A Framework for the Design, Development and Evaluation of Cognitive Wireless Sensor Networks

Alvaro Araujo Pinto, Elena Romero, Javier Blesa and Octavio Nieto-Taladriz
E.T.S.I Telecomunicacion, Electronic Engineering Dep.
Universidad Politécnica de Madrid
Madrid, Spain
{araujo,elena,jblesa,nieto}@die.upm.es

***Abstract*—Cognitive Wireless Sensor Networks are an emerging technology with a vast potential to avoid traditional wireless problems such as reliability, interferences and spectrum scarcity. Cognitive Wireless Sensor Networks frameworks are a key issue in the future developments of these networks because they allow a lot of protocols, strategies and optimization algorithms to be tested. A framework composed of a network simulator based on Castalia is presented in this paper. This simulator has been improved with cognitive features and feedback from Cognitive Wireless Sensor Networks real devices. This framework allows an easy and intuitive development of complete cognitive networks with spectrum sensing, learning and collaboration features. This is a crucial issue in order to facilitate the design and development of new algorithms, strategies and protocols for Cognitive Wireless Sensor Networks, and evaluate their performance. The benefits of the proposed framework are demonstrated with four different scenarios and simple cognitive communications strategies. Results show how new concepts have been successfully integrated in the framework and how several areas of research could take advantage of it.**

Keywords-Cognitive radio; Wireless Sensor Network; framework; simulator.

I. INTRODUCTION

According to the Wireless World Research Forum (WWRF), seven trillion wireless devices will be serving seven billion people by 2020 [2]. In recent years, Wireless Sensor Networks (WSN) have undoubtedly been one of the fastest growing sectors in wireless and mobile communications. As stated in [3], the WSN market will grow rapidly from $0.45 billion in 2011 to $2 billion in 2021. WSNs consist of spatially distributed autonomous sensors that monitor a wide range of environmental conditions and cooperate to share data across the network.

WSNs are being increasingly introduced into our daily lives. Potential fields of application of WSNs are automatic monitoring of forest fires, avalanches, hurricanes, failures of country wide utility equipment, traffic, hospitals, home monitoring, military operations, critical infrastructure protection, to name a few. The emergence of wireless technologies such as Zigbee and IEEE 802.15.4 has allowed the development of interoperable commercial products, which is important for ensuring scalability and low cost. IEEE 802.15.4 based systems constitute 88% of all WSN deployments up to 2011 [4].

The increasing demand for wireless communication presents the challenge of improving the spectrum utilization efficiency. Most WSN solutions operate on unlicensed frequency bands. In general, they use Industrial, Scientific and Medical (ISM) bands, like the worldwide available 2.4 GHz band. This band is also used by a large number of popular wireless technologies, like Wi-Fi or Bluetooth. For this reason, the unlicensed spectrum bands are becoming overcrowded with the increasing use of WSN based systems. As a result, coexistence issues in unlicensed bands have been the subject of extensive research [5] [6]. In particular, it has been shown that IEEE 802.11 networks can significantly degrade the performance of Zigbee/802.15.4 networks when operating on overlapping frequency bands [6].

To address the spectrum saturation challenge, Cognitive Radio (CR), which enables opportunistic access to the spectrum, has emerged as the key technology. A CR is an intelligent wireless communication system that is aware of its surrounding environment, and adapts its internal parameters to achieve reliable and efficient communications. CR has three main technical features: the cognitive capabilities of its devices, collaboration among terminals and the fact that they learn from their history. CR differentiates between two kinds of users; Primary Users (PUs) are the licensed users, and Secondary Users (SUs) are those who try to use the same bands when they detect a spectral hole.

Adding cognition to the existing WSN infrastructure will bring about many benefits. In fact, WSNs are one of the areas with a highest demand for cognitive networking. In WSNs, the node resources are constrained mainly in terms of battery and computation power, but also in terms of spectrum availability. Hence, with cognitive capabilities, WSNs could find a free channel to communicate either in the unlicensed or the licensed band. However, the cognitive technology will not only provide access to new spectrum bands but will also achieve better propagation characteristics. By adaptively changing system parameters like modulation schemes, transmit power, carrier frequency and constellation size, a wide variety of data rates can be achieved. This will certainly improve power consumption, network life and reliability in WSNs.

The Cognitive Radio Wireless Network (CWSN) field has not been fully explored yet. Real or simulated scenarios rarely exist. The non-existence of a complete simulator for CWSNs contributes to the scarce appearance of results in the area. In order to enable the design and development of new algorithms, strategies or protocols for CWSNs, and evaluate their performance, simulation and emulation environments are necessary.

As for common WSNs, the initial research of CWSNs is usually done with a WSN simulator. Simulators help developers to avoid possible failures in hardware. Cost and time reduction is another advantage of simulators. There are a lot of WSN simulators that will be further explained and referenced in Section II, but a complete CWSN simulator does not exist. Therefore, in this work we propose a new framework for the development of CWSNs. It consists of a complete CWSN simulator based on Castalia, adapted and improved with cognitive modules.

However, the challenge in simulators is to determine if its simulations achieve a good enough correspondence with real deployments. In this work, the simulator is fed with data provided by real CWSN devices to obtain a more realistic approach. Therefore, a complete simulation framework for CWSNs using regular standards is presented.

This framework can be used to simulate different approaches, such as strategies to reduce energy consumption in CWSNs (to increase lifetime), countermeasures for CWSN specific threats (to increase reliability) or various spectrum sensing strategies.

This paper is organized as follows. In Section II, works in the CWSN simulator and emulator frameworks are reviewed. In Section III, the new CWSN framework is described. In Section IV, a proof of concept is shown. Finally, the conclusions are drawn in Section V.

II. RELATED WORK

Because of the novelty of this research field, there are not many specific frameworks for the design of green communications over CWSNs. It is natural that most of the works are based on WSN simulators.

As can be seen in the previous published work [1], CWSN research is still in the first stage, the design. The second stage requires suitable tools for the simulations. Nowadays, these tools do not exist or they do not have the required characteristics to develop a complete CWSN scenario. There are several WSN simulators used by researchers to develop their work. For example, NS-2 [7] is one of the best-known simulators. The majority of the WSN research society uses this simulator, although its latest release took place in 2008. NS-2 is a discrete network simulator built as an Object-Oriented extension of Tool Command Language and C++. This simulator is open source and provides online documentation. NS-2 can support a considerable range of protocols in all layers.

Another WSN simulator is TOSSIM [8], which is an emulator specifically designed for WSNs running on TinyOS, an open source operating system. This is a very simple but powerful emulator. EmStar [9] is a trace-driven simulator running in real time, specifically designed for WSNs and built in C. EmStar has a robustness feature that can mitigate faults among the sensors. It also provides a lot of modes, making debugging and evaluation much easier. OMNET++ [10] [11] is another very well-known framework among researchers. It proposes a modular library which can be used to develop network simulators. Just by arranging different modules, the developer can create his/her own simulator or scenario.

Several other simulators have been developed for WSNs [12], such as COOJA, OPNET, NetSIm, J-Sim, ATEMU, Avrora, QualNet, etc.

None of the above simulators have cognitive features. As it was described in the previous section, a cognitive network has two main characteristics:

- Maintaining awareness of its environment, including the spectrum.
- Optimizing its radio parameters according to the requirements.

These two main features could be divided into several derived characteristics. Any CWSN simulator should be implemented taking into account these necessary characteristics. In the past, some attempts to develop a CWSN simulator have been made. The first approach has been to develop cognitive features over an existing WSN simulator. In [13], the authors propose a new routing model for cognitive networks over NS-2. With this goal in mind, changes to the NS-2 architecture are explained, such as support for multiple channels. Multiple channel support is one of the first changes that any simulator should handle. However, the implemented changes in [14] are not enough because only a few radio parameters, such as transmit power and propagation, can be changed. Besides, it is impossible to use more than one radio interface at the same time with this cognitive layer for NS-2. Moreover, other CWSN aspects like power management, collaboration, scalability and learning are not mentioned or implemented in this NS-2 model.

Despite the limitations of the NS-2 simulator in cognitive scenarios, multiple researchers have chosen NS-2 for their investigations because it is the only WSN simulator with some cognitive features. For example, the work in [15] presents an algorithm to optimize the route selection in a disaster situation. The main idea is that the wireless nodes sense the spectrum. According to some parameters such as latency, jitter or packet error rate, the nodes choose the optimal communication interface between them. The decision is made individually and is transmitted by satellite communications. This scenario is simulated in NS-2 using nodes with three wireless interfaces. The interfaces change during the simulation time. That is, the nodes cannot use more than one interface at the same time. However, the work does not explain how the spectrum sensing works, and collaboration and learning do not exist. Finally, the scenario presented in this work is a cognitive application with different traffic and protocols than the ones from WSNs.

Another work is presented in [16]. It evaluates a solution for coexistence with wireless LANs, based on a new MAC layer. The simulations are implemented over NS-2, but it does not focus on WSNs. The authors describe how the cognitive nodes sense the channels and agree on the active channel. In order to make this decision, they use a control channel. In this work, the simulator architecture was not modified and the cognitive features are poorly explained. For example, they talk about predicting the length of a spectrum hole. However, it is not explained how this data is obtained. Power management is not presented and the scalability is untested. In addition, only a simulation with 10 nodes is provided.

Apart from the NS-2 simulator and its new cognitive features, some cognitive simulators have appeared over the past few years. The following simulators have not been implemented for WSNs but they include some cognitive features. For example, in [17], the platform presented focuses on spectrum sharing. For that reason, the physical layer and the spectrum resource manager are the modules that implement cognitive features to avoid primary-user collisions. Important characteristics of WSNs such as mobility, consumption and protocols are not in the scope of this simulator.

The authors of [18] present some software to simulate cognitive networks scenarios. They divide their architecture into five modules: the scheduling module, the mobile node module, the statistics module, the wireless environment module and the interface. The cognitive features are implemented in the mobile radio module and the statistics module. Among the mentioned implemented characteristics are spectrum sensing and information storage. Collaboration between nodes is not explained. As in the work in [17], WSN key features are not presented.

Another example of a cognitive simulator is presented in [19]. It focuses on the definition of an Autonomic Communication Element (ACE) architecture. The architecture is not for WSNs and the development is done in RuleML language. The authors present an interesting approach to cognitive nodes. They define some modules that represent the cognitive features. Some of them are the spectrum sensing or the experimental database modules. However, the authors do not explain the implementation of the cognitive strategies in detail and no results or evaluations are presented.

The emerging problem of spectrum saturation in WSNs that we explain in the introduction and the current state of cognitive simulators provide the motivation for this work. Only NS-2 supports today's cognitive characteristics in WSN scenarios, but it has a lot of limitations. The other simulators present more cognitive features but they obviate the WSN ones.

After the simulation stage, researchers generally use a test-bed, before the real implementation. There are multiple test-beds for specific developments. TWIST [20] and VT-CORNET [21] are the most important test-beds nowadays because of their general purpose features and their quality.

The TKN Wireless Indoor Sensor Network Testbed (TWIST) is a multiplatform, hierarchical test-bed architecture developed at the Technische Universität Berlin. The self-configuration capability, the use of hardware with standardized interfaces and the inclusion of open source software make the TWIST architecture scalable, affordable, and easily replicable. The TWIST instance at the TKN office building is one of the largest remotely accessible test-beds with 204 System Under Test (SUT) sockets, currently populated with 102 eyesIFX and 102 Tmote Sky nodes. The nodes are deployed in a 3D grid spanning 3 floors of an office building at the TUB campus, resulting in more than 1500 m^2 of instrumented office space.

The Virginia Tech COgnitive Radio NEtwork Testbed (VT-CORNET) is a collection of Cognitive Radio nodes deployed throughout a building at the Virginia Tech main campus. The test-bed consists of a total of 48 Software-Defined Radio nodes. It is implemented with a combination of a highly flexible RF front end, and an openly available Cognitive Radio Open Source System framework.

The ORBIT project, launched in 2003 [22] is a large-scale open-access wireless test-bed. It can be used by the research community working in new wireless communications. In some aspects it is similar to the TWIST test-bed, a large deployment of wireless nodes with spectrum sensing capabilities, but it lacks the possibility of different radio interfaces combined into the same node.

Research on CWSN simulators is emerging, but it is still in a primary state. A simulation with a high number of nodes is necessary in WSN scenarios. It is very expensive to build a lot of real devices to test a concrete low-power strategy. The integration of real data devices and a high number of nodes is only possible using a feedback relation. Currently, there is not a CWSN simulator with standard protocols and feedback from real devices that uses cognitive characteristics for an intelligent energy management in order to test new policies, assess collaboration schemes or validate different optimization mechanisms.

SENDORA, the only simulator with cognitive capabilities does not use real device data for the power model.

Other simulators like NS-2 lack cognitive capabilities such as learning, using different radio interfaces or manage collaboration between nodes. Therefore, an implementation of a completely new cognitive module over an existing WSN simulator, specifically the Castalia Simulator (based on the OMNET++ framework), and a new CWSN device with three different radio standard interfaces are proposed.

III. CWSN FRAMEWORK

Most common network simulators have tested energy models, but these are theoretical models covering general cases. So, it is necessary to introduce real data measured by a cognitive radio prototype developed to make these simulations more realistic. Thus, it is also possible to find differences in commercial solutions using the same technology.

Moreover, the deployment of a network of real devices is very difficult and expensive, especially for a network with a large number of devices. This is the great advantage of the introduction of simulators. By adding data taken from functional prototypes to simulation results, the accuracy of the simulations improves.

Thus, the combination of both elements results in a complete and useful framework to validate optimization mechanisms for energy consumption.

As seen in Section I, cognitive characteristics are applicable to intelligent energy management. Thus, it is important to provide a CWSN framework to test new policies, to assess collaboration schemes and to validate different optimization mechanisms.

The CWSN framework is composed of two fundamental elements: a network simulator and low-power cognitive radio real devices.

A. *CWSN Simulator*

The CWSN simulator described in this section is based on the well know Castalia simulator, which is in turn based on OMNET++. As it can be seen in Section II, the amount of WSN simulators is very large. Those, along with the fact that attempts to create a cognitive simulator have not reached a decent level of development, have led us to create our own simulator based on a WSN simulator. The decision about which simulator was better was made according to [23], focusing on these reasons:

- The Castalia simulator focuses on WSNs. This feature is very important because of the scope of the simulator. Despite Cognitive Radio Networks (CRNs) having multiple applications and scenarios, this work is focused on CWSNs.
- The Castalia simulator is based on OMNET++, which has a modular and simple implementation. If the goal of this work is to develop a cognitive architecture inside the simulator, new modules and interfaces will be included. OMNET++ makes these additions very easy.
- The Castalia and OMNET++ development is very active with releases every few months. The work is based on Castalia and OMNET++ in order to create a usable tool for any cognitive project. The other important simulator for cognitive scenarios, NS-2, has not received a new release since November, 2011 and the one before that was in 2009.
- Castalia includes a resource manager module in order to monitor parameters such as energy or memory consumption in the nodes.
- The Castalia physical layer and radio models are some of the most realistic models that any researcher can find in the simulator field. As cognitive radio is based on spectrum sensing, a realistic physical layer is an important advantage.

Emphasizing the physical and radio layer, Castalia offers multiple characteristics such as path loss, mobility in the nodes, simple interferences, multiple modulations and sleep states. The cognitive simulator can use all these features in order to create more realistic scenarios.

Having chosen the simulator, the next step is to define the requirements needed by the cognitive simulator, so that it can offer enough features for future works.

1) *Requirements*

The three main characteristics of cognitive radio are environment awareness, learning, and acting capacity. All the requirements imposed on this simulator try to implement or to improve these characteristics.

- Spectrum sensing. If the cognitive nodes have to be aware of the context, they need to extract that information from the spectrum.
- Multiple frequencies, channels and modulations. An essential characteristic needed to reach the flexibility of a cognitive network is to introduce the possibility of changing between multiple frequencies, channels and modulations.
- Virtual Control Channel (VCC). As we will explain later, the VCC allows the nodes to share information.
- Primary and secondary users. The two roles present in cognitive networks have to be implemented in the simulator too.
- Information storage and learning. The cognitive nodes learn from the captured information.
- Results and data representation. They are essential for the analysis of the results.

Although the Castalia simulator's physical layer is one of the best compared with other simulators, the sensing block is critical for cognitive networks. Consequently, some changes need to be made to improve the sensing stage.

The Castalia simulator supports most common modulations but it is also prepared to include new ones. Moreover, some typical radios for WSNs are included, such as CC1010 or CC2430. Interferences are another important aspect of the sensing module. Noise detected in the spectrum can affect the network's behavior. For that reason, the interference model must be very precise.

Section II shows some attempts to implement multiple radios and multiple channels in simulators. There can be no doubt about the importance of supporting different real wireless radio interfaces in each node, allowing changes in all parameters: modulation, transmission power, consumption, frequency, etc. Cognitive networks differ from other types of networks due to the adaptation of their parameters according to the information gathered from the environment. Although the Castalia simulator presents the possibility of a limited spectrum sensing, it is not enough for a cognitive network. Multiple changes are necessary in Castalia, starting from a complete spectrum sensing, following with the storing of this information and concluding with the spread of the information which is an important feature of cognitive networks. A Virtual Control Channel (VCC) has been implemented for this purpose.

Normally, WSN simulators make differences in the nodes only when a particular technology forces them. For example, coordinators and end nodes on ZigBee protocol. In these cases, the differences are related to the functionality of the network. However, cognitive networks introduce two roles for all the CRNs: Primary Users (PUs) and Secondary Users (SUs).

Finally, when the simulator executes an application or scenario, the developer needs a simple way to extract the results. Moreover, the number of parameters that the developer can monitor needs to be the highest possible. For this requirement, changes in the resource manager block are necessary.

Once the requirements have been explained, the CWSN simulator is going to be described in detail.

2) Cognitive Radio extension for Castalia

In this work, the structure of Castalia has been modified in order to provide the simulator with Cognitive Radio support. Figures 1 and 2 show the new simulator structure. The code has been modified as little as possible in order to introduce the minimum changes to third-party applications and module implementations.

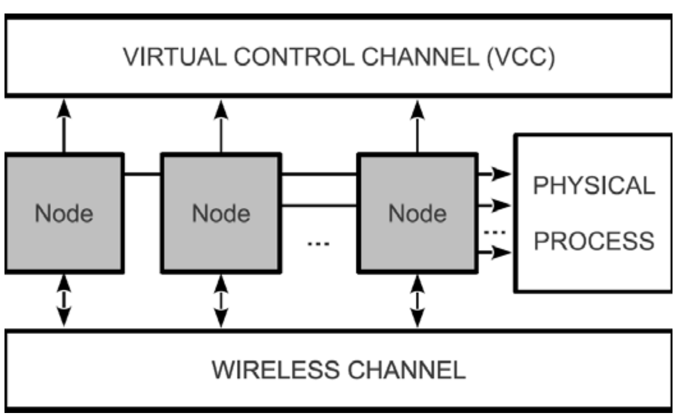

Figure 1. Castalia network architecture adapted to Cognitive Radio

In the new model, the nodes have multiple communication modules which can be configured with different parameters. They simulate the multiple interfaces of a wireless node. Every interface is connected to the application module and the wireless channel. The new simulator provides the developer with functions to change the default interface used to send data. It provides complete backwards compatibility so previous non-cognitive samples and modules do not have to be modified.

A node with multiple interfaces brings flexibility to the network in a lot of aspects: comparison of performance and consumption between technologies, protocols and modulations, cognitive strategies that imply two or more radios, and freedom to change the parameters of each interface independently.

The existing differences in the scenario configuration file of a multiple interface experiment can be seen in the following code:

SN.numIFaces = 2
...
SN.node[*].Communication[0].Radio.RadioParametersFile = "CC2420.txt"
SN.node[*].Communication[1].Radio.RadioParametersFile = "ZigBee.txt"
...

According to these lines, the nodes would have two interfaces. The first one is a CC2420 node and the second one is a generic ZigBee node. Each one can have a different transmission power, different carrier frequencies or a different modulation.

The parameter *numIFaces* indicates the number of interfaces per node. By default, this parameter is one due to backwards compatibility. Since it is possible to manage more than one communication simultaneously, developers must specify the interface they are referring to.

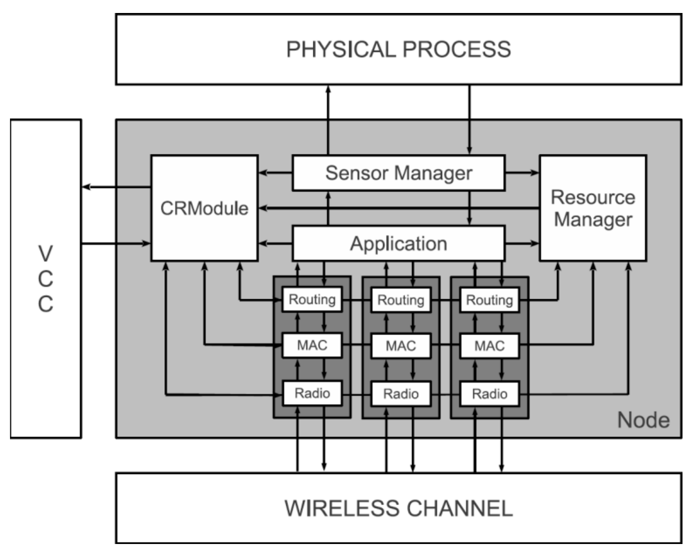

Figure 2. Castalia inner blocks adapted to Cognitive Radio

The Radio module of each communication module provides new API methods for changing the active channel. This change lets developers perform spectrum scans and hops among channels easily. The channel changing feature completes a set of modifications in order to increase the flexibility of the network but also to complete wireless protocols such as Wi-Fi or ZigBee, in which nodes have multiple channels.

Another implemented change in the simulator is the creation of PUs and SUs. Most cognitive applications have both roles, where PUs have preference in the use of the spectrum and SUs try to take advantage of spectrum holes. The application layer is responsible for providing this feature.

The new functionalities have been carried out with the minimum number of modifications to the public API of Castalia, so developers can keep on using the same experiments without changing a large amount of code.

These changes transform Castalia into a simulator capable of running Cognitive Radio experiments, although it still lacks any cognitive capabilities. In order to turn Castalia into a real cognitive simulator it has been equipped with a new module that includes all the cognitive features of the nodes.

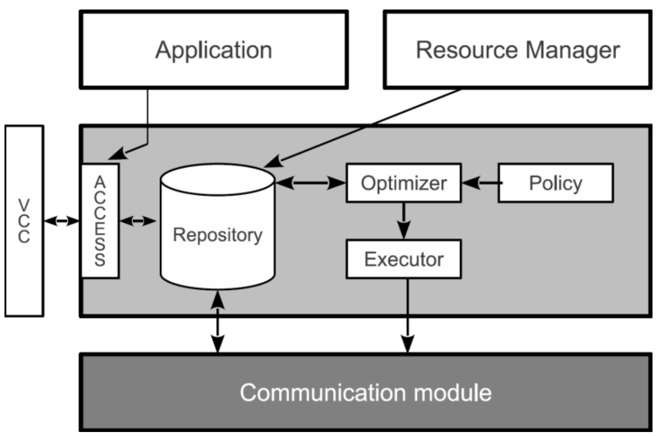

Figure 3. Cognitive Radio Module structure

The CR module structure is shown in Figure 3. Each module is composed of the following elements, extracted from the work in [24]. These modules have been adapted to the existing Castalia structure and provide the developer with a lot of possibilities for the creation of different scenarios. Therefore, multiple interactions between the next modules exist.

a) Repository

An essential requirement for an effective cooperation and collaboration is that the cognitive nodes make the learned information, the decisions made and the current state, available to all interested parties. These neighbors may be secondary users that cooperate in order to make better decisions or to optimize some policies. This is enabled by means of a distributed repository structure. The nodes store the information they capture in the repository and eventually they also store information from other nodes when they need it. Each node publishes part of its own repository to the network, making it public through the VCC. When a node requires information from another node's repository, it sends a request packet through the VCC. If the information is available and public, the access will be granted.

The kind of information stored depends on the context and the requirements of the system. Some of the modules that feed the repository with information are the communication modules, the application, the resource manager and the optimizer. The repository complements the resource manager module. The resource manager stores information about general characteristics of the nodes and the network such as power and memory consumption. The repository inside the CR module saves information related to cognitive features such as sensing, learning or strategies. The repository is the backbone of the CR module framework and the fundamental component that enables cooperation and dynamic information exchange among cognitive wireless technologies.

An example of the use of the repository could be a collaboration strategy in which nodes learn about the usage of multiple interfaces and frequency bands. When this learning is complete, the nodes can use this information to transmit over empty channels saving energy and improving communications.

b) Access

Information stored in the repository can be an important source for malicious intentions. For this reason, or simply because of the general goal of the network, the access module does not let all nodes access the repositories. The access module controls which part of the repository is public and which nodes are allowed to access it.

As we have said before, security is completely associated with the access module. An example is an application in which nodes store the reputation of the network nodes. If the attacker accesses the repository and increases its own reputation, the attack can affect the behavior of the network.

c) Policy

This module enforces the requirements for the global system depending on several factors: energy consumption, interferences or noise, quality of service, or security. In simple terms, the policy module is a set of weighting parameters that control the priority of the different network goals. The nodes act according to the final composition of services and weights. These policies and weights may vary dynamically and the nodes should be consistent with these variations.

For example, a CWSN can be responsible for monitoring a large forest area. The primary policy could be low energy consumption, but when a fire is detected, the policy changes to offer the best QoS to transmit the alarm.

d) Optimizer

It processes the repository information bearing in mind the requirements imposed by the policy module. Decisions regarding the behavior of the local node are the results of processing. They are stored in the repository and evaluated by the executor module. To summarize, the optimizer makes decisions according to the stored information.

Probably, the optimizer is the most complex submodule of the CR module. To understand how it works, an example is presented: in a scenario where security is the predominant policy, a new node wants to join the network. The optimizer of the coordinator analyzes the information (stored in the repository) about the new node, namely location and transmission power, and it makes the final decision allowing or not the new node to join the network.

e) Executor

The decisions made by the executor need to be distributed to the modules responsible for modifying the parameters. The executor usually sends orders to the communication module, where the radio parameters can be changed or the routing protocol can be modified.

f) Virtual Control Channel

The Virtual Control Channel (VCC), a new method for sharing cognitive information among the CR modules of the nodes, has been included in the architecture. CR modules can access exported information of remote repositories through this channel. It allows the CR modules to be aware of their surroundings and even of the whole network. The VCC gives the nodes a common interface to communicate, ignoring the details of how the data is delivered, and the precise nature and location of the communication partners.

Since all the elements are developed as Castalia modules, they communicate and access each other via the OMNET++ message system. The modularity of OMNET++ and its high level portable language makes this architecture very easy to transfer to a real device. Usually, the standard protocols of WSNs for real devices have resources to create these modules. For example, the repository message can be implemented in the stack of multiple protocols, such as ZigBee. The definition of different interfaces complicates the integration work but it is completely possible.

3) Changes in radio module

Most of the work in this project focuses on developing the cognitive radio module that introduces cognitive behaviors into the simulator. However the Castalia simulator has some characteristics that can be improved.

As we said before, the Castalia simulator only supports one radio interface per node. Nodes can have some of the different MAC layers that Castalia includes: 802.15.4, tunable MAC, etc. However, none of these MACs implement different channels. The goal is that every interface could have a different channel bandwidth, first and last frequencies and number of available channels. By combining multiple channels and multiple interfaces, scenarios are very realistic.

Spectrum sensing is a key factor in cognitive radio. Nodes must analyze the spectrum to detect primary users or to find the best medium to share information with other secondary users. The decision about which channels or interfaces are the best at each time should be based on realistic and plentiful data. For this reason the interference model on the Castalia simulator has been changed. Before these changes, a node only detected a packet if the transmitter operated in the exact same carrier frequency as the receiver. If the frequencies were different, the packet was dropped and it did not create any interferences. Now, the model is more realistic and the packets create interferences if they are within the signal bandwidth. These interferences are proportional to the distance between carriers and are related to the modulation.

Finally, some minimum changes have been made to the resource manager block. As explained before, the resource manager controls the node parameters such as the energy spent or the memory occupied. In accordance with the idea of having the least possible parameters modified in the simulator, changes to control the power consumption of the multiple interfaces have been made.

B. CWSN devices

A test-bed platform to develop cognitive radio communications for WSNs and to obtain its energy consumption model data has been implemented (Figure 4).

The CWSN device tries to optimize communications in real time according to different application needs. Therefore, the device design has to consider power consumption, data rate, reliability, and security in order to be useful for a large number of applications.

For our goal, energy consumption is a very important challenge. It is necessary to control the consumption of each separate component, and to implement shared strategies that try to reduce the overall consumption of the network.

Interference with other wireless devices or noise problems have to be avoided, which implies that nodes have to change their frequency and modulation as fast as possible. For this reason the prototype has three different network interfaces. The reduction of interference can be an important factor to reduce the overall consumption of the network.

CWSNs need to be connected to different kinds of standard commercial devices or internet gateways. Consequently a widely extended wireless solution as an interface has to be implemented.

This prototype has to be capable of collecting data about the state of the network and of sharing the information with other nodes. In addition, each node has to be able to change protocol parameters, the entire protocol and wireless interfaces in real time. Thus, it is mandatory to coordinate all the network devices.

Figure 4. Cognitive Wireless Sensor Network Device prototype

The control function is made by a Microchip PIC32MX795F512H, which is a 32-bit flash microcontroller. It is a high performance processor with low consumption and low cost. The CWSN platform has three radio interfaces:

- A Wi-Fi Microchip device (MRF24WB) which can handle data rates of 2 Mbps and operates in the band between 2.412 and 2.484 GHz. Wi-Fi is based on the IEEE 802.11 standards.
- A MiWi interface, a Microchip protocol which can handle data rates of about 250 kbps and operates in the band between 2.405 and 2.48 GHz. This is a proprietary wireless protocol designed by Microchip Technology that uses small, low-power digital radios based on the IEEE 802.15.4 standard for WPANs. The device used is the MRF24J40MA.
- The last interface is based on the Texas Instruments CC1010. It can handle data rates of 76.8 kbps and operates in a band around 868 Mhz. This interface provides a new communications band in an ISM frequency.

The software has to be able to discover other nodes, sense the radio-electric environment, exchange configuration information, establish communication channels, turn the radio interfaces on or off, and manage the active or asleep state of the node. The network manages data routes optimizing consumption, data rate, reliability and security.

Three wireless interfaces have been used in this device, each with different standards and protocols. The integration of a new interface or device in the consumption model of the simulator is very easy. The only thing necessary is to fill the file with the real device measures.

IV. Demonstrative Use of the Framework

In this section, the results of simulations related to the cognitive framework design are presented. It is not the goal of the simulation to prove the algorithm or mechanism itself. The goal is to check that several new policies, collaboration schemes or optimization mechanisms can be implemented in this framework.

The presented architecture permits the simulation of complete cognitive scenarios with the following characteristics:

- Scalable cognitive networks with a large number of sensors.
- Multiple interfaces and channels in the nodes at the same time.
- Node mobility.
- Resources control such as battery energy.
- Spectrum sensing for multiple channels and interfaces.
- Control communication and collaboration among nodes.
- Knowledge database.
- Learning in execution time.
- Application of multiple policies.

Some different scenarios have been developed in order to test the cognitive framework.

The first scenario demonstrates the new concept of changing the transmission parameters and the energy consumption optimization. The second scenario is similar to the first, but it includes different applications. The third scenario shows how the cognitive feature of spectrum sensing can be useful for some optimizations. Finally, the fourth scenario shows how the spectrum sensing, the repository and the VCC can be used to implement complex cognitive simulations.

The first scenario is related to the capability of changing transmission parameters. It is composed of five nodes, each with 802.11 and 802.15.4 radio interfaces. Four nodes send data to a central node. In this scenario the nodes simulate two different applications. The first one is a multimedia application where both the bit rate and the packet size are high. The transmission rate needs a Wi-Fi interface while a WPAN does not have the capacity for multimedia applications. However, in a WSN, general applications only have monitoring functions (temperature, light, etc.) and the bit rate and the amount of information are very low. In this case, the low-power optimization strategy consists of using the interface with the lowest energy consumption for a specific data rate. When the data rate is high, only 802.11 is possible, but for some specific data rates 802.15.4 is better because of its lower energy consumption. This algorithm could be dynamically changed according to other constraints such as battery life, distance between nodes or quality of service. Real data is used in the power model from a

MRF24J40MA-based device for the 802.15.4 protocol and from a MRF24WB-based device for the Wi-Fi transmissions. In the simulation, power measures from the CWSN device are included (Wi-Fi transmissions consume 74.8 mW while WPAN transmissions reach 3.6 mW). As shown in Figure 5, when the data rate is high, 802.11 is used for the transmission, but when the data rate decreases, 802.15.4 is better because of its lower energy consumption. The second part of this figure (zoomed in Figure 6) shows the consumption with the Wi-Fi and WPAN common sensing application with the same packet size and the same interval between messages.

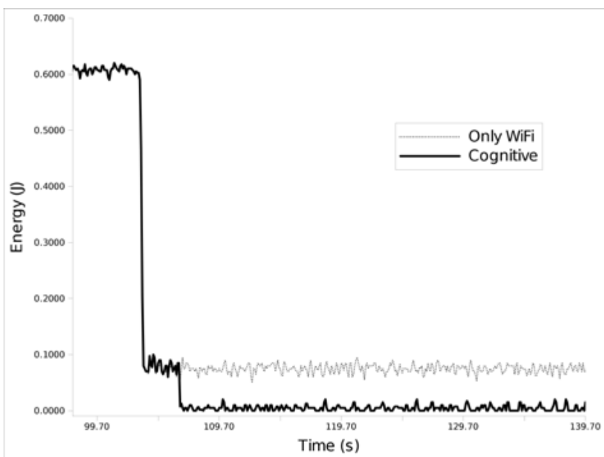

Figure 5. Energy consumption for the Cognitive algorithm and Wi-Fi

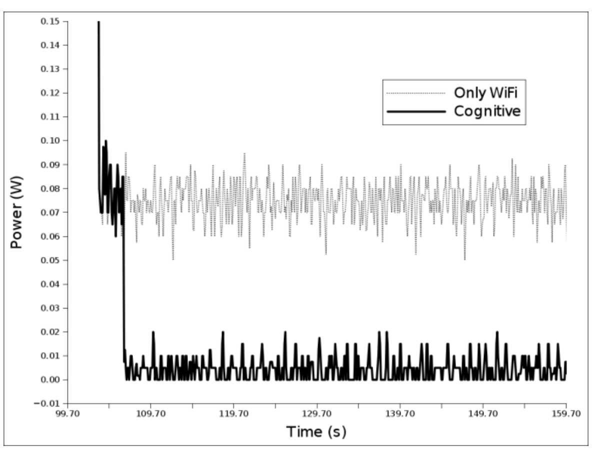

Figure 6. Detail of the energy consumption for the Cognitive algorithm and Wi-Fi

Using a low-power protocol system saves 94% of the energy (Figure 5). Only in the commutation period, where the nodes need to communicate the interface change, the consumption is similar to Wi-Fi. After that, the energy saving is considerable.

The second scenario simulates an application whose nodes send packets with the maximum payload allowed by the simulator (1000 bytes with 802.11 and 100 bytes with our implementation of the WPAN protocol). The application starts sending a package every 10 ms and this time is increased until the bit rate reached by the 802.11 interface is supported by the WPAN protocol (reached at time 600). Figure 7 shows how the consumption of the WPAN interface in the first period of the simulation time is greater than the one of Wi-Fi. This is because the WPAN protocol needs more transmissions for the same amount of data. This means that using 802.15.4 does not automatically reduce the consumption of every application with a low bit rate, but a cognitive module choosing the right protocol at every moment can achieve that goal.

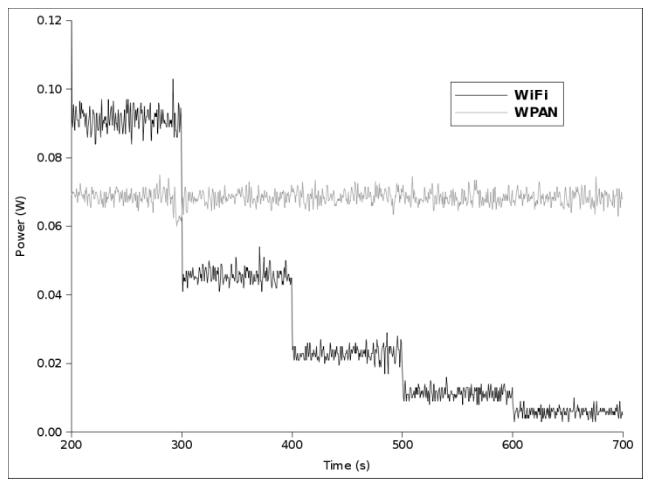

Figure 7. Energy consumption for the Cognitive algorithm and Wi-Fi (Scenario 2)

The third scenario shows the optimization by means of spectrum sensing. It consists of two nodes with 802.15.4 radio interfaces. One of them, the receiver node (B), moves (in a total time of 70 seconds) through space and the other, the transmitter node (A), is fixed (Figure 8). Within the path of movement experienced by the mobile node, sometimes node B will be closer to node A than others. In a common network design, node A will transmit information with a fixed power. That can cause certain packets to be lost (because of the distance between nodes) and others to be transmitted with more power than necessary. Adding cognitive capabilities to this scenario, the network could be aware of the minimum power necessary to ensure the reception of packets while minimizing energy consumption. For this simulation, a power model real data from a MRF24J40MA-based device for the 802.15.4 protocol has been used.

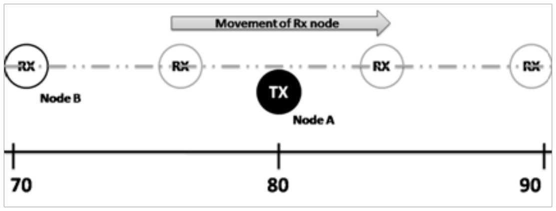

Figure 8. Mobile node scenario

In Figure 9, the instantaneous energy consumption of the transmitter node (node A in Figure 8) is shown. The dotted line represents the consumption of node A in a network without cognitive capabilities and the solid line shows the consumption of the same node when the low-power consumption algorithm is applied. Modifying the transmission power in relation to the distance between nodes can reduce the energy consumption. Using this simple algorithm implies a reduction of up to 60% in some sections.

Increasing the complexity of algorithms or dealing with the problem of consumption in a holistic way (combining several techniques), it will be possible to obtain higher reductions.

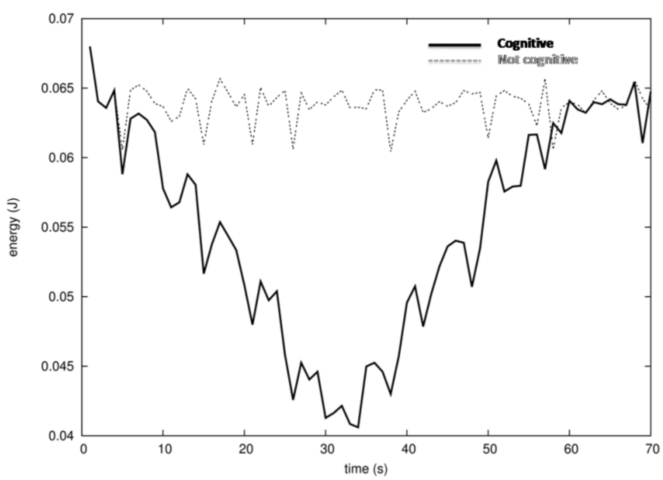

Figure 9. Energy consumption for the Cognitive algorithm and 802.15.4 (Scenario 3)

The fourth scenario is a demonstration of a complex cognitive application where all the cognitive features implemented in the simulator are used to detect anomalies such as broken nodes or intruders.

The simulation consists of 50 nodes with different roles. There is a server that periodically receives information from multiple sensors. The sensors are PUs if they have priority or SUs if they do not. Moreover, SUs collaborate in order to detect anomalies in node behaviors. More specifically, they detect these anomalies by sensing the spectrum and learning about the transmission power of each node.

During the first seconds of the simulation the nodes learn about the transmission power: number of transmissions, average power transmitted and its standard deviation. This information is stored in the repository inside the cognitive module. After these seconds, the nodes start to detect differences in this learned parameter. The optimizer notifies the application if some anomaly happens, and the application, after multiple warnings, retransmits the alarm to the rest of the nodes. When a fixed number of nodes agree on the anomaly of some node, it is marked as an abnormal node and it cannot further participate in the working of the network.

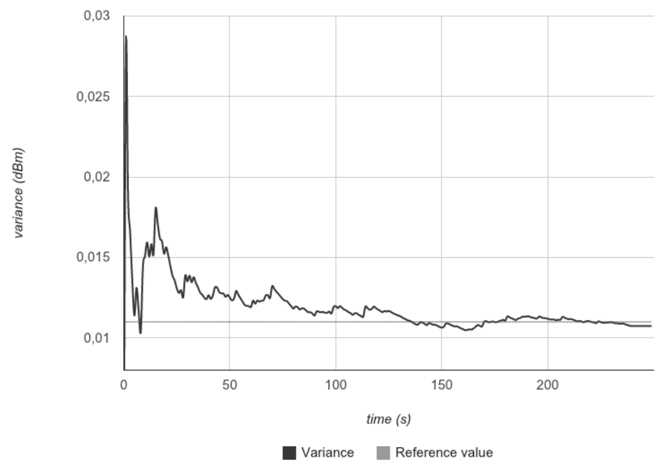

Figure 10. Sensing power and learning average

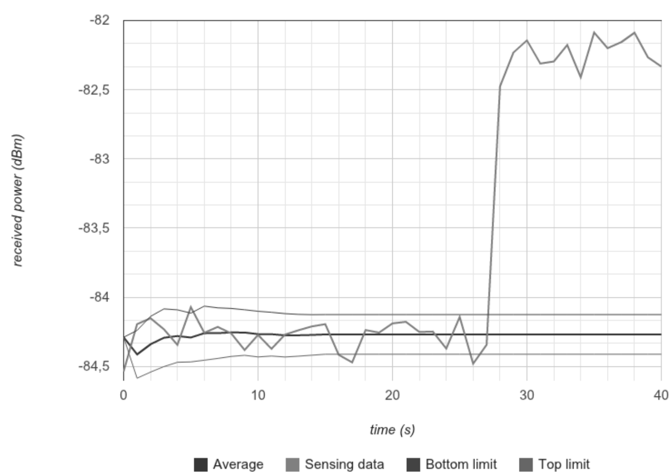

Figure 11. Learning power variance

Therefore, spectrum sensing allows the nodes to detect and analyze the received power; the repository stores all the learned information, the optimizer analyzes the data according to some policy and finally, the nodes collaborate using the VCC.

The results of a network with 50 nodes are shown in Figure 10 and Figure 11. Figure 10 represents how a SU learns about the transmission power of another node that has an abnormal operation. Sensing power is the power received by the node when it receives a new packet and the learning average is the average of all the packets received by this node. The x axis represents the number of packets that the node has received since the beginning of the simulation.

As we can see, the average is stable with a few samples. The top limit and the bottom limit form a range where the sensing data is considered normal. When the data is out of the limits, the node interprets it as an anomaly.

We can observe how fast the system learns. With a few samples, the variance fluctuates but when the node has more information the variance stabilizes over 1%.

Figure 11 shows the scalability of the solution with 50 nodes in the same network. Moreover, it shows some of the most important features such as spectrum sensing, learning and knowledge storage.

Beside these four scenarios, the simulator has been tested in other works such as in [25], where a security strategy has been implemented over the simulator. Collaboration, security policies, scalability and spectrum sensing are the most important factors in this work. Another example of the use of this simulator is presented in [1]. This work focuses on the use of the cognitive features of the simulator to implement green scenarios with energy saving strategies.

V. Conclusion And Future Work

The new cognitive paradigm has appeared to cope with very important network problems like spectrum scarcity, interference or reliable connections. Cognitive network features expose new interesting research challenges. The implementation of cognitive features in WSNs is an emerging field with specific characteristics to explore, like new energy saving strategies or new security approaches. CWSNs are still in the first stage of a project, the design. The second stage requires the suitable tools for the simulations and implementations.

At this moment, it is important to have a CWSN framework to test new policies, to assess collaboration schemes and to validate different optimization mechanisms. In this article a CWSN framework is presented. The framework is composed of a network simulator and low-power CWSN real devices. A new cognitive module has been developed over the Castalia simulator and different real interfaces and energy consumption models have been integrated. A CWSN platform has been built using a microcontroller and three different radio interfaces (IEEE 802.11, IEEE 802.15.4, and a CC1010-based interface in the 868 MHz band). This framework uses real devices' implementations to measure different power and transmission characteristics that are included in the energy consumption model and the communication model. This feedback achieves simulation results that are closer to a real scenario than the ones from regular simulators.

The benefits of the proposed CWSN framework have been demonstrated by implementing four scenarios. Low-power optimization strategies, cognitive applications and security approaches have been implemented using this framework. Results show how cognitive concepts (collaboration, learning, different communication parameters) have been integrated in the simulator with a positive outcome.

In conclusion, this framework represents a good opportunity for the development of new cognitive wireless communication strategies for the new paradigm of CWSNs.

Acknowledgment

This work was funded by the Spanish Ministry of Science and Innovation through the Secretariat of State for Research, under Research Grant AMILCAR TEC2009-14595-C02-01, and through the General Secretariat of Innovation under Research Grant P8/08 within the National Plan for Scientific Research, Development and Technological Innovation 2008-2011.

References

[1] A. Araujo, E. Romero, J. Blesa and O. Nieto-Taladriz, “Cognitive Wireless Sensor Networks Framework for Green Communications Design”, II International Conference on Advances in Cognitive Radio. COCORA 2012, pp. 34-40.

[2] K. David, D. Dixit, and N. Jefferies, “2020 Vision”, IEEE Vehicular Technology Magazine, 2010, vol. 5, pp. 22 –29.

[3] Cisco Systems Inc, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015”. White Paper. Feb. 2011.

[4] S. Methley and C. Forster, “Wireless Sensor Networks Final Report”, Technical Report, Plextek Limited, 2008.

[5] I. Howitt and J. Gutierrez, “IEEE 802.15.4 Low Rate-Wireless Personal Area Network Coexistence Issues”. Proc of IEEE Wireless Communications and Networking Conference 2003 (WCNC 2003), vol. 3, pp.1481-1486, doi:10.1109/WCNC.2003.1200605.

[6] D. Cavalcanti, R. Schmitt and A. Soomro, “Achieving Energy Efficiency and QoS for Low-Rate Applications with 802.11e”, IEEE Wireless Communications and Networking Conference, 2007.WCNC 2007, pp. 2143 –2148, doi:10.1109/WCNC.2007.401.

[7] Y. Xue, H.S. Lee, M. Yang, P. Kumarawadu, H. Ghenniwa, and W. Shen, “Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks”. Canadian Conference on Electrical and Computer Engineering 2007. CCECE 2007. pp. 1372 –1375, doi:10.1109/CCECE.2007.345.

[8] P. Levis, N. Lee, M. Welsh and D. Culler, “TOSSIM: accurate and scalable simulation of entire TinyOS applications”, Proc of the 1st ACM International Conference on Embedded Networked Sensor Systems 2003; SenSys ’03. pp. 126–137, doi:10.1145/958491.958506.

[9] L. Girod, N. Ramanathan, J. Elson, T. Stathopoulos, M. Lukac and D. Estrin, “Emstar A software environment for

developing and deploying heterogeneous sensor-actuator networks". ACM Transactions on Sensor Networks. Vol 3 Issue 3, August 2007, doi:10.1145/1267060.1267061.

[10] G. Pongor, "OMNeT: Objective Modular Network Testbed", Proc of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems 1993; MASCOTS '93. pp. 323–326.

[11] X. Xian, W. Shi and H. Huang,. "Comparison of OMNET++ and other simulator for WSN simulation", 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA 2008. pp. 1439 –1443, doi:10.1109/ICIEA.2008.4582757.

[12] M. Imran, A. Said and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks", International Symposium in Information Technology 2010 (ITSim 2010), Vol. 2, pp. 897 –902, doi:10.1109/ITSIM.2010.5561571.

[13] P. Lee, and G. Wei, "NS2 Model for Cognitive Radio Networks Routing", International Symposium on Computer Network and Multimedia Technology 2009. CNMT 2009. pp. 1 –4, doi:10.1109/CNMT.2009.5374689.

[14] J. Zhong and .J. Li, "Cognitive Radio Cognitive Network Simulator", http://stuweb.ee.mtu.edu/~ljialian/, Accessed on 2012.

[15] N. Uchida, G. Sato, K. Takahata and Y. Shibata, "Optimal Route Selection Method with Satellite System for CognitiveWireless Network in Disaster Information Network", IEEE International Conference on Advanced Information Networking and Applications 2011 (AINA 2011), pp. 23 –29, doi:10.1109/AINA.2011.42.

[16] S.Y. Hung, Y.C. Cheng, E.K. Wu and G.H. Chen, "An Opportunistic Cognitive MAC Protocol for Coexistence with WLAN", IEEE International Conference on Communications, 2008. ICC '08. pp. 4059 – 4063, doi:10.1109/ICC.2008.762.

[17] X. Tian, T. Tian and S. Li, "A Network Simulation Platform for Hierarchical Spectrum Sharing Based Cognitive Radio Network", International Conference on Computational and Information Sciences 2010 (ICCIS 2010), pp. 613 – 617, doi:10.1109/ICCIS.2010.155.

[18] M. Zhan, P. Ren and M. Gong, "An Open Software Simulation Platform for Cognitive Radio", 6th International Conference on Wireless Communications Networking and Mobile Computing 2010 (WiCOM 2010), pp. 1 –4, doi:10.1109/WICOM.2010.5600841.

[19] S. Wang, H. Liu, L. Xie and W. Hu, "Cognitive radio simulation environment realization based on autonomic communication", IEEE 3rd International Conference on Communication Software and Networks 2011 (ICCSN 2011), pp. 402 – 407, doi:10.1109/ICCSN.2011.6013743.

[20] V. Handziski, A. Kopke, A. Willig and A. Wolisz, "TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks", 2nd International Workshop on multi-hop Ad Hoc Networks: from Theory to Reality 2006 (RealMAN'06), pp. 63-70, doi:10.1145/1132983.1132995.

[21] T.R. Newman, A. He, J. Gaeddert, B. Hilburn, T. Bose and J.H. Reed, "Virginia tech cognitive radio network testbed and open source cognitive radio framework", 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops 2009, TridentCom 2009, pp. 1-3, doi:10.1109/TRIDENTCOM.2009.4976217.

[22] 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", Proc of the IEEE Wireless Communications and Networking Conference (WCNC 2005), Vol 3, pp. 1664-1669, doi:10.1109/WCNC.2005.1424763.

[23] D. Pediaditakis, Y. Tselishchev and A. Boulis, "Performance and scalability evaluation of the Castalia wireless sensor network simulator", Proc of the 3rd International ICST Conference on Simulation Tools and Techniques 2010. SIMUTools '10, pp. 53:1–53:6, doi:10.4108/ICST.SIMUTOOLS2010.8727.

[24] J. Rabaey, A. Wolisz, A.O. Ercan, A. Araujo, F. Burghardt, S. Mustafa, A. Parsa, S. Pollin, I.H. Wan and P. Malagon, "Connectivity Brokerage - Enabling Seamless Cooperation in Wireless Networks", (A White Paper), October 2010. [Online]. Available: http://bwrc.eecs.berkeley.edu/php/pubs/pubs.php/1484.html <retrieved: February, 2012>

[25] A. Araujo, J. Blesa, E. Romero and O. Nieto, "Cooperative jam Technique to Increase Physical layer Security in CWSN", II International Conference on Advances in Cognitive Radio COCORA 2012, pp. 11–14.