

Evaluation of Open-source Software Frameworks for High Fidelity Simulation of Cognitive Radio Networks

Vincent Le Nir and Bart Scheers

Department Communication, Information Systems & Sensors (CISS)

Royal Military Academy

Brussels, BELGIUM

Email: {vincent.lenir,bart.scheers}@rma.ac.be

Abstract—High fidelity simulation of cognitive radio networks (CRN) requires a CRN software framework with libraries of components at all layers of the open systems interconnection (OSI) model (e.g. modulation schemes, protocols, routing algorithms) along with realistic channel models for simulating a high number of nodes. At the same time, the CRN software framework should provide interfaces with hardware platforms for testbed evaluation and hardware-in-the-loop (HIL). Unfortunately, there is no such software framework in the research community. This paper studies the combination of some open-source frameworks for high fidelity simulation of CRN. In particular, some challenges are highlighted for the combination of radio simulators (e.g. GNU Radio, CogWave) and network simulators (e.g. OMNeT++, ns-3) at the level of network layer, data link layer and physical layer. Based on this evaluation, new extensions to the CogWave open-source software framework are proposed for high fidelity simulation of CRN.

Keywords—Cognitive Radio, Software Defined Radio, CogWave, GNU Radio, OMNeT++, ns-3

I. INTRODUCTION

A cognitive radio network (CRN) is a network of software defined radios (SDR) which can autoconfigure and autonomously adapt its configurable parameters according to the user needs and the electromagnetic environment [1], [2]. CRN offers various dimensions of configurable parameters at all layers of the open systems interconnection (OSI) model. At physical layer (layer 1), this includes spectrum sensing (e.g. energy detection, feature detection, sensing time), dynamic spectrum access (e.g. overlay, underlay, interweave), modulation scheme (e.g. QAM, OFDM, SC-FDM, DSSS, FH), forward error correction (e.g. block coding, convolutional coding, turbo coding), and other parameters (e.g. frequency, bandwidth, power). At data link layer (layer 2), this includes media access control (MAC) (e.g. CSMA, OFDMA, TDMA, Aloha), logical link control (e.g. flow control, error control, ARQ) and topology control (e.g. clustering). At network layer (layer 3), this includes network protocol (e.g. IPv4, IPv6), routing algorithms (e.g. OLSR, AODV, OSPF). High fidelity simulation of CRN should provide the flexibility to adjust manually or autonomously these configurable parameters according to the user needs and the electromagnetic environment.

The CRN software framework should provide libraries of components at all layers of the OSI model (e.g. modulation schemes, protocols, routing algorithms) along with realistic

channel models for simulating a high number of nodes. At the same time, the CRN software framework should provide interfaces with hardware platforms for testbed evaluation and hardware-in-the-loop (HIL). High fidelity simulation means that there is no abstraction at the level of data link layer, physical layer and that it includes realistic channel models. The CRN software framework should also provide the possibility to reuse the same code either in a testbed mode with hardware platforms (small number of nodes with real-time scheduling), in an emulation mode (HIL with real-time scheduling) or in a simulation mode with realistic channel models (high number of nodes). A list of CRN software frameworks and SDR hardware along with experiments is provided in [3], [4], [5].

Unfortunately, there is no such software framework in the research community. This paper studies the combination of some open-source software frameworks for the high fidelity simulation of CRN. Several types of open-source software frameworks have been identified for this purpose. On one hand, radio simulators such as GNU Radio or CogWave provide many signal processing blocks and modulation schemes for the physical layer. However, they provide only basic MAC functionalities for the data link layer and do not provide functions for upper layers. On the other hand, network simulators such as OMNeT++ and ns-3 provide libraries for the internet stack (e.g. TCP, UDP, IPv4, IPv6), wired and wireless protocols (e.g. Ethernet, PPP, IEEE 802.11, IEEE 802.16, LTE), mobile ad-hoc network (MANET) protocols, mobility, and many other protocols and components. They offer real-time schedulers for integration into testbed and virtual machine environments. Moreover, they provide fairly accurate representation of the layers above and including the data link layer. However, they abstract significantly the physical layer, the channel models and some parts of the data link layer.

Based on the evaluation of the combination of some open-source software frameworks for high fidelity simulation of CRN, we propose new extensions to the CogWave open-source software framework for the data link layer, the interfaces at the physical layer and the channel models. These extensions allows CogWave to be used either in a testbed mode with hardware platforms, in an emulation mode for HIL or in a simulation mode with realistic channel models for a higher number of nodes.

This paper is organized as follows. Section II reviews some

open-source frameworks for high fidelity simulation of CRN. Section III provides some challenges for the combination of open-source frameworks at the network layer, the data link layer and the physical layer. Section IV proposes new extensions to the CogWave open-source software framework for high fidelity simulation of CRN. Finally, Section V concludes this paper.

II. OPEN-SOURCE SOFTWARE FRAMEWORKS FOR HIGH FIDELITY SIMULATION OF CRN

There are many open-source software frameworks for wireless communications in the research community. This Section reviews some open-source frameworks and evaluates the layers at which they operate for high fidelity simulation of CRN.

A. GNU Radio

GNU Radio [6] is an open-source software framework which provides signal processing blocks to implement software radios. The GNU Radio framework is built by a combination of Python and C++. The main signal processing blocks are written in C++ and integrated with Python using simplified wrapper and interface generator (SWIG). The GNU Radio Companion provides a graphical user interface (GUI) to connect GNU Radio blocks and to design flow graphs similar to Simulink. Many signal processing blocks are available such as filters (FIR, IIR), fourier transforms (DFT, FFT), equalizers, modulation schemes (GMSK, PSK, QAM, OFDM), error-correcting codes (Reed-Solomon, Viterbi, turbo codes), realistic channel models (AWGN channel, fading channel, frequency-selective fading, hardware impairments), interfaces with hardware (USRP, OsmoSDR platforms). However, the GNU Radio framework provides only basic MAC functionalities (simple CSMA MAC [7]) at the data link layer and do not provide functions for upper layers. The GNU Radio framework can be interfaced with other applications by named pipes (FIFO) or virtual network interfaces at data link layer (TAP) or network layer (TUN). Most works in the literature consider to interface GNU Radio with the Click modular router software (CSMA/routing) [8] or to implement time-critical MAC functionalities (Bluetooth, IEEE 802.11) in the field-programmable gate array (FPGA) of the hardware platform [9].

B. CogWave

CogWave [10] is an open-source software framework aiming at developing cognitive radio (CR) modulation schemes. The CogWave framework uses the Qt framework, the communications library IT++, the USRP universal hardware driver (UHD) and other libraries to enable real-time transmission between USRP devices. The CogWave framework provides many modulation schemes, such as the multichannel DAA-OFDM [10], the DADS modulation scheme with a short spreading sequence [11], and other modulation schemes ported from the GNU Radio framework (OFDM, BPSK, QPSK, GMSK, CPFSK etc.). Compared to the GNU Radio framework, the CogWave framework is able to reconfigure the modulation scheme during run-time (e.g. switching from DADS to multichannel DAA-OFDM in the presence of a jammer) and allows precise timing control for burst transmissions (FDD and TDD). The CogWave framework can be interfaced with

other applications by named pipes (FIFO) or virtual network interfaces at data link layer (TAP) or network layer (TUN). However, the CogWave framework does not provide functions for the data link layer and upper layers.

C. OMNeT++

OMNeT++ [12] is a discrete-event network simulator which provides C++ libraries and model frameworks to support various types of networks (e.g. wireless ad-hoc networks). The INET framework contains models for the wired protocols (PPP, Ethernet, etc.), wireless protocols (IEEE 802.11, IEEE 802.16, IEEE 802.15.4, etc.), routing protocols (OSPF, AODV, DYMO, DSDV, DST, OLSR, etc.), network layer (IPv4, IPv6, ARP, etc.), transport layer (TCP, UDP, SCTP, RTP, etc.), application layer (HTTP, DHCP, Video, Voice, P2P, etc.). A cognitive radio extension to the OMNeT++ framework has been developed in [13]. The OMNeT++ framework provides strong GUI support for simulation visualization. It also supports real-time simulation by the real-time scheduler which synchronizes the simulation time with the computer system time. Hardware interfacing is supported by a real-time socket scheduler and a raw internet socket connected to an underlying physical interface. The real-time socket scheduler waits for incoming messages from an external device and send outgoing messages to the same device. However, the OMNeT++ framework abstracts significantly the physical layer, the channel and some parts of the data link layer.

D. ns-3

Ns-3 [14] is a discrete-event network simulator which provides C++ libraries of network simulation models wrapped in Python. The ns-3 framework provides models for wired protocols (PPP, Ethernet, etc.), wireless protocols (IEEE 802.11, IEEE 802.16, LTE, etc.), routing protocols (OSPF, AODV, DSDV, OLSR, etc.), network layer (IPv4, IPv6, ARP, etc.), transport layer (TCP, UDP, etc.), application layer (HTTP, etc.). A cognitive radio extension to the ns-3 framework has been developed in [15]. This extension is based on the cognitive radio extension to the ns-2 framework [16]. Another cognitive radio extension to the ns-2 framework has been developed in [17]. The ns-3 framework supports real-time simulation by the real-time scheduler. Hardware interfacing is supported by a raw internet socket connected to an underlying physical interface. However, the ns-3 framework abstracts significantly the physical layer and the channel.

E. CORE

CORE (Common Open Research Emulator) [18] is an open-source framework for emulating networks on one or more PCs. The CORE framework emulates the network layer (IPv4, IPv6, etc.), routing protocols (OSPF, etc.), transport layer (TCP, etc.) and above layers using virtual network stacks in the Linux operating system. The emulation is controlled by and easy-to-use Tcl/Tk GUI. Because the CORE framework is a live-running emulation, CORE networks can be connected in real-time to physical networks. However, the CORE framework abstracts significantly the data link layer, the physical layer and the channel.

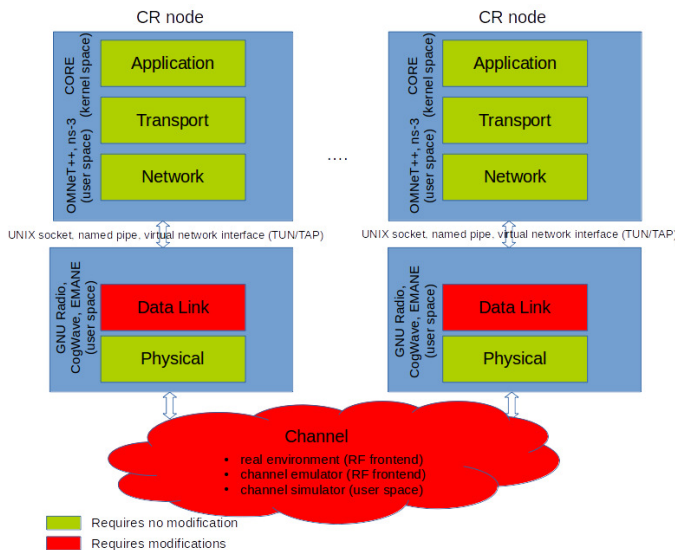


Fig. 1. Combination at the network layer

F. EMANE

EMANE (Extendable Mobile Ad-hoc Network Emulator) [19] is an open-source framework which provides C++ libraries of network simulation models wrapped in Python. The EMANE framework provides models for wireless protocols (IEEE 802.11abg, etc.). However, the EMANE framework abstracts significantly the channel and the physical layer which is based on BER vs SNR curves instead of actual transmitted IQ samples. The EMANE framework does not provide interfaces with hardware devices.

III. COMBINATION BETWEEN OPEN-SOURCE FRAMEWORKS FOR HIGH FIDELITY SIMULATION OF CRN

Unfortunately, none of the open-source frameworks reviewed in Section II can provide libraries of components at all layers of the OSI model, realistic channel models and interfaces with hardware platforms. A possible solution is either to build a new CRN software framework from scratch, to extend existing frameworks for the missing layers or to combine several open-source software frameworks. This Section provides some challenges for the combination of open-source frameworks at the network layer, the data link layer and the physical layer.

A. Combination at the network layer

The combination of open-source frameworks at the network layer is given in Figure 1. The layers above and including the network layer are provided by the network simulators (OMNeT++, ns-3). The layers below the network layer are provided by the radio simulators (GNU Radio, CogWave). The network and radio simulators are connected by a UNIX socket, named pipe (FIFO), or virtual network interface (TUN/TAP). The combination at the network layer requires the development of the data link layer in radio simulators (GNU Radio, CogWave). Indeed, the GNU Radio framework provides only basic MAC functionalities (simple CSMA MAC [7]) at the data link layer. The CogWave framework does not provide functions for the data link layer.

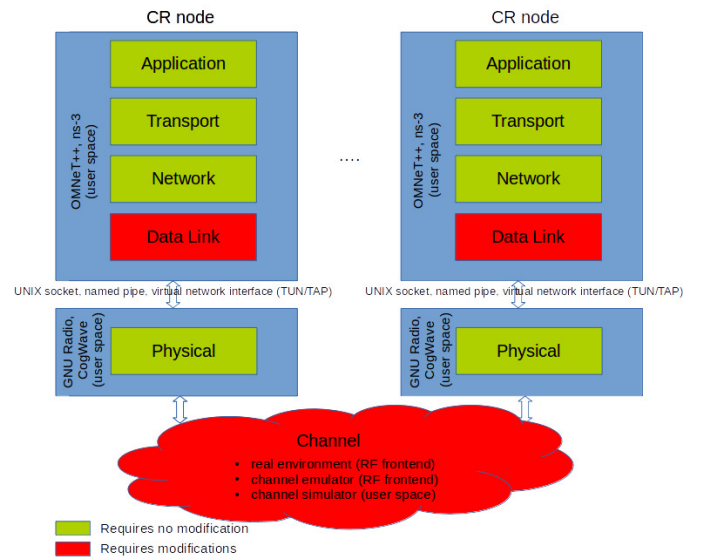


Fig. 2. Combination at the data link layer

Radio simulators (GNU Radio, CogWave) provide interfaces with hardware devices to test in a real environment or in a channel emulator hardware (testbed mode). They also provide realistic channel models (AWGN channel, fading channel, frequency-selective fading, hardware impairments) for the simulation of small number of nodes. The combination at the network layer requires either the development of realistic channel models for the simulation of high number of nodes in the radio simulators (similarly to [20]), or the development of realistic channel models for the simulation of high number of nodes in the network simulators (which significantly abstract the channel). In the latter case, an interface between the radio simulator and the network simulator is necessary at the network layer and the physical layer.

The CORE framework and the EMANE framework can be combined by TUN/TAP interfaces at the network layer [21]. The CORE framework can also run in conjunction with underlying ns-3 simulated wireless networks. However, the combination of the CORE and EMANE frameworks or the combination of the CORE and ns-3 frameworks require significant work on the physical layer and the channel due to their abstraction. A possible solution is to combine the CORE and EMANE frameworks or the CORE and ns-3 frameworks with a radio simulator (GNU Radio, CogWave).

B. Combination at the data link layer

The combination of open-source frameworks at the data link layer is shown in Figure 2. The layers above and including the data link layer are provided by the network simulators (OMNeT++, ns-3). The physical layer is provided by the radio simulators (GNU Radio, CogWave). The network and radio simulators are connected by a UNIX socket, named pipe (FIFO), or virtual network interface (TUN/TAP). This combination requires the adaptation of the data link layer provided by the network simulators (OMNeT++, ns-3). Indeed, the OMNeT++ framework abstracts significantly some parts of the data link layer. Ns-3 has the advantage over OMNeT++ to internally represent packets as bit vectors in network byte

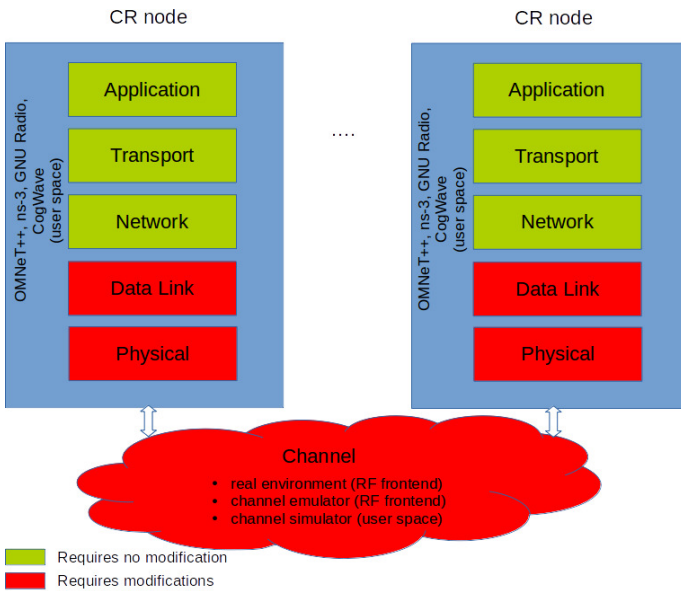


Fig. 3. Combination at the physical layer

order, resembling real-world packet formats [22]. OMNeT++ and ns-3 provide models for wired and wireless protocols which gather the data link layer, the physical layer and the channel into network interface cards (NIC). In this case, the data link layer needs to be extracted from the NIC and extended by an external interface (socket, FIFO, TUN/TAP).

C. Combination at the physical layer

The combination of open-source frameworks at the physical layer is shown in Figure 3. This combination requires either the development of the data link layer, the physical layer and the channel into the network simulators (OMNeT++, ns-3) or the fusion of the radio simulators (GNU Radio, CogWave) into the network simulators. For the first case, some works in the literature have proposed different approaches. In [23], the authors propose an accurate representation of the physical layer of IEEE 802.11 as well realistic channel models in the ns-3 framework. In [24], the authors propose a high-fidelity simulator including the data link layer, physical layer and realistic channel models in the OMNeT++ framework. However, these works do not provide interfaces with hardware devices to test in a real environment (testbed mode) or in an emulation mode for HIL. Moreover, they do not provide the flexibility to replace the data link layer and the physical layer. In [25], [26], the authors propose a high-fidelity network emulation simulation testbed using a combination of a new framework called CREATE (Cognitive Radio nETworking AtchiTecturE), with the CORE framework, the EMANE framework (IEEE 802.11abg) or the GNU Radio framework [27]. However, the CREATE framework is not an open-source framework and does not provide the flexibility to replace the data link layer and the physical layer.

IV. NEW EXTENSIONS TO THE COGWAVE OPEN-SOURCE SOFTWARE FRAMEWORK FOR HIGH FIDELITY SIMULATION OF CRN

We tested the combination of the radio simulator CogWave with the network simulators (OMNeT++, ns-3) at the network layer in a testbed environment and succeeded to execute different protocols and applications (e.g. routing protocols, TCP, UDP, ping application, client-server application). We observed that the combination between radio simulators and network simulators at the data link layer or at the physical layer requires a major rewrite of the different frameworks. We estimated that the combination between radio simulators and network simulators is best done at the network layer (cf. Figure 1), although this combination requires the development of the data link layer, realistic channel models and interfaces in the radio simulators. Therefore, we developed in the CogWave open-source software framework several MAC protocols (Point-to-Point-TDD, Point-to-Point-FDD, Aloha, non-persistent CSMA, 1-persistent CSMA, p-persistent CSMA, TDMA, OFDMA) for the data link layer and realistic channel models (AWGN channel, log-distance path loss, fading channel, frequency-selective fading) for simulating a high number of nodes as well as interfaces with hardware platforms for testbed evaluation and hardware-in-the-loop (HIL). The extensions have been implemented in such a way that the same MAC protocol (data link layer) and modulation scheme (physical layer) can be used without modifications either in a testbed mode with hardware platforms (small number of nodes with real-time scheduling), in an emulation mode (HIL with real-time scheduling) or in a simulation mode with realistic channel models (high number of nodes). In a testbed mode with a hardware platform such as the USRP, the physical layer is connected to an UHD device class which transmit and receive IQ samples in real-time according to the USRP time clock. Another device class could be created for other hardware platforms (e.g. OsmoSDR platforms). In an emulation mode, the physical layer is connected to a virtual device class which transmit and receive IQ samples in real-time according to the computer system time using realistic channel models. In a simulation mode, the physical layer is connected to a simulator device which adds the IQ samples of all the transmitting nodes to all the receiving nodes at a particular time using realistic channel models in simulation-time.

The CogWave open-source software framework has been reorganized into a physical layer directory, data link layer directory and channel directory. The physical layer directory contains the different modulation schemes (e.g. multichannel DAA-OFDM, DADS, OFDM, BPSK, QPSK, GMSK, CPFSK). Performance tests are provided for each modulation scheme for BER evaluation using realistic channel models. The data link layer directory contains the different MAC protocols (Point-to-Point-TDD, Point-to-Point-FDD, Aloha, CSMA, TDMA, OFDMA). Performance tests are provided for each MAC protocol for throughput and delay evaluation. The channel directory contains the different interfaces (UHD device, virtual device, simulator device) and realistic channel models (AWGN channel, log-distance path loss, fading channel, frequency-selective fading).

The network simulators (e.g. OMNeT++, ns-3) provide models for wired protocols (e.g. PPP, Ethernet), wireless protocols (e.g. IEEE 802.11, IEEE 802.16, LTE), routing protocols

(e.g. OSPF, AODV, DSDV, OLSR), network layer (e.g. IPv4, IPv6, ARP), transport layer (TCP, UDP, etc.), and application layer (HTTP, etc.). Moreover, the network simulators (e.g. OMNeT++, ns-3) and network emulators (e.g. CORE) have a real-time scheduler which synchronize simulation time with the computer system time. They can be connected with radio simulators by a raw internet socket with TUN/TAP interfaces at the network layer. Therefore, the new extensions on the data link layer, the interfaces at the physical layer and the realistic channel models in the CogWave open-source software framework can be combined with the networks simulators (e.g. OMNeT++, ns-3) and network emulators (e.g. CORE) to obtain a CRN software framework for high fidelity simulation of CRN. This combination is able to provide libraries of components at all layers of the OSI model (e.g. modulation schemes, protocols, routing algorithms) along with realistic channel models for simulating a high number of nodes as well as interfaces with hardware platforms for testbed evaluation and hardware-in-the-loop (HIL).

Preliminary tests show that the new extensions in the CogWave open-source software framework allow accurate simulation, emulation and connection with RF hardware for all the modulation schemes and MAC protocols which have been implemented. The combination of the CogWave open-source software framework with the network simulators (OMNeT++, ns-3) allows to execute different protocols and applications (e.g. routing protocols, TCP, UDP, ping application, client-server application).

V. CONCLUSION

The simulation of CRN requires a CRN software framework with libraries of components at all layers of the OSI model (e.g. modulation schemes, protocols, routing algorithm) along with realistic channel models for simulating a high number of nodes. At the same time, the CRN software framework should provide interfaces with hardware platforms for testbed evaluation and HIL. Unfortunately, there is no such software framework in the research community. This paper has studied the combination of some open-source frameworks for the high fidelity simulation of CRN. In particular, some challenges were highlighted for the combination of radio simulators (e.g. GNU Radio, CogWave) and network simulators (e.g. OMNeT++, ns-3) at network layer, data link layer and physical layer. New extensions to the CogWave open-source software framework have been proposed for the high fidelity simulation of CRN. These new extensions in the CogWave open-source software framework can be combined with the networks simulators (e.g. OMNeT++, ns-3) and network emulators (e.g. CORE) to obtain a CRN software framework for high fidelity simulation of CRN.

REFERENCES

- [1] J. Mitola III, "Software radios: Survey, critical evaluation and future directions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, no. 4, pp. 25–36, Apr. 1993.
- [2] —, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
- [3] K. Chowdhury and T. Melodia, "Platforms and Testbeds for Experimental Evaluation of Cognitive Ad Hoc Networks," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 96–104, September 2010.

- [4] P. Pawelczak, K. E. Nolan, L. Doyle, S. W. Oh, and D. Cabric, "Cognitive radio: Ten years of experimentation and development," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 90–100, 2011.
- [5] M. T. Masonta, M. Mzyece, and F. Mekuria, "A comparative study of cognitive radio platforms," in *Conference on Management of Emergent Digital EcoSystem*. ACM, 2012, pp. 145–149.
- [6] GNU Radio. [Online]. Available: <http://www.gnuradio.org>
- [7] J. Malsbury, "Modular, Open-Source Software Transceiver for PHY/MAC Research," in *Second Workshop of Software Radio Implementation Forum (SRIF'2013)*, Hong-Kong, China, Aug. 2013, pp. 31–36.
- [8] R. Dhar, G. George, A. Malani, and P. Steenkiste, "Supporting Integrated MAC and PHY Software Development for the USRP SDR," in *IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks*, Reston, USA, Sep. 2006.
- [9] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC Protocol Implementations on Software-Defined Radios," in *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'2009)*, Boston, USA, Apr. 2009, pp. 91–105.
- [10] V. Le Nir and B. Scheers, "CogWave: Open-source Software Framework for Cognitive Radio Waveform Design," in *IST-123 Symposium on Cognitive Radio and Future Networks*, The Hague, The Netherlands, May 2014. [Online]. Available: <https://github.com/vlenircissrma/CogWave>
- [11] —, "DADS with short spreading sequences for high data rate communications or improved BER performance," in *Military Communications and Information Systems Conference (MCC'2013)*, Saint Malo, France, October 2013.
- [12] OMNeT++ Network Simulation Framework. [Online]. Available: <http://www.omnetpp.org>
- [13] S. Khan, M. Kalil, and A. Mitschele-Thiel, "crSimulator: A discrete simulation model for cognitive radio ad hoc networks in OMNeT++," in *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC'2013)*, Dubai, United Arab Emirates, Apr. 2013.
- [14] Network Simulator ns-3. [Online]. Available: <http://www.nsnam.org/>
- [15] A. Al-Ali and K. R. Chowdhury, "Simulating Dynamic Spectrum Access using ns-3 for Wireless Networks in Smart Environments," in *IEEE SECON Workshop on Self-Organizing Wireless Access Networks for Smart City*, Singapore, Jun. 2014.
- [16] M. D. Felice, K. R. Chowdhury, W. Kim, A. Kassler, and L. Bononi, "End-to-end protocols for cognitive radio ad hoc networks: An evaluation study," *Performance Evaluation*, vol. 68, no. 9, pp. 859–875, 2011.
- [17] J. Zhong, "Development of NS-2 Based Cognitive Radio Cognitive Network Simulator," Master's thesis, Michigan Technological University, US, 2009. [Online]. Available: http://faculty.uml.edu/Tricia_Chigan/Research/CRCN_Simulator.htm
- [18] J. Ahrenholz, "CORE: A real-time network emulator," in *Military Communications Conference (MILCOM'2008)*, San Diego, USA, Nov. 2008. [Online]. Available: <http://www.nrl.navy.mil/itd/ncs/products/core>
- [19] "The extendable mobile ad-hoc network emulator" (EMANE). [Online]. Available: <http://www.nrl.navy.mil/itd/ncs/products/emane>
- [20] J. Elsner, M. Braun, S. Nagel, K. Nagaraj, and F. K. Jondral, "Networks In-the-Loop: Software Radio as the Enabler," in *Software Defined Radio Forum Technical Conference*, Washington, USA, Dec. 2009.
- [21] J. Ahrenholz, "Integration of the CORE and EMANE Network Emulators," in *Military Communications Conference (MILCOM'2011)*, Baltimore, USA, Nov. 2011.
- [22] E. Weingaertner, F. Schmidt, H. vom Lehn, T. Heer, and K. Wehrle, "Slicetime: A platform for scalable and accurate network emulation," in *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI'2011)*, Boston, USA, Apr. 2011.
- [23] S. Papanastasiou, J. Mittag, E. Strm, and H. Hartenstein, "Bridging the Gap between Physical Layer Emulation and Network Simulation," in *IEEE Wireless Communications and Networking Conference (WCNC'2010)*, Sydney, Australia, Apr. 2010.
- [24] L. Rose, R. Massin, L. Vijayandran, M. Debbah, and C. Le Martret, "CORASMA Program on Cognitive Radio for Tactical Networks: High Fidelity Simulator and First Results on Dynamic Frequency Allocation,"

in *Military Communications Conference (MILCOM'2013)*, San Diego, USA, Nov. 2013.

- [25] L. Ding, Y. E. Sagduyu, J. Yackoski, B. Azimi-Sadjadi, R. L. J. Li, and T. Melodia, "High fidelity wireless network evaluation for heterogeneous cognitive radio networks," in *SPIE Defense, Security and Sensing Conference*, Baltimore, US, April 2012.
- [26] L. Ding, Y. Sagduyu, T. Melodia, J. Li, J. Feldman, and J. Matyjas, "CREATE-NEST: A Distributed Cognitive Radio Network Platform with Physical Channel Awareness," in *Military Communications Conference (MILCOM'2014)*, San Diego, US, November 2014.
- [27] S. Soltani, E. Sagduyu, H. Li, J. Feldman, and J. Matyjas, "Demonstration of plug-and-play cognitive radio network emulation testbed," in *IEEE Dynamic Spectrum Access Networks (Dyspan'2014)*, Mclean, US, April 2014.