Towards A Real-time Cognitive Radio Network Testbed: Architecture, Hardware Platform, and Application to Smart Grid

(Invited Paper)

Robert C. Qiu†, Zhe Chen†, Nan Guo†, Yu Song†, Peng Zhang†, Husheng Li† and Lifeng Lai§
† Cognitive Radio and Smart Grid Institute, Department of Electrical and Computer Engineering, Center for Manufacturing Research, Tennessee Technological University, Cookeville, TN
‡ Department of Electrical Engineering and Computer Science, The University of Tennessee Knoxville, TN
§ Department of Systems Engineering, University of Arkansas, Little Rock, AR

Point of Contact: rqiu@tntech.edu

Abstract—A real-time cognitive radio network testbed is being built. This is the first paper to capture the overall picture of this project. Project scope and philosophy, design architecture, hardware platform, and key algorithms are reported. The use of cognitive radio network for smart grid is for the first time proposed in this paper. This unique testbed is ideal for such purpose.

I. INTRODUCTION

The advent of cognitive radio [1] is inevitable. The Moore’s law ruthlessly drives computing faster and cheaper. Software defined radio (SDR) reflects this trend. Still, the combination of field programmable gate array (FPGA) and digital signal processor (DSP) hardware is still too expensive. Is the general purpose computation on graphics processing units (GPGPU) promising? Ultimately, artificial intelligence and machine learning will become the “brain” of cognitive radio—arguably any machine. How to put “cognition” into cognitive radio network is paramount at this moment. Experimental knowledge is badly needed to dispel the fiction part of cognitive radio. What is cognitive radio? Obviously, cognitive radio includes dynamic spectrum access (DSA) as a special case.

A cognitive radio network testbed is being built. This is the first paper to capture the overall picture of this project—that may last at least for another 3-4 years. The scope and the philosophy of this project are reported here.

A lot of future applications are network-centric. The design of this testbed has taken into account the following applications:

- Cognitive networking: What is the cognitive radio?
- Smart grid: Cognitive radio enhances the network security—the central challenge in smart grid.
- Radio frequency tomography imaging: The low-cost network has revolutionized remote sensing, e.g. through-wall imaging.

The goal of smart grid is to integrate the two-way communication sensors with the electric power grid, by capitalizing on the revolutionary progress of wireless communication industry in the last two decades. The current standards are based on the current wireless technologies such as ZigBee, WiMAX, etc. With the next-generation cognitive radio in the horizon, a natural question arises [2]: how can we tie together cognitive radio in the next-generation smart grid network?

A. Cognitive Radio Project at Tennessee Tech University

It takes many years of research to accumulate the resources to commit a real-world cognitive radio network testbed with at least tens of nodes. This has been made possible by the 2010 Defensive Earmark Project to build the Cognitive Radio Institute at Tennessee Tech University, and the 2010 DURIP award. This is potentially a multi-million dollar project.

The work on cognitive radio has started since 2004. A comprehensive review of our previous work has been done in a 73-page work [3]. The original emphasis is to leverage the in-house expertise on wideband communication. The multi-GHz RF front-end is especially challenging.

A fundamental change of directions is made by recognizing the state-of-the-art cognitive radio network testbed: several nodes are common but networks with bigger numbers of nodes are few (if none). The very word of “cognitive” is so elusive: concrete networks and architectures are required to “define” the concept. The Army Research Laboratory’s (ARL’s) Collaborative Technology Alliances (CTA) Program has been launched, where cognitive radio is part of the program. But few results are available in the public literature. With few experimental results in the public domain, the research has reached its bottleneck on the networking theory. It is our goal to make contributions towards this direction, by publishing the experimental network testbed results. The focus, at this point, is on the physical layer and the cross-layer issues including medium access control (MAC) layer. Its application to smart grid is highlighted.

II. SPECTRUM SENSING

Spectrum sensing is an essential function in cognitive radio. In this section we briefly discuss our proposed spectrum sensing algorithm in [4] and implementation issues.
Our algorithm learns the signal feature as leading eigenvector then uses that feature for signal detection. Since the feature is learned in an online manner, channel effects are also included in the feature. In addition, the feature based detector does not use energy information at all, therefore it avoids the noise uncertainty problem. The performance of the detector in [4] outperforms other blind detectors under -20 dB signal to noise ratio (SNR), given a small amount of overhead for feature extraction. More details of the algorithm can be referred in [4].

The main challenge for implementing the proposed algorithm is covariance matrix acquisition and leading eigenvector calculation. Here we show that it can be implemented in FPGA and DSP with low complexity. In our design, covariance matrix acquisition is implemented in FPGA, and leading eigenvector calculation is implemented in a fixed point DSP. Covariance matrix is acquired by the following equation:

\[
\Sigma_x = \frac{1}{N_s} \sum_{i=1}^{N_s} x_i x_i^T
\]

\[
= \begin{bmatrix}
  r[0] & r[1] & \cdots & r[N-1] \\
  r[1] & r[0] & \cdots & r[N-2] \\
  \vdots & \vdots & \ddots & \vdots \\
  r[N-1] & r[N-2] & \cdots & r[0]
\end{bmatrix}
\]

where

\[ x_i = [x[i], x[i+1], \ldots, x[i+N-1]]^T \]

are the \( N \times 1 \) sample vectors of received signal and

\[ r[n] = \frac{1}{N_s} \sum_{m=1}^{N_s} x[m] x[m-n], \quad n = 0, 1, \ldots, N-1 \]

are the autocorrelation samples of the received signal.

As can be seen, \( \Sigma_x \) is a symmetric Toeplitz matrix. For an \( N \times N \) matrix, we only need to calculate \( N \) elements of the autocorrelation, which can be obtained parallelly in FPGA in real-time, regardless of \( N_s \). We have implemented a real-time 32 \( \times \) 32 covariance matrix acquisition module using Xilinx Virtex 4 SX35 FPGA.

The leading eigenvector calculation can then be implemented using a simplified Fast-PCA algorithm for fixed-point DSP [5], [6]. The simplified algorithm is illustrated in Table I.

\[ \phi \] converges within a small number of iterations [5]. The total complexity is approximately \( O(N^2) \). We have implemented this algorithm in Texas Instruments (TI) C64x+ DSP.

The eigenvector extracted by the fixed point DSP is shown in Fig. 1. The normalized mean square error of this result and the one from floating point eigen-decomposition algorithm in Matlab is about 6%.

Another spectrum sensing algorithm, which is based on fast Fourier transform (FFT), and demonstration of real-time spectrum sensing can be referred in [7].

III. WIDEBAND COGNITIVE RADIO PLATFORM

The goal of the in-house wideband cognitive radio platform is to study the wideband nature of cognitive radio. Ideally, we want a radio that covers all the desired frequency bands. This requires several GHz bandwidth [3]. Rather than pursuing the theoretical aspects, we explore the feasible solutions by cashing out our previous expertise in this area.

Spectral holes, or chunks of available spectra, can exist in a wide frequency range at different time, and discontinued spectral holes may appear at the same time. Obviously, narrowband spectrum sensing is not able to detect multiple available spectra simultaneously. An existing approach is to use a tunable narrowband bandpass filter at the radio frequency (RF) frontend to search one narrow frequency band at a time [8], however, this is time consuming. In order to ultimately use the scarce spectrum wideband spectrum sensing, a real-time overall picture about current spectral usage is required. Some approaches use signal processing algorithms like compressed sensing [9] and wavelet approach [10] to relax the burden on RF frontend and sensing time. Joint detection methods are also proposed to maximize the aggregate opportunistic throughput [11]. All these algorithms ask for a very wide processing bandwidth, posing tremendous challenges in system design and implementation. Specifically, wideband spectrum sensing requires (1) wideband RF frontend with large dynamic range, (2) GHz digitizing and high data-rate real-time processing, and (3) great computing power for sophisticated detection and control algorithms.

Having known the goal and challenges of wideband spectrum sensing, a preliminary design is proposed and its conceptual architecture is shown in Fig. 2. This wideband spectrum sensing platform consists of three major building blocks: an RF frontend, a digitizer with FPGA array for real-time

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SIMPLIFIED ALGORITHM FOR COMPUTING THE LEADING EIGENVECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Compute covariance matrix ( \Sigma_x )</td>
<td></td>
</tr>
<tr>
<td>2. Initialize eigenvector ( \phi ) of size ( N \times 1 ) e.g. randomly</td>
<td></td>
</tr>
<tr>
<td>3. Update ( \phi ) as ( \phi \leftarrow \Sigma_x \phi )</td>
<td></td>
</tr>
<tr>
<td>4. Normalize ( \phi ) by dividing it by its norm: ( \phi \leftarrow \phi / | \phi | )</td>
<td></td>
</tr>
<tr>
<td>5. If ( \phi ) has not converged, go back to step 3</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1. Extracted signal eigenvector in DSP.
processing, and a computing engine for heavy computation and control.

A. RF Frontend

The RF frontend can be built using some off-the-shelf products. Key parameters for this module would include bandwidth of at least 500 MHz, frequency range of 400-900 MHz (or wider) covering the digital TV (DTV) band, and dynamic range of at least 60 dB required by spectrum sensing. For DTV band spectrum sensing, frequency conversion is not necessary, but down-conversion is preferred to ease sampling rate requirement when the center frequency is higher.

B. Digitizer with FPGA Array

An off-the-shelf digitizer with up to 3 Gsps sampling rate can be considered. The digitizer uses a PXI Express Peripheral Module (3U), which integrates two 3 Gsps 8-bit analog-to-digital converters (ADCs), a clock circuitry, 2 banks of 1GByte double data rate 2 (DDR2) memory and a Xilinx Virtex-5 FPGA, under the 3U format. This product can work in a standalone fashion or with other boards to form an FPGA array in a PXI Express chassis. One potential problem with this digitizer might be insufficient dynamic range due to limited quantization. Additional means such as noise dithering or subband approaches should be considered to enhance the dynamic range.

C. Computing Engine

The purpose to employ a computing engine is multi-fold. Traditionally, FPGAs and DSPs are bundled together and this makes perfect sense. However, there is a trend to rely on general personal computer (PC) for digital processing tasks and control. This trend is catalyzed by recent hardware advance in PC industry. The “magic” is made by a GPU that is originally for highly paralleled video signal processing. GPGPU refers to a relatively new method by which the multiple cores of a GPU can be utilized for general purpose parallel computing [12]. This idea of utilizing GPUs for non-graphical applications first became popular in 2003 but was limited by the amount of knowledge required to successfully write such programs. November 2008 saw the introduction of Nvidia’s G80 architecture which brought greater versatility through support of the C computing language and a more generalized and programmer friendly hardware structure [13]. Enhanced by a GPU, a regular PC is turned into a supercomputer so that some highly computational tasks such as machine learning, can be considered and implemented for spectrum sensing. Of course, currently the FPGA based real-time processing cannot be replaced by a computer with GPU acceleration due to unacceptable latency.

IV. NARROWBAND PLATFORMS FOR COGNITIVE RADIO

One way of building cognitive radio networks is using off-the-shelf hardware platforms. Till today, most of the available off-the-shelf hardware platforms in this area are designed for software defined radio (SDR). They can be used to build cognitive radio nodes. In this section, two popular hardware platforms for SDR will be reviewed and compared.

A. Small Form Factor Software Defined Radio Development Platform

The small form factor (SFF) software defined radio (SDR) development platform (DP) provided by Lyrtech in collaboration with TI and Xilinx is a self-contained platform consisting of three separate boards: digital processing module, data conversion module, and RF module, as shown in Fig. 3 [14], [15].

The digital processing module is designed based on the TMS320DM6446 system-on-chip (SoC) from TI and Virtex-4 SX35 FPGA from Xilinx. The TMS320DM6446 SoC has a C64x+ DSP core running at 594 MHz together with an advanced RISC machine (ARM9) core running at 297 MHz. The digital processing module also comes with a 10/100 Mbps Ethernet port. The data conversion module is equipped with a 125 Msps 14-bit dual channel ADC and a 500 Msps 16-bit dual channel digital-to-analog converter (DAC). It also has a Xilinx Virtex-4 LX25 FPGA. The low-band tunable RF module can be configured to have either a 5 MHz or a 20 MHz bandwidth with working frequencies of 200-1050 MHz for the transmitter and 200-1000 MHz for the receiver. The nominal noise figure of this RF module is 5 dB. Other frequency bands may be covered by several other RF modules.
B. Universal Software Radio Peripheral

The next generation universal software radio peripheral (USRP2) provided by Ettus Research became available for sale in 2009 [16]. Unlike the SFF SDR DP, USRP2 consists of a motherboard and one or more selectable RF daughterboards, as shown in Fig. 4.

The major computing power on the motherboard comes from a Xilinx Spartan-3 XC3S2000 FPGA. The motherboard is also equipped with a 100 Msps 14-bit dual channel ADC, a 400 Msps 16-bit dual channel DAC, and a Gigabit Ethernet port that can be connected to a host computer. There are some RF daughterboards available for USRP2. Among them, a newly developed RF daughterboard called WBX covers a wide frequency band of 50 MHz to 2.2 GHz, with a nominal noise figure of 5-7 dB.

C. Comparisons between SFF SDR DP and USRP2

Both SFF SDR DP and USRP2 are powerful tools for research on cognitive radio. However, there are still something to be considered to choose between them in practice.

a) Computing Power on the Board: SFF SDR DP has one Xilinx Virtex-4 SX35 FPGA connected with one TI TMS320DM6446 SoC on the digital processing module, while USRP2 only has a Xilinx Spartan-3 XC3S2000 FPGA on the motherboard. Obviously, SFF SDR DP has more computing power on the board and it has the ability of handling more digital processing jobs on the board. On the other hand, it is possible to run more complex algorithms on the board in real-time. Since USRP2 has relatively low computing power on the board, in most cases, it needs to connect to a host computer for extra computing powers.

b) RF Modules: Till today, there are six RF modules that can be used for SFF SDR DP, covering the receiver frequency bands of 200-1000 MHz, 1.6-2.7 GHz, and 3.3-3.8 GHz, with a RF bandwidth varying from 3.5 MHz to 22 MHz. For USRP2, there are more than ten RF modules including third party solutions, covering almost every frequency band under 4 GHz, with an RF bandwidth up to 50 MHz.

c) Interfaces: SFF SDR DP has various interfaces on the board, such as RS-232 serial port, 10/100 Mbps Ethernet port, universal serial bus (USB) port, and secure digital (SD) memory card slot. Unfortunately, till today the USB port and the SD memory card slot are not supported. USRP2 has a Gigabit Ethernet port using raw Ethernet for exchanging data with a host computer. It also supports SD memory card for booting the motherboard. Moreover, USRP2 has a flexible expansion port that can connect to other boards.

d) Available Resources: USRP2 can work with the open source software GNU Radio [17], which has many available source codes and technical discussions on the Internet. Even the source code for the FPGA is open to everyone. Compared to USRP2, SFF SDR DP has less available and stable source code that can be reused for further developments. Moreover, since there is not too much technical information about SFF SDR DP on the Internet and not all the technical details are included in the product documents, sometimes technical supports provided by the company become the major source for solving technical problems, which could be a bottleneck for a project.

e) Expenses: The listed price for each set of USRP2 is about $1500-$2000, depending on the RF daughterboards and the antennas. The development tools and source code for USRP2, i.e., GNU Radio, is free. The listed price for each set of SFF SDR DP is a few times higher than that of USRP2. Moreover, in order to employ the model-based development kit (MBDK) for developments, more commercial softwares, such as Lyrtech ADP, and certain versions of Xilinx ISE Foundation with System Generator, TI CCS, and MathWorks MATLAB with Simulink, are required.

f) Developments: Before starting developments on SFF SDR DP and USRP2, development environments should be set up first. Setting up all the development environments for USRP2 usually takes hours, while setting up development environments for SFF SDR DP is a procedure that could last up to months. The major development languages for SFF SDR DP are C language and VHDL. And the major development languages for USRP2 are C++ and Python on computer, which are easier for development, debugging and maintenance.
V. BUILDING COGNITIVE RADIO NETWORK TESTBED

A cognitive radio network testbed is being built using off-the-shelf hardware platforms.

A. Nodes for Cognitive Radio Networks

What should be done in the nodes for cognitive radio networks? The following architecture is proposed.

Fig. 5 shows the proposed architecture of nodes for cognitive radio networks. The hardware platform can be any one suitable for cognitive radio. The hardware abstraction layer (HAL) is an interface for upper-level drivers or softwares that screens hardware specific details. It provides data interface to both receiver data path and transmitter data path, as well as an access interface to non-volatile memories. The spectrum and channel manager manages all the spectrum and channel related resources, including links, frequencies, and modulation methods. There are several functional modules interface with the spectrum and channel manager. The spectrum detection and prediction module provides the information regarding the availability of some frequency segments. The decision making module utilizes decision algorithms to make decisions like which channel will be used and when it will be used. More learning algorithms can be implemented as an independent module to learn and reason from inputs. The geolocation module outputs the latitude and longitude of the node. The spectrum and channel manager can use such geolocation information to load prior information about current location from the knowledge/policy/data base. The routing manager employs routing algorithms to select the best route for sending and relaying data packages. While the data manager organizes all the data from upper-level applications and the data to be relayed. The security manager provides encryption to data manager, routing manager, and spectrum and channel manager. The knowledge/policy/data base stores prior knowledge, policies, data, and experience.

With cognitive radio nodes built up, a cognitive radio network testbed can be deployed.

B. A Scenario of Cognitive Radio Networks

Fig. 6 shows a scenario of cognitive radio networks where no dedicated cognitive radio base station exists. A primary users’ base station stands in the center of the figure for serving primary users. At the same time, secondary users co-exist in the same region. Some of the secondary users may also connect with other networks like the Internet, although these links may not be permanent. They play a role like gateway.

C. Challenges

Although cognitive radio networks are promising, there are still problems and difficulties in building it.

How to avoid interference to primary networks? How to perform collaborated spectrum sensing among secondary users? How to establish links between secondary users? How to route among secondary users dynamically? How to improve the throughput of the entire secondary networks?

Difficulties on the node implementation include but not limit to: (1) system design; (2) synchronization among the whole networks; (3) collaboration and participation among hardware, firmware, and software; (4) algorithm optimization;
(5) data conversion from floating-point to fixed-point; (6) firmware/software architecture; (7) code optimization for real-time implementation; (8) system debugging.

All of the above questions need to be answered in the future research.

VI. Enabling Smart Grid With Cognitive Radio Network

We are planning to use this hardware testbed to study the networking problems in smart grid using cognitive radio. The following four aspects in smart grid will be studied, designed and demonstrated over this testbed, particularly for the physical security (e.g., stability, controllability and confidentiality) of power grid: (a) Scheduling in MAC layer: the real-time and large volume data traffic in smart grid requires a system state aware scheduling mechanism; the FPGA arrays on the hardware board can carry out online and intelligent computation for evaluating the importance of data packet according to the current system state of power grid; the performance will be evaluated using the whole cognitive radio network testbed. (b) Routing algorithm: the main focus will be a spectrum and situation aware routing algorithm; the route will be updated according to the spectrum occupancies and power grid situation throughout the whole network; the algorithm will be implemented and evaluated in the testbed, demonstrating the corresponding reliability. (c) Key management algorithms: the focus here is to use the testbed to implement the key generation schemes developed in [18], [19] for smart grid. The basic idea is to dynamically generate keys from the random channel gains in both wireless channel and power line. We will use the testbed to quantify the key rate that can be generated in various real time environment. (d) Attacks defense strategies: We also plan to use the testbed to study the effects of various attacks, including jamming attack, Byzantine attack etc. on the stability of the smart grid network. More importantly, we will design algorithms to detect and further mitigate these attacks.

VII. Conclusion

A real-time cognitive radio network testbed is being built. This is the first paper to capture the overall picture of this project. Project scope and philosophy, design architecture, hardware platform, key algorithms are reported. Security and cognitive nature will be two focused issues. Especially, how to tie together these two issues? Both algorithms and hardware implementations will be explored.

This is the first paper towards this new direction. Many open problems will be reported elsewhere. For example, a critical decision is to decide on the two platforms (Lytech or USRP2). The unfriendly development environment of the Lytech platform and the associated high cost have major impacts on the future decision. Data is being collected from USRP2 nodes for this purpose.

The goal of our in-house wideband radio platform is to explore the wideband nature of the cognitive radio. It will not be mature for the large-scale network testbed.

The collaboration of three institutions is very lucky, which is made possible by the joint NSF award. The use of cognitive radio network for smart grid is for the first time proposed in the invited talk [2] and in this paper. This unique testbed is ideal for such purpose.

ACKNOWLEDGMENT

This work is funded by National Science Foundation through grants (ECCS-0901420), (ECCS-0821658), and (ECCS-0622125), Office of Naval Research through a grant (N00014-07-1-0529), 2010 DURIP grant, and 2010 Defensive Earmark Project. The technical discussions with government employees Santanu K. Das, Brian Sadler, Robert Ulman, Robert Headrick, Mike Wicks, Vasu Chakravarthy, Lee Liyeh Liou, David Lin, Thomas Dalrymple, Stephen Hary, and Seng Hong are very helpful.

REFERENCES