Cost (USD)	RF Bandwidth	Comments
USRP	Gigabit Ethernet & USB	FPGA & embedded Processor	1 MHz to 2.4 GHz	$700 - $1700	8 MHz – 100 MHz	Also available from National Instrument.
eFALCON	Gigabit Ethernet, & Serial RapidIO (SRIO) link	FPGA & DSP	300 –to 4800 MHz	Unknown (Not for sale)	Not specified	Platform under development.
COBRA	For Mobile devices	multi-processor system-on-chip	Cellular Band	Estimated ~$350	1Gbit/s stationary >100 MHz mobile	heterogeneous baseband platform
WARP	Integrated to Software, twisted pair	FPGA	2.4 GHz and 5 GHz	$2000 - $12000, Academic Discount	40 MHz	Provide active user support
SORA	PCI card	FPGA	2.4 GHz and 5 GHz	$400 (exclude RF front-end)	20 MHz	Need to buy 3rd party RF front-end
CORAL	Embedded, support Ethernet	General Purpose Processor	2.4 GHz and 5 GHz	$600	20 MHz	CORAL team offers training on-site at CRC
SFF SDR	10/100 Mbps Ethernet	FPGA & DSP	360 to 960 MHz	Evaluation module: $2900 SFF SDR: $9900	5 MHz and 20 MHz	Operates of Microsoft Windows

[2] A. Alvarez *et al.* In pursuit of massive service emulation: a methodology for testbed building. *IEEE Communications Magazine*, 49(9):162–168, 2011.

[3] ASGARD. ASGARD software radio website, http://asgard.lab.es.aau.dk.

[4] Z. Chen, N. Guo, and R. Qiu. Building a cognitive radio network testbed. In *IEEE Southeastcon Proc.*, pages 91–96, Nashville, March 17 - 20 2011.

[5] C. Cheng, J. Wawrzynek, and R. W. Brodersen. BEE2: a high-end reconfigurable computing system. *IEEE Design & Test of Com.*, 22(2):114–125, 2005.

[6] GNU-Radio. GNU radio website, http://gnuradio.org/redmine/.

[7] C. R. A. Gonzalez *et at.* Open-source sca-based core framework and rapid development tools enable software-defined radio education and research. *IEEE Communications Magazine*, 47(10):48–55, 2009.

[8] O. Gustafsson *et at.* Architectures for cognitive radio testbeds and demonstrators an overview. In *CROWNCOM Proc.*, Cannes, June 9 - 11 2010.

[9] IMEC-COBRA. Imec cobra, http://www.imec.be/.

[10] C. Kocks, A. Viessmann, A. Skrebtsov, G. H. Bruck, and P. Jung. Concept and design of a cognitive radio prototyping platform. In *Int. Conf. on Cog. Radio and Advanced Spec. Management Proc.*, Barcelona, Spain, October 26 - 29 2011.

[11] S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: the incredibles. *SIGMOBILE Mobile Computing Communications*, 2005.

[12] K. Mandke, S. Choi, G. Kim, R. Grant, R. C. Daniels, W. Kim, R. W. Heath, and S. Nettles. Early results on hydra: A flexible MAC/PHY multihop testbed. In *IEEE VTC Proc.*, Dublin, Ireland, April 23 - 25 2007.

[13] Mark Ettus. The Ettus research website, http://www.ettus.com/.

[14] M. T. Masonta, D. Johnson, and M. Mzyece. *The White Space Opportunity in Southern Africa: Measurements with Meraka Cognitive Radio Platform.* Springer Lecture Notes, february 2012.

[15] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, 33(5):26–38, May 1995.

[16] J. Mitola and G. Q. Maguire. Cognitive radio: making software radio more personal. *IEEE Personal Communications Magazine*, 6(4):13–18, August 1999.

[17] P. Pawelczak, K. Nolan, L. Doyle, S. W. Oh, and D. Cabric. Cognitive radio: ten years of experimentation and demonstration. *IEEE Communications Magazine*, 49(3):90–100, 2011.

[18] R. Ruby, S. Hanna, J. Syndor, and V. C. M. Leung. Interference sensing using coral cognitive radio platforms. In *Int. Conf. on CHINACOM Proc.*, 2011.

[19] P. D. Sutton, J. Lotze, H. Lahlou, S. A. Fahmy, K. E. Nolan, B. Ozgul, T. Rondeau, J. Noguera, and L. E. Doyle. Iris: an architecture for cognitive radio network testbeds. *IEEE Comm. Mag.*, 48(9):114–122, 2010.

[20] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker. Sora: high performance software radio using general purpose multi-core processors. In *USENIX Symp. on Net. Systems Design and Impl. Proc.*, 2009.

[21] Texas Instrument. SFF SDR platform, http://www.ti.com/lit/ml/sprt434a/sprt434a.pdf.