UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LUCAS BONDAN

Kitsune: A Management System for Advanced Radio Networks based on Cognitive Functions

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Lisandro Zambenedetti Granville
Advisor

Prof. Dr. Cristiano Bonato Both
Coadvisor

Porto Alegre, January 2014

CIP – CATALOGING-IN-PUBLICATION

Bondan, Lucas

Kitsune: A Management System for Advanced Radio Networks based on Cognitive Functions / Lucas Bondan. – Porto Alegre: PPGC da UFRGS, 2014.

101 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Advisor: Lisandro Zambenedetti Granville; Coadvisor: Cristiano Bonato Both.

1. Wireless Communication. 2. Cognitive Radio Networks. 3. Cognitive Functions. 4. Network Management. I. Granville, Lisandro Zambenedetti. II. Both, Cristiano Bonato. III. Título.

“Whether you think that you can, or that you can’t, you are usually right”.
— HENRY FORD

ACKNOWLEDGMENTS

I would like to express my gratitude to my father Gentil Luiz Bondan and my mother Marli Pedroso Bondan, my life examples that helped me in all difficult times. I also would like to thank my brothers Ulysses Bondan and Moisés Bondan, that, together with my parents, always provided me all the emotional support that I needed during my whole life. I am grateful to my girlfriend, Camila Grando, who spent the last three years by my side, in the good and bad times. She is my companion and my friend.

I would like to express my gratitude to my advisor, Lisandro Zambenedetti Granville, by always encouraging me and guide me to be a better researcher. In addition, I would like to thank my co-advisor, Cristiano Bonato Both, for his good advices and time, always trying to polish my inner researcher and helping me with the most difficult problems. Also, I am thankful to my professors that participates on my masters, professors Juergen Rochol, Luciano Paschoal Gaspary, Marinho Pilla Barcellos, and Liane Margarida Rockenbach Tarouco.

I am grateful to all of my friends and partners at the Computer Network Group, mainly, Marcelo Antonio Marotta, Leonardo Roveda Faganello, Maicon Kist, Juliano Araújo Wickboldt, José Felipe Carbone, Ricardo Luis dos Santos, José Jair Cardoso Santanna, and Wanderson Paim de Jesus. All of them were always supporting me by giving their best effort to help me or by just being there and making me happy. I would like to thank also the rest of my friends, colleagues, professors, and staff of the Informatics Institute from UFRGS, by their good work and assistance.

Finally, I would like to express my gratitude to those people that influenced me directly or indirectly during the masters and also my life. Thank you all.

Kitsune*: Um Sistema de Gerenciamento para Redes de Rádio Avançadas Baseado nas Funções Cognitivas*

RESUMO

Considerando a atual subutilização do espectro de rádio frequências para comunicação sem fio, o rádio cognitivo é visto como um conceito chave para permitir uma melhoria da utilização deste recurso de comunicação. A implementação de dispositivos de rádio cognitivo deve basear-se nas quatro principais funções cognitivas: sensoriamento espectral, decisão espectral, compartilhamento espectral e mobilidade espectral. Através dessas funções, um dispositivo de rádio cognitivo é capaz de procurar canais livres para transmitir de forma oportunista em uma rede de rádios cognitivos. No entanto, as redes de rádios cognitivos devem ser gerenciadas, com o objetivo de garantir seu pleno funcionamento, melhorando o desempenho destes dispositivos. Este gerenciamento deve melhorar o conhecimento do administrador sobre o funcionamento da rede. Assim, a configuração, o monitoramento e a visualização das funções cognitivas são fundamentais para o processo de aprendizagem contínua do administrador de rede. Neste trabalho, propõe-se *Kitsune*, um sistema de gerenciamento com base em um modelo hierárquico que permite gerenciar as informações sobre as funções cognitivas em redes de rádios cognitivos. *Kitsune* é projetado para gerenciar todas as quatro funções cognitivas, permitindo que o administrador da rede possa configurar os dispositivos de rádio cognitivo, monitorar os resultados de cada função cognitiva e analisar importantes visualizações destes resultados. Além disso, um protótipo de *Kitsune* foi desenvolvido e avaliado por meio de um cenário experimental baseado na norma IEEE 802.22. O resultado obtido mostra que *Kitsune* fornece ao administrador um melhor conhecimento sobre a rede, melhorando a taxa de transferência média para cada canal.

Palavras-chave: Comunicação wireless, redes de rádios cognitivos, funções cognitivas, gerencia de redes.

Kitsune: A Management System for Advanced Radio Networks based on Cognitive Functions

ABSTRACT

Considering the current underutilization of radio frequency spectrum for wireless communication, the Cognitive Radio is seen as a key concept to enable the improvement of the radio frequency spectrum utilization. The implementation of cognitive radio devices must be based on the four main cognitive functions: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility. Through these functions, a cognitive radio device is able to search for vacant channels to opportunistically transmit in a cognitive radio network. However, cognitive radio networks should be managed, aiming to guaranty the proper operation of the cognitive radio devices, improving the performance of these devices. This management should improve the administrator knowledge about the cognitive radio network operation. Therefore, the configuration, monitoring and visualization of the cognitive functions are fundamental to the continuous knowledge building process of the network administrator. In this paper we propose *Kitsune*, a management system based on a hierarchical model allowing to manage summarized information about cognitive functions in radio networks. *Kitsune* is designed to manage all four cognitive functions, enabling the network administrator to configure the cognitive radio devices, monitor the results of each cognitive function, and make important visualizations of these results. Moreover, a *Kitsune* prototype was developed and evaluated through an experimental IEEE 802.22 scenario. The result obtained show that *Kitsune* allows the administrator to achieve a better knowledge about the network and improve the average throughput for each channel.

Keywords: Wireless communication, cognitive radio networks, cognitive functions, network management.

LIST OF ABBREVIATIONS AND ACRONYMS

API	*Application Programming Interface*
BS	*Base Station*
CCL	*Candidate Channels List*
CF	*Cognitive Functions*
CogMesh	*Cognitive Radio Mesh Network*
CPE	*Customer-Premises Equipment*
CR	*Cognitive Radio*
CRAHN	*Cognitive Radio Ad-Hoc Network*
DSA	*Dynamic Spectrum Access*
GPS	*Global Positioning System*
GCL	*Global Channels List*
GUI	*Graphic User Interface*
HAL	*Hardware Abstraction Layer*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
ISM	*Industrial, Scientific and Medical*
ISP	*Internet Service Provider*
JSON	*JavaScript Object Notation*
LAN	*Local Area Network*
MAC	*Medium Access Control*
MIB	*Management Information Base*
NOC	*Network Operation Control*
OCL	*Ordained Channels List*
PBNM	*Policy-Based Network Management*

PC	*Personal Computer*
PHP	*Hypertext Preprocessor*
QoS	*Quality of Service*
REST	*REpresentational State Transfer Protocol*
RF	*Radio Frequency*
ROA	*Resource Oriented Architecture*
RSSI	*Received Signal Strength Indicator*
SDR	*Software Defined Radio*
SMI	*Structure of Management Information*
SNMP	*Simple Network Management Protocol*
SOA	*Service Oriented Architecture*
TDD	*Time Division Duplex*
URI	*Uniform Resource Identifier*
USRP	*Universal Software Radio Peripheral*

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The use of the Radio Frequency (RF) spectrum bands is regulated by a spectrum management process driven by governments in most countries. This process is actually guided through licensing policies, where a given RF spectrum band is assigned to licensed spectrum holders that can explore the channels in this band exclusively for its licensed users. The actual licensing policies, combined with the increasing demand for RF spectrum bands for wireless users, caused a problem of spectrum scarcity (MASONTA; MZYECE; NTLATLAPA, 2013). Additionally, the users of the scarce spectrum present a dynamic usage behavior of the channels, once that these users may transmit in different periods of time, during different amounts of time. This behavior leads to an underutilization of RF spectrum in periods where licensed users are not transmitting in their assigned band.

Considering these problems, Cognitive Radio (CR) concept emerged, as a solution to improve the usage of the RF spectrum (MITOLA J.; MAGUIRE G.Q., 1999). CR devices are able to search for channels of the RF spectrum, that are not been used by licensed users. Then, the CR devices can use these channels opportunistically to perform transmission, since interference with licensed users does not occur (AKYILDIZ et al., 2006). Due to the dynamic usage of the RF spectrum by the licensed users, where vacant channels may become occupied in an unspecific time, CR devices should implement four mainly cognitive functions to operate properly (AKYILDIZ et al., 2006): (*i*) Spectrum Sensing enables CR devices to analyze a range of frequencies to detect the existence of vacant channels (MASONTA; MZYECE; NTLATLAPA, 2013); (*ii*) Spectrum Decision allows CRs to decide which one of the detected vacant channels will be assigned for transmissions (TRAGOS et al., 2013); (*iii*) Spectrum Sharing enables different CRs to share a vacant channel (BIAN et al., 2013); (*iv*) Spectrum Mobility allows CRs to seamless switch between different channels when a vacant channel becomes occupied (DE DOMENICO; CALVANESE STRINATI; DI BENEDETTO, 2012).

Cognitive functions have been gradually implemented in radio devices changing the management paradigm of current wireless networks to a CR network (AKYILDIZ et al., 2008). The configuration, monitoring and visualization of these functions represents a challenge in network management, because instead of managing typical network elements, *e.g.*, ports and routing tables, it is necessary to manage the cognitive functions in their entirety, *i.e.* the configuration parameters, such as the range of channels to be sensed, sensing duration, transmission duration, list of shareable channels, etc.; and results obtained after the execution of the cognitive functions in the CR network. The first proposal of a CR network was defined in the IEEE 802.22 Standard (CORDEIRO et al., 2005). In this standard, a CR network is composed of two types of devices: Base Stations (BS) and Customer-Premises Equipment (CPE) (IEEE, 2011), where both implement cognitive functions. The standard also provides a Management Information Base (MIB) for the information management of the spectrum sensing function and the transmissions performed by the CPEs and BSs, among other information. As proposed in IEEE 802.22 Standard, the usage of CR concept was deployed in traditional networks such as ad-hoc (CRAHN) (AKYILDIZ; LEE; CHOWDHURY, 2009) and mesh (CogMesh) (CHEN et al., 2007), improving the communication over the RF spectrum in these networks.

In the last decade, much has been discussed about CR networks (DARDAILLON et al., 2012) and, recently, a growing number of discussions have taken place about the management of these networks, since a management system should offer summarized information about the cognitive functions. Inside these discussion, researches were carried out about different ways to manage these networks, *e.g.*, using management protocols. (POTIER; QIAN, 2011; ZUBAIR et al., 2012), policies (SHERMAN et al., 2010; VANDERHORN et al., 2010), spectrum visualization (MANFRIN; ZANELLA; ZORZI, 2010; NAGANAWA et al., 2011), and machine learning algorithms (MANOJ; RAO; ZORZI, 2008; STAVROULAKI et al., 2012). However, these researches focused on RF spectrum, showing how the sensed spectrum is occupied and are the best ways to achieve spectrum usage efficiency. The researches fail to show how each cognitive function is being performed and also how these functions can influence the operation of CR network.

Considering the lack of CR network management solutions, in this dissertation we first investigate the problems involved in the management of CR networks. Based on the results of this investigation, we propose and develop a management system called *Kitsune*, the first solution designed to manage CR networks based on the four main cognitive functions. A prototype of *Kitsune* was developed, allowing the evaluation of the proposed solution. The proposed system supports a continuous knowledge building process to the network administrator, based on three stages: (*i*) configuration, (*ii*) monitoring, and (*iii*) visualization. Each of these stages operates according to an hierarchical management model, composed of three different entities: a Manager, responsible for configuring and requesting the information of the CR devices to compose visualizations; a Gateway that delivers the configuration from the Manager to CR devices and summarizes the information about the CR devices to the Manager; and an Agent, responsible for handling local information in the CR devices through an IEEE 802.22 MIB and communicating with the Gateway. The main advantage of the proposed system is to enable dynamic configuration and monitoring through an intuitive visualization interface. Moreover, complementary contributions were obtained during the development of this dissertation. These contributions were designed and developed to enable the management of the cognitive functions using *Kitsune*:

1. An extension of the IEEE 802.22 MIB that can be used to the management of different cognitive functions that are not mapped by the IEEE 802.22 Standard, such as the spectrum decision and spectrum sharing functions;

2. A channel list management system, used to prove the management of the spectrum decision, spectrum sharing and spectrum mobility functions. This system is able to define the best channel to be used by the CR devices. Such function is directly managed by *Kitsune*, showing the proper operation of the proposed management system;

3. A set of visualizations of the cognitive functions that allows the network administrator to achieve a new level of knowledge about the CR devices operation in the network. These visualization considers the results of the cognitive functions performed by the CR devices.

A prototype of *Kitsune* based on the Resource Oriented Architecture (ROA) Web Services architecture was implemented using the REpresentational State Transfer Protocol (REST) to explore and to evaluate the proper operation of the proposed system in the management of cognitive functions. *Kitsune* prototype was evaluated in an experimental scenario based on the IEEE 802.22 Standard (IEEE, 2011) deployed using the GNU Radio toolkit[1] and Ettus USRP N210 devices[2]. An energy detection algorithm was used to evaluate the management of the spectrum sensing function. Moreover, we developed a channel list management system, that enables the analysis of the spectrum decision, spectrum sharing and spectrum mobility functions. Gathered results were analyzed, demonstrating that the proposed system enables the administrator to achieve a better knowledge about the network. It enables, among other enhancements, the improvement of the throughput obtained in the transmission performed by the CPEs, directly impacting in the CR network communication, *i.e.* the effectiveness of the performed transmissions.

The rest of this dissertation is organized as follows. In Chapter 2, we present a background on CR networks and the management of such networks, discussing the related work in this research area. In Chapter 3, the *Kitsune* management system is explained in details. In Chapter 4, *Kitsune* implementation is presented for proof of concept, through the management of two cognitive functions. In Chapter 5, the evaluation methodology is explained and results obtained are discussed. Finally, conclusions and future work are provided in Chapter 6.

---

[1] http://gnuradio.org/

[2] http://www.ettus.com

2 BACKGROUND AND RELATED WORK

In this chapter, a discussion about the aspects involved in the concept of CR devices and networks is presented, analyzing the main research efforts found in the literature in the last years. In Section 2.1, we explain the fundamentals of CR devices and the networks composed of these devices, discussing the main challenges in the development of such networks. Moreover, in Section 2.2, the aspects involved in the management of CR networks are discussed in details, emphasizing the challenges in the development of management solutions for CR networks.

2.1 Cognitive Radio Devices and Networks

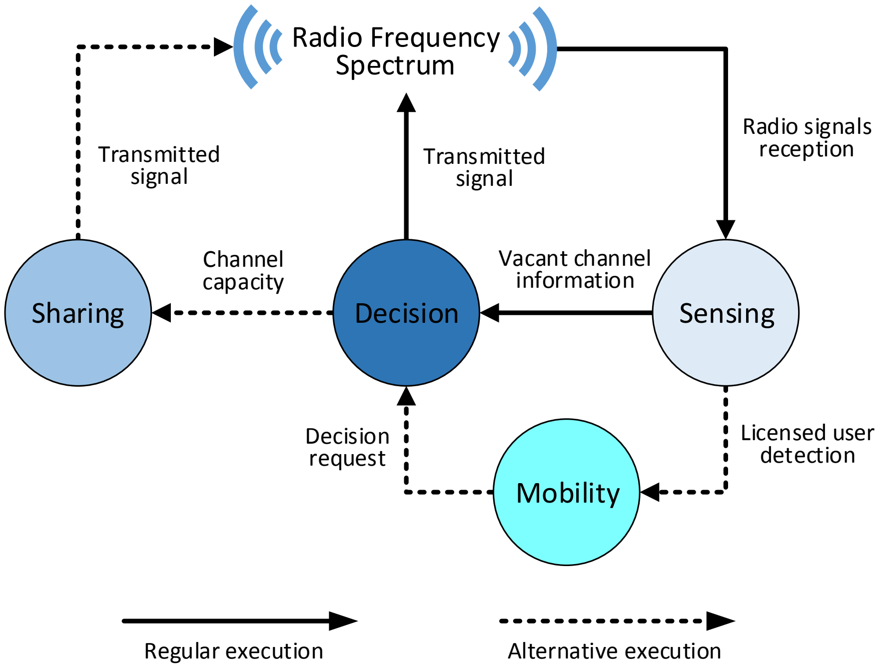
CR devices are foreseen as the future of radio devices, improving the utilization of the RF spectrum (MASONTA; MZYECE; NTLATLAPA, 2013). The most important characteristic of the CR is the ability to search for vacant channels in the RF spectrum and opportunistically perform transmissions over these channels. To enable these opportunistic transmissions, nearly all aspects of its operation should be dynamically configured (MITOLA J.; MAGUIRE G.Q., 1999). The literature divides the functionality of these devices in four main functions: spectrum sensing, spectrum decision, spectrum sharing and spectrum mobility (AKYILDIZ et al., 2006). The spectrum sensing allows the CR device to identify vacant channels (YUCEK; ARSLAN, 2009). Spectrum decision selects the best available channel to transmit, according to some criteria, such as historical availability, channel quality, among others (MASONTA; MZYECE; NTLATLAPA, 2013). Spectrum sharing comprises the coordinated access of two or more CR devices to a vacant channel (PEHA, 2009). Finally, spectrum mobility enables a CR device to vacate the channel when another transmission is detected (CHRISTIAN et al., 2012).

Among the four cognitive functions of CR devices, the spectrum sensing provides the capability to analyze the RF spectrum and evaluate whether or not a channel is vacant. In addition, all the other functions can use the spectrum sensing results as input to perform their operations (YUCEK; ARSLAN, 2009). Directly associated with the spectrum sensing we found the spectrum decision. In this cognitive function, the CR devices processes the results obtained through the spectrum sensing and selects a channel for communication. This selection may require many parameters, depending on the used algorithm, such as reinforcement learning, fuzzy logic, graph theory, evolutionary algorithms, linear programming, among others (TRAGOS et al., 2013). After the selection of a channel to transmit, the CR devices must reconfigure its transmitting and/or receiving parameters to match the result of the channel selection. The coordinated access to the channels is provided by the spectrum sharing. Through this cognitive function, a channel may be shared by different CR devices, adjusting the bandwidth used in each channel according to the

CR devices requirements (STOTAS; NALLANATHAN, 2011). When the presence of another user is detected in the transmitting channel, the CR device should start the process of stopping its transmission to avoid interference and to select another channel to continue the transmission. This process is called spectrum handoff and is possible through the spectrum mobility (BICEN et al., 2013).

The relation among the cognitive functions is shown in Figure 2.1. The solid arrows represent the regular execution of the cognitive functions. In this execution, the sensing function analyses the radio frequency spectrum and detects opportunities for transmission, *i.e.*, vacant channels. Afterwards, the decision function considers this information to select a channel for transmission. Dashed arrows represent an alternative execution that may be performed when: (*i*) an interference with the licensed user is detected and the mobility function request to change the operation channel, or (*ii*) when a channel is defined to be shared by many CR devices. It is important to emphasize that all functions are not necessarily executed in the same CR device. For example, a device with low computational resources may perform the spectrum sensing, providing information about vacant channels to a second device that will be responsible for the spectrum decision. This second device could use the result of the decision to transmit or to coordinate the transmission of other devices. Moreover, the spectrum sharing function may never be performed, for example, in situations where only one CR device needs to transmit. Similarly, if the presence of the licensed users is not detected in the current transmitting channel, the CR devices may not perform the spectrum mobility, keeping its transmission in the current channel.

Figure 2.1: Cognitive functions



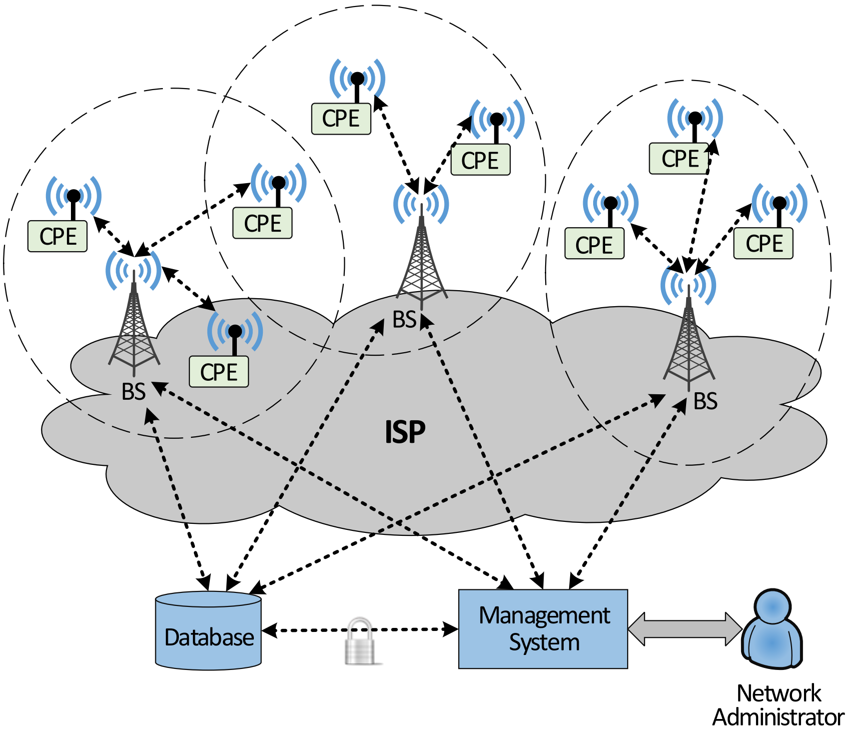
An example of CR network where the cognitive functions are performed in different CR devices is defined in the IEEE 802.22 Standard (IEEE, 2011). This Standard defines a centralized CR network composed by two types of devices: Base Station (BS) and Customer-Premises Equipments (CPEs). The BS is responsible for coordinating the Internet connection of the CPEs, while these devices should enable the Internet access to the users. Thus, the CPEs are responsible for performing the spectrum sensing, retrieving the results to the BS, which invoke the spectrum decision to select the best channel (according to a decision criteria) and inform the CPEs, through a control channel, to access the selected channel. Is important to highlight that in the IEEE 802.22 Standard, the CPE operation is fixed, *i.e.*, no roaming is performed between CPEs and BSs. As result of the spectrum decision performed by the BS, the channel can be classified according to three different types: operating channel, backup channel and candidate channels. The BS selects a channel to be the operating channel, that can be used for transmission. It is important to highlight that the operating channel must not necessarily be used to transmit, but the operating channel should be defined when CPE wants to transmit. If the operating channel becomes occupied, the BS will automatically select the backup channel to be used as the new operating channel. When the current backup channel is selected as new operating channel, any candidate channel can become the new backup channel. The functions performed by BS and CPE should be managed to guarantee the proper operation of the CR network. However, because a CR network may comprise a large number of BSs and CPEs, its management may become complex, given that each BS and CPE can be dynamically configured, monitored and visualized.

Figure 2.2 shows the CR network proposed by the IEEE 802.22 Standard, as well as the main elements that compose the network. The BSs are directly connected to the Internet Service Provider (ISP) and are responsible to provide Internet access to CPEs. The management system is located out of the CR networks and access the information of these networks through a connection, such as an Internet or a Local Area Network (LAN). Moreover, the IEEE 802.22 Standard proposes the use of a database to maintain the information collected from the CR networks. The access to the database must be secure, avoiding information leakage. This access can be performed by the management system to obtain the information regarding the CR networks operation. In CR networks as defined by the IEEE 802.22 Standard, all management messages exchanged between BS and CPEs are usually transmitted over a control channel (IEEE, 2011), avoiding the overhead of control and management messages in the communication. This control channel is selected before the beginning of the network operation and presents the best propagation conditions. Usually, this channel comprises a 6 MHz bandwidth, with a transfer rate of $18Mbps$ and can be used only for control and management data. In the next section, the management issues of CR networks are explained in details.

2.2 Management of Cognitive Radios

There is not a widely accepted taxonomy that defines and characterizes what are the network management model types existing, both for traditional and CR networks. Some authors separate the network management area in centralized or hierarchical, poorly , highly distributed and cooperative management, considering both the number of managed elements and scalability of the system (SCHONWALDER; QUITTEK; KAPPLER, 2000). Other authors classify the network management according to organizational models (MARTIN-FLATIN; ZNATY; HUBAUX, 1999) (LEINWAND; CONROY, 1996). To simplify the management of CR networks, in this work we separate the network management in three entities to be used according to a hierarchical model: (*i*) the Management Station, responsible for the management tasks, such as monitoring each CR device in the network or producing summarized reports about the overall network; (*ii*) the BS, where a specific management component is capable of executing ordinary actions, such as collecting information about the CR devices in the network; and (*iii*) CPEs, with components responsible for responding the requests performed by the BS component and applying new configurations to these devices. In the literature, the BS usually operates the IEEE 802.22 MIB (IEEE, 2011).

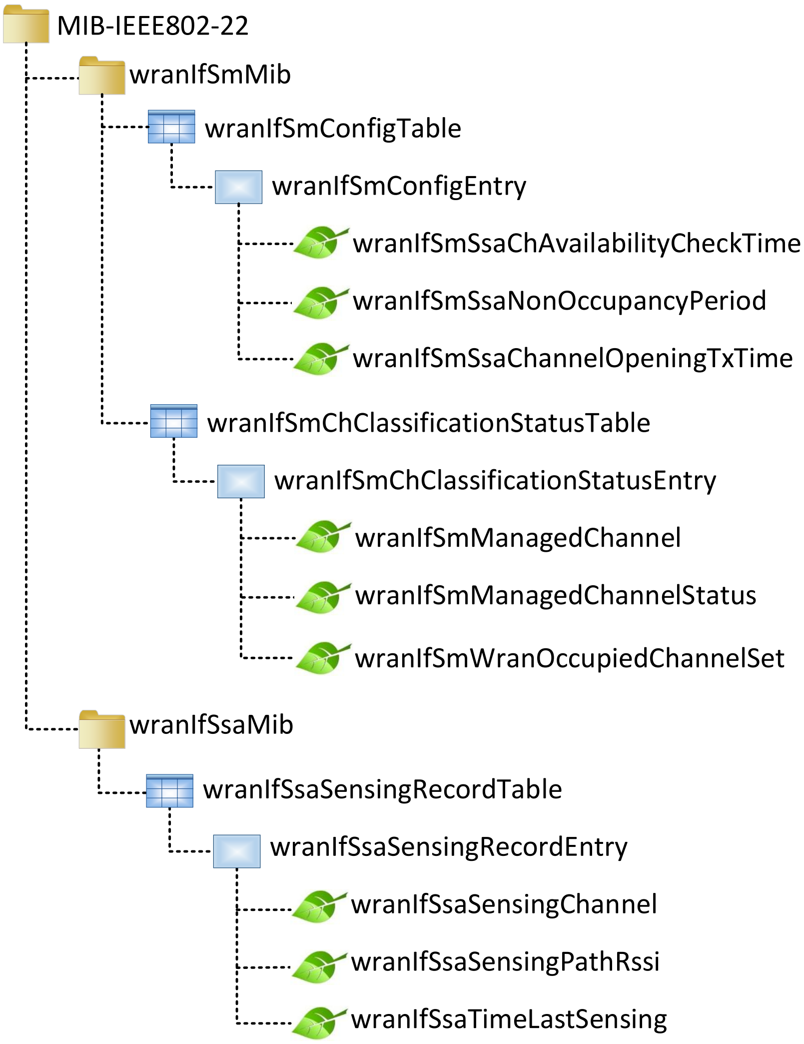
Figure 2.2: Overview of a CR network as defined in the IEEE 802.22 Standard



A MIB is a conceptual database that organizes the information about current settings and results of the cognitive functions. The IEEE 802.22 Standard specifies a MIB component (IEEE, 2011) that provides relevant information for CR networks management systems. This MIB is usually handled by an Agent, responsible for responding the management requests with values present in the MIB component, which is organized in seven groups. The importance attached to the cognitive functions is perceptible due to the *wranIfSmMib* and the *wranIfSsaMib* groups. The first group presents objects associated with the RF spectrum management, while the second group has objects related to the spectrum sensing function management. These groups are presented in Figure 2.3.

Considering the presented IEEE 802.22 MIB groups, we can describe some important objects for a CR network management system. These objects may provide useful information regarding the cognitive functions in each CR device. In the *wranIfSmMib* group, we highlight the channel availability check time object (*wranIfSmSsaChAvailabilityCheckTime*), which is the time during which a channel shall be sensed for the presence of licensed users, *i.e.*, the sensing window. The non occupancy period object (*wranIfSmSsaNonOccupancyPeriod*) indicates the required period during which a CR device transmissions shall not occur in a given channel, because it was defined as occupied. The channel transmission time (*wranIfSmSsaChannelOpeningTxTime*) is also fundamental information regarding to the maximum time taken by CR device to transmit data over the selected channel. The managed channel identification (*wranIfSmManagedChannel*) and its status (*wranIfSmManagedChannelStatus*) represent information of the managed channels, *i.e.*, the channels that can be used in the CR network. Another important object is the set of occupied channels (*wranIfSmWranOccupiedChannelSet*), which represents all the channels evaluated as occupied during the spectrum sensing. In the second group, we highlight the object associated with the sensed channel number (*wranIfSsaSensingChannel*). The Received Signal Strength Indication (RSSI) during the sensing (*wranIfSsaSensingPathRssi*) and the timestamp of most recent sensing (*wranIfSsaTimeLastSensing*) are also fundamental information for CR network management systems. Despite the IEEE 802.22 MIB be able to maintain relevant information about the operation of IEEE 802.22 networks, this MIB is not switchable to be used by other infra-structured CR networks that are not according to the IEEE 802.22 network operation, as for example, in infra-structured networks where the CPEs are able to take their own decisions. However, the IEEE 802.22 MIB has significant objects that can be used to manage other networks, by extending this MIB to accomplish the functions not mapped, *i.e.*, creating new classes and objects to encompasses more functions performed by the CR devices.

Figure 2.3: IEEE 802.22 MIB groups structure



The IEEE 802.22 Standard indicates the use of the Simple Network Management Protocol (SNMP) to exchange management messages between BS and CPEs. SNMP is a simple solution that requires little code to implement, which enables vendors to easily build SNMP agents for their products. In addition, SNMP is often the foundation of the network management architecture. SNMP defines how management information is exchanged between network management applications and management agents. Thus, considering an IEEE 802.22 network with a maximum of 512 Agents and 2 seconds as the smallest time interval between two consecutive requests performed by the Gateway, the SNMP is particularly interesting to be used for communication between BS and CPE, since that the SNMP traffic will correspond to less than 15% of the control channel capacity. Despite the time interval of cognitive functions executions may be less than 2 seconds, the organization of the cognitive functions results in the MIB avoids information loss, even that the Gateway request these information with grater time intervals.

Despite all advantages in using the MIB combined with SNMP, this protocol is not suitable to realize communications over the Internet, mainly due to the absence of a control channel and also because firewalls may block SNMP messages. One way to avoid SNMP over the Internet is using management approaches based on Web Services (WS) (MOURA et al., 2007). These approaches may be implemented according to two main architectures: Service Oriented Architecture (SOA) (BOX et al., 2000) and Resource Oriented Architecture (ROA) (FIELDING, 2000). Both architectures may use Hypertext Transfer Protocol (HTTP) to transmit data, avoiding firewalls. Moreover, SOA and ROA take advantage of HTTP Secure (HTTPS) protocol to easily secure the messages exchanged. However, Pautasso *et al.* (PAUTASSO; ZIMMERMANN; LEYMANN, 2010) proved that ROA is better than SOA to achieve higher performance when implementing a WS based solution.

ROA is a loosely coupled approach to the client-server model that uses a Uniform Resource Identifier (URI) to directly access devices resources. In general, this approach follows the Representational State Transfer (REST) architectural style (FIELDING, 2000). This style defines HTTP as the only application protocol and standardizes the access interface through methods of this protocol (*i.e.*, GET, PUT, POST, and DELETE). Each REST message represents a state of the accessed resource, *i.e.*, the current collection of meaningful information, *e.g.* sensed channel list, sensing window, and sensing evaluation. Other features of REST includes the support to cache messages and resources inheritance. Despite all advantages, ROA is not used, at the best of our knowledge, to implement a sophisticated management application to CR networks. In the next section, we investigate and explain the related work about CR networks management.

2.3 Related Work

Much has been discussed about CR in the last decade and some research efforts about CR management emerged in the last years. We can notice that the academic community is mainly concerned with the improvement of the spectrum utilization, through the design of new solutions to the spectrum management. However, it is important to highlight the difference between the spectrum management and the CR network management. The spectrum management is related to the coordinated access to the RF spectrum through the cognitive functions of CR devices. On the other hand, the CR network management comprises the configuration, monitoring and visualization of the cognitive functions performed by the CR devices, aiming to maintain and improve the CR network operation. Taking into account this difference, research on CR management mainly addressed the utilization of different approaches in both the spectrum and network management.

We organized the main research in four main classes: (*i*) management protocols, designed to exchange information between management systems and managed devices; (*ii*) policies, applied to CR networks aiming to facilitate the network management; (*iii*) spectrum visualization, whose concern is to map the utilization of the RF spectrum and expose this information to the network administrator; and (*iv*) machine learning, aiming to improve both the spectrum utilization and the management decisions. These four classes of research are important to map the aspects of the CR network that should be managed by a management system. Moreover, these research efforts allow to understand how both the spectrum and network management directly influence in the CR network operation. Each class of research is discussed in details in the next subsections.

2.3.1 Management Protocols

Management protocols should retrieve, deliver, and share information about the network operation. One of the most traditional management protocols is the SNMP, which became the de facto standard for network management. However, with the growth of advanced network infrastructures, the utilization of SNMP alone is not sufficient to cover the management of these networks. Thus, new solutions to the network management are emerging in the last years, by incrementing the SNMP utilization or proposing new protocols based on traditional management models.

Among these solutions, we can highlight the work of Qin and Cui (QIN; CUI, 2009), in which a coordination protocol to manage RF spectrum assignment is proposed, aiming to enable the configuration management of CR devices. The authors designed a specific cooperative spectrum etiquette protocol called ASCP (Adaptive Spectrum Coordination Protocol). This protocol provides the distribution of control messages required to optimize spectrum access coordination by the radio devices that compose the network. In another work, Potier and Qian (POTIER; QIAN, 2011) addressed the problem of management for CR ad-hoc networks and presented how the management of these networks is different than the traditional management for wired and wireless networks. Moreover, the authors proposed a cognitive network management protocol for CR ad-hoc networks, mapping their solution to the Fault, Configuration, Accounting, Performance, Security (FCAPS) model. Zubair *et al.* (ZUBAIR et al., 2012) proposed a protocol to manage the dynamic selection of a control channel, using a learning algorithm to choose the best channel in terms of availability. The Light Distributed Geographical (LDG) protocol proposed by the authors uses learning algorithms to choose the channel that presents high availability time, selecting the best one to exchange control messages over the CR devices that compose the network.

2.3.2 Policies

In the network management area, a policy is defined as a set of rules to network administrator, manage and control the access to the network resources. The objective of the Policy-Based Network Management (PBNM) (SLOMAN, 1994) is to control the network behavior according to predefined policies. There are several solutions based on this model, but the most accepted definition is the PBNM architecture proposed by the Internet Engineering Task Force (IETF) (MOORE, 2003). Usually, policies are written by stakeholders, as governmental spectrum regulators and network administrators, and applied to the network, aiming to ensure that restrictions for the network devices will be respected, such as maximum bandwidth per device or maximum transmission power.

Based on policy management models, the utilization of policies in CR networks is investigated in many works. The research of Wang *et al.* (WANG et al., 2008) provides an overview of the essential functionalities of the spectrum sensing and discuss its impact on making business level policies for CR networks management. The authors argue that using a business level abstraction is a key element that simplifies the writing of policies, by approximating its description to a real-word language. Sherman *et al.* (SHERMAN et al., 2010) proposed a generic framework for policies enforcement of military Dynamic Spectrum Access (DSA) systems. The authors suggest the utilization of policy writing tools to allow the creation and validation of DSA policies to be applied by the framework in the network management. Morover, the author developed a tiered policy distribution architecture based on the proposed framework. In other work, Vanderhorn *et al.* (VANDERHORN et al., 2010) developed a framework to the dynamic adaptation of policies based in two cases: the analysis of channels capacity and interference minimization by adaptive frequency hopping strategies. The framework proposed by the authors can be extended to accommodate different policy engines and enforcement mechanisms, as well as the need for distributed policy learning and localized adaptation.

2.3.3 Spectrum Visualization

An important step to improve the RF spectrum utilization is to collect detailed data about this resource. Currently, governmental regulators conducts surveys of RF spectrum utilization using questionnaires. In a survey, a governmental regulator requests each licensed spectrum holder to answer questions including the type of wireless station being operated, the number of wireless stations, their availability and operating hours.

However, such surveys are conducted between long periods of time, but more frequent updates are necessary. There are also some additional problems. First, only registered wireless systems are surveyed, *i.e.*, wireless systems that can operate without a license or registration are not surveyed because the government cannot send the survey to be answered. In the case where a wireless system operates under a comprehensive license, which allows multiple terminals to operate with a single license, the RF spectrum utilization of each terminal cannot be surveyed. Additionally, the actual RF spectrum occupancy cannot be determined because only the operation hours is surveyed, *i.e.*, the period of time when licensed spectrum holders are operating. For example, even in the case of sparse transmissions with a low RF spectrum occupancy, the questionnaire will indicate total utilization of the RF spectrum band as long as a wireless device is turned on. Waiting time is also classified as RF spectrum utilization. Moreover, there is a possibility of false answers in the questionnaires because the survey depends on self-enumeration. An operator who is afraid of blame for inefficient RF spectrum use might report operation hours that are longer than the actual utilization.

Aiming to improve the RF spectrum utilization knowledge and its management, the research on spectrum visualization presents solutions to map the RF spectrum utilization graphically. Manfrin, Zanella, and Zorzi (MANFRIN; ZANELLA; ZORZI, 2010) proposed CRABSS[1], an open platform to monitor unlicensed spectrum bands. The proposed system exports information about the spectrum sensing to management tools, simplifying the development of spectrum utilization maps. Moreover, the proposed platform enables the data exchange between radio stations, while collecting link related performance measurements. In their work, Naganawa *et al.* (NAGANAWA et al., 2011) developed a system to perform real time spectrum analysis using heterogeneous devices. The authors defined the interface between software components based on an abstraction of the spectrum utilization into binary information. The proposed system was designed to expand the measurement coverage, by performing distributed spectrum sensing. We also highlight two

---

[1] ARAGORN Project – http://www.ict-aragorn.eu/

important tools for spectrum visualization: Google Spectrum Database[2], a channel occupation visualization tool based on the USA Federal Communication Commission (FCC) spectrum allocation databases; and Microsoft WhiteFiService[3], an Application Programming Interface (API) for RF spectrum measurements.

2.3.4 Machine Learning

Machine learning algorithms are a branch of artificial intelligence in which a computer generates rules underlying or based on raw data that has been fed into it (MITCHELL, 1997). The utilization of machine learning algorithms has been explored to provide to CR devices the cognitive feature needed to learn about the radio environment and adapt its operation (CLANCY et al., 2007). For example, machine learning algorithms can be designed to perform the spectrum decision function, by creating a historical knowledge obtained over the time to improve the selection of channel to be used for transmissions (FAGANELLO et al., 2013). With the rise of more sophisticated networks, such as cognitive networks and CR networks, research efforts about the utilization of machine learning algorithms in the network management are emerging. In some works, concepts of machine learning algorithms are used for the network management, analyzing the taken decisions to improve the following ones.

In this class of research, Thomas *et al.* (THOMAS et al., 2006) proposed a framework to the implementation of learning mechanisms by analyzing the relationship of CR networks with other technologies. The authors highlight the need of integrated architectures to network management, due to the autonomy that different cognitive networks presents (*e.g.* CR networks). Manoj, Rao, and Zorzi, (MANOJ; RAO; ZORZI, 2008) proposed CogNet[4], a system which gathers, processes, analyses, and stores information available through a variety of devices and protocols to build a cooperative repository. This repository holds spatiotemporal experience information about the network operation. In the proposed system, agents are responsible for gather the information from CR devices and forward to a shared information base, and also control the configuration of the respective CR device. Finally, Stavroulaki *et al.* (STAVROULAKI et al., 2012) presented an overview about machine learning mechanisms to perform automatic decisions about configuration in CR networks. The authors also presented basic learning functionalities for the identification and processing of information that can lead to exploitable knowledge in CR networks. The information collected from the network compose a knowledge database, used as baseline to take new configuration decisions.

The research efforts described are specific solutions to CR management. We can associate each class of research with the work present in this dissertation. Management protocols are used to exchange information among the management system and the managed devices. Policies can be used to configure the managed devices. Spectrum visualization are used to show the network environment to the network administrator. Finally, machine learning mechanisms are used to implement cognitive functions of the CR devices. In a CR network, CR devices should operate automatically, but we argue that this operation can be improved by combining the analysis of cognitive functions results and the administrator intervention. An administrator may configure the cognitive functions parameters such as the range of channels to be sensed, sensing window, transmission window, list of shareable channels, etc. However, to the best of our knowledge, no management system

---

[2] http://www.google.org/spectrum/whitespace/channel/

[3] http://whitespaces.msresearch.us/

[4] ARAGORN Proejct – http://www.ict-aragorn.eu/

for CR networks based on the configuration, monitoring and visualization of the cognitive functions has been proposed so far. To minimize this lack, we propose a management system called *Kitsune*, which allows administrators to acquire a better knowledge about a managed CR network. In the next chapter, we present *Kitsune* system, explaining the entities, components, and modules that compose its architecture.

3 PROPOSED SOLUTION

Although many works about spectrum management emerged in the last years, no management approaches for CR networks based on the cognitive functions have been proposed. Considering this opportunity, we propose a new management system for CR networks called *Kitsune*. In this chapter, we detail each operation performed by the proposed system, describing each functional block and entities that compose *Kitsune*. Moreover, in this chapter we present *ChiMaS*, a channel list management system proposed to operate under the management of *Kitsune*, describing its operation in details.

3.1 *Kitsune* Management System

CR devices must be able to analyze the RF spectrum looking for channels that are not being used by the licensed users and opportunistically transmit in these channels. To allow this opportunistic operation, CR devices are designed through the development of four main cognitive functions. Considering the network management perspective, these functions differ the CR devices management from the traditional radio devices management in some aspects. First, the operation of CR devices is dynamic, because its operation depends on the behavior of the RF spectrum users. The cognitive functions are directly associated to this behavior, once that the cognitive functions results vary along with the users behavior. Considering this fact, the second difference is that the management of CR devices should be performed considering the dynamicity of the cognitive functions, as opposed to traditional management over static parameters.

CR devices should operate autonomically, *i.e.*, once defined the values for the cognitive functions parameters, the CR network should operate without interventions. However, in a CR network, the operation of the CR devices may be improved analyzing the results of the cognitive functions in combination with administrator intervention. For example, an administrator may analyze the RF spectrum utilization and may reconfigure the sensing channel list, which must be in concordance with governmental regulations. This configuration must be defined by a human, *e.g.*, the network administrator. Another example is regarding the spectrum decision function, where the administrator can observe situations in which vacant channels are not selected for transmission, overloading the remaining channels. This situation may indicate that the decision algorithm is not properly configured. We classify the administrator intervention in the CR network operation as a continuous knowledge building process, which improves the administrator knowledge about the CR network operation.

Kitsune was not designed only to manage the configuration parameters of the CR devices, but also to provide the continuous knowledge building process to the network administrator. To make it possible, *Kitsune* aims to manage a CR network based on configuration, monitoring and visualization of the cognitive functions, following a hierarchical management model. We chose an hierarchical model mainly because the managed elements of centralized CR networks, as defined in the IEEE 802.22 Standard, usually have an hierarchical distribution, *i.e.*, devices with high computational resources are responsible to coordinate the devices with low computational capacity. For example, the CPEs are computationally less powerful than the BS. This scenario complicates the deployment of more advanced management tools in the CPEs. Thus, the hierarchical management model is more appropriate for centralized networks, since that more complex components of the management system can be deployed in the entities with higher computational capacity. Moreover, using this model, it is possible to isolate the management entities from the network devices, easing the maintenance and the replacement of the management entities when needed, without interfering in the network operation.

Based on the hierarchical management model, we designed *Kitsune* to be composed of two functional blocks: the first is the Network Operation Center (NOC) and the second is the CR network. An important difference between the functional blocs is the continuous process performed in each one. The network administrator continuous knowledge building process occurs in the NOC and presents a higher action time than the radio operation process, that is performed in the CR Network. It means that the loop that involves the network administrator intervention is slower than the loop of operations performed by the CR devices. These two functional blocks exchange information through a backhaul, which may be a private infrastructure, as a cellular network, or a shared infrastructure, as the Internet. This separation allows the management entities to operate out of the network devices, avoiding interference with each other. Figure 3.1 illustrates in details *Kitsune* system using a top-down approach.

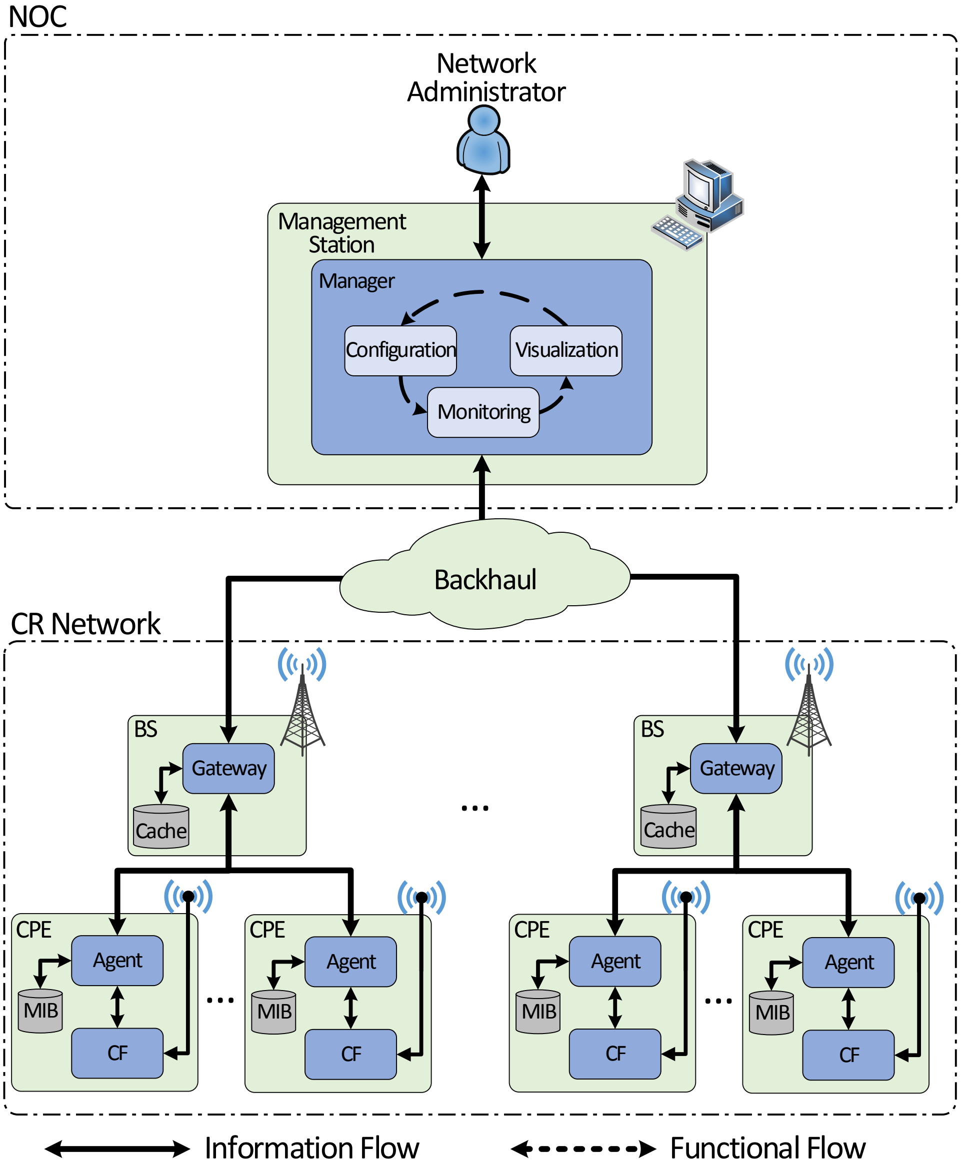
Each functional block is composed of different entities that support *Kitsune* operation. In the NOC, a Management Station executes a Manager component, composed of the Configuration, Monitoring and Visualization modules. Moreover, in CR Network block are the BS and CPE entities. The first is composed of a Gateway, responsible to receive and forward management messages from both Manager and CPEs. The CPE entity is composed of a MIB component, responsible to organize the managed information; an Agent, designed to directly manipulate the MIB; and a Cognitive Functions (CF) module, which executes the cognitive functions of the CPE. In the next subsections, the entities of *Kitsune* are described in details, exploring the components and modules present in each entity in both NOC and CR Network blocks. First, the Management Station component and its modules are described in details. Next, we describe the BS component present in the CR network. Finally, the CPE components are described. Additionally, the importance of the network administrator intervention in each stage of *Kitsune* operation is highlighted in all subsections.

3.1.1 Management Station Component

NOC is composed of a physical entity called Management Station, as can be seen in Figure 3.1. We designed the NOC aside from the CR Network to allow a management station to manage multiple CR networks. Moreover, due to placing the Management Station in the NOC, if one CR network breaks down, the management of the remaining CR networks must continue. The Management Station must (*i*) receive the configuration from the network administrator, (*ii*) forward the configuration to the CR devices that compose the CR network, (*iii*) collect the cognitive functions results from the CR devices, and (*iv*) provide visualizations of the cognitive functions results to the network administrator.

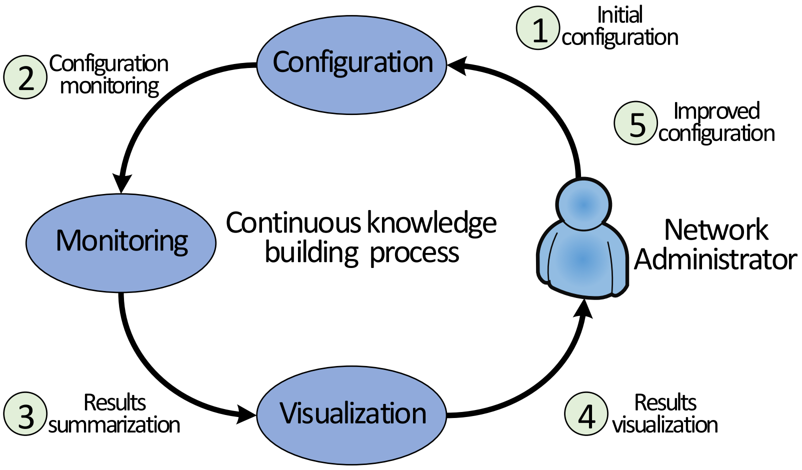
The Management Station comprises a Manager component, the top element of the *Kitsune* system. Using the Manager component, a network administrator can manage the CR Network operation through three different points of view. In the first one, he can select which CR network is desired to be managed. In the second one, the network administrator can select BSs to configure and/or analyze. Finally, the network administrator can select specific CPEs from a CR network to configure and/or to analyze the results of the cognitive functions. We designed the Manager component in this way to enable the management in different points of view, easing mainly the configuration of the CR devices and the analysis of the cognitive functions results by the network administrator.

Figure 3.1: *Kitsune* management system



The Manager component is composed of three modules: Configuration, Monitoring, and Visualization. Through these three modules, a network administrator that uses *Kitsune* can improve the configuration of the cognitive functions of each CR device by monitoring and analyzing their results. Therefore, the functional flow between the Manager component and the network administrator represents the continuous knowledge building process. *Kitsune* allows the continuous knowledge building process of the network administrator through three stages, each one performed in a specific module present in the Manager component. In Figure 3.2, we present the continuous knowledge building process by illustrating the steps involved in each cycle of this process.

Figure 3.2: Continuous knowledge building process



The first step of the continuous knowledge building process is the definition of an initial configuration (1) for the cognitive functions by the network administrator. This configuration usually does not consider any prior knowledge about the network operation, using the default configuration of the radio devices. Using *Kitsune*, the network administrator can define the configuration parameters, using the Configuration module. In this sense, the network administrator chooses which parameters of the cognitive functions can be managed, *e.g.*, sensed channel list, sensing window, and transmission time. The Configuration module was designed to allow the network administrator to configure the parameters and its respective values quickly and intuitively. Moreover, *Kitsune* provides three different levels to configure and analyze the results of the network: CR network, BS or CPE. In the first one, parameters are applied for all the CR network. In the second one, only BS is configured. Finally, in the last one, specific CPEs are configured or analyzed individually. Once selected the level of configuration, the network administrator can quickly apply the configuration for the desired parameters.

In the next step, the configuration of the network devices should be monitored (2), by monitoring the results of each cognitive function performed in the network. The gathered results should be summarized (3). The Monitoring module will monitor and summarize the cognitive functions results in each device of the CR network. Using the Monitoring module, the results of the cognitive functions can be obtained from the CR devices in the network. It is an important operation for the management system, since a mistaken monitoring can lead to erroneous summarization of the results and, consequently, mistaken conclusions about the CR network operation. For example, if the results of the spectrum sensing function cannot be collected in a specific interval, there is no information about the channel occupancy at the time. Thus, the network administrator will not be able to analyze the channel occupancy status.

The results summarization is useful to make the visualization (4), provided by the Visualization module, which can show a composition of the historical information of the cognitive functions results of each CR device. These visualizations are specific for each cognitive function, because each one presents the results in different formats. For example, while the spectrum sensing function can provide the occupancy status of the channels, the spectrum decision function may return as result the index of the selected channels. Thus, since the knowledge building process exhibits a cyclic behavior, the visualization provided by the Visualization module may be used as a feedback by a network administrator to modify the configurations of the CR network. These visualizations will be analyzed by a network administrator. Through the results visualization, the network administrator may achieve a new level of knowledge about the network operation that is difficult to be achieved without the analysis of the cognitive functions results.

Finally, after analyzing the visualizations, a network administrator may take accurate decisions about the previous configuration, *e.g.* defining the network operation as good or poor and modifying the configuration if he judge necessary (5). The five steps presented represents a cycle of the continuous knowledge building process of the network administrator. Performing the last step (5), the network administrator does not only finish a cycle, but also starts a new one, causing the configuration, monitoring and visualization of the cognitive functions in the continuous knowledge building process to improve the CR network operation. Moreover, this cycle may be performed, at any time, and as often as the administrator deems necessary without interfering with the current network operations.

3.1.2 BS Component

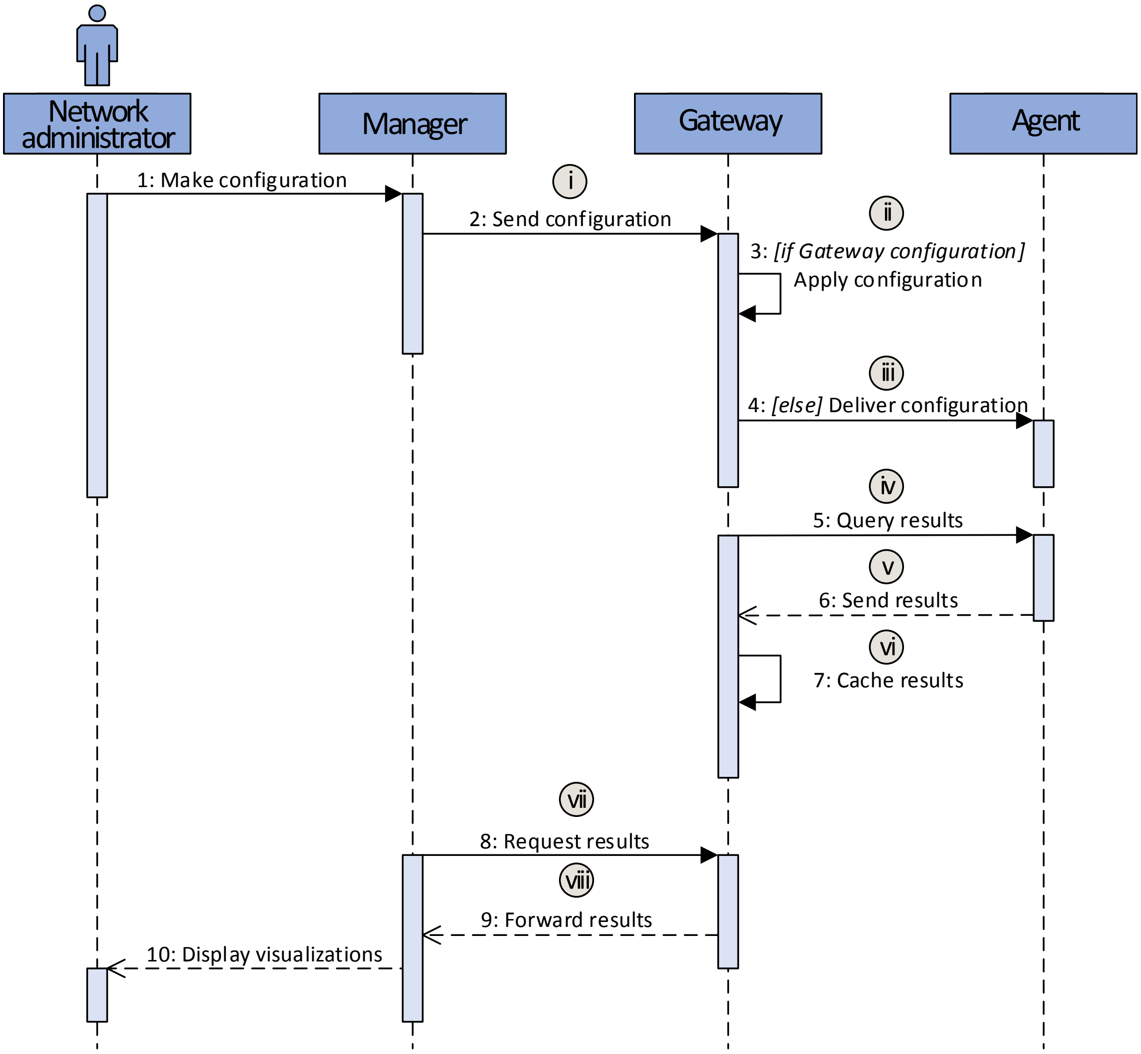
A CR network is usually composed of two physical entities: BS and CPE. In an infrastructured wireless communication system, as the network defined in the IEEE 802.22 Standard (IEEE, 2011), the BS is responsible to provide Internet access to CPEs. Moreover, the BS is also responsible for managing the channels utilization by the CPEs, coordinating which channels can be used by CPEs in a specific period of time. Thus, the BS is characterized by concentrating all the CR network information. For this reason, we design the second component of *Kitsune* to operate in the BS. This component is called Gateway and its main functionality is to provide the communication among the Manager and the radio devices of the CR network, *i.e.*, exchange management information between these two components. In Figure 3.3, we describe the messages exchanged by *Kitsune* components, the importance of the Gateway in the management communication.

Once defined the configuration of the CR devices by the network administrator (1), the Manager sends a configuration message to the Gateway (2). Then, the Gateway deliver to the Agents the configuration of each CPE. When the period to the Gateway obtain the cognitive functions results ends, the Gateway sends a request to the Agents (4). Then, the cognitive functions results are sent to the Gateway (5), which cache the results (6) and forward it to the Manager (7). Finally, the Manager composes and displays the results visualization to the network administrator (8).

The Gateway is one of the most important components in the *Kitsune* system, since all management messages should pass through this component. To the correct operation of the *Kitsune* system, ten management messages should be exchanged:

1. **Make configuration**: three types of configuration can be performed by the network administrator: one to all the CR network, one to the Gateway, and one to the CR devices. Once defined, the Manager sends the configuration to the Gateway.

Figure 3.3: Management communication in *Kitsune*



2. **Send configuration**: once defined the configuration to be applied in the CPEs by the network administrator, the Configuration module in the Manager component communicates with the Gateway component to send the CR devices configuration.

3. **Apply configuration**: the Gateway applies the configuration received from the Manager if this configuration is specific to the Gateway parameters.

4. **Deliver configuration**: if the configuration is to be applied to the CPEs, the Gateway should deliver the configuration received from the Manager to the Agents, selecting only the CPEs that were selected to be configured.

5. **Query results**: periodically the Gateway queries the results of the cognitive functions from the CPEs and stores these results in a local cache. The interval between requests can be configured using the Manager Configuration module. The Gateway requests the cognitive functions results from the CPEs in a smaller interval than

the Manager requests these results from the Gateway. It leads to the necessity of a cache to maintain the results collected earlier.

6. **Send results**: after receiving a message requesting the cognitive functions results from the Gateway, the Agent component should send the cognitive functions results organized in the MIB to the Gateway.
7. **Cache results**: the caching of the cognitive functions results by the Gateway is an important task in the hierarchical management model, because it enables the Manager to obtain results from the Gateway without major overheads caused by messages exchanged directly with CPEs.
8. **Request results**: periodically, the Manager should request the cognitive functions results to the Gateway, which should send all the cached new results to the Manager.
9. **Forward results**: once received the request sent by the Manager, the Gateway should select only the results that were not forwarded for the Manager in the last request, *i.e.*, new results, and forward it to the Manager
10. **Display visualizations**: after receive the cognitive functions results from the Gateway, the Manager should summarize these results and make the visualizations to the network administrator.

The Gateway is responsible for: (*i*) receiving the configurations sent by the Manager; (*ii*) applying the received configuration to its parameters, if the configuration is to the Gateway; (*iii*) delivering the configuration for the CR devices to the Agents; (*iv*) requesting the results of the cognitive functions to the CR devices; (*v*) receiving these results; (*vi*) caching these results; (*vii*) receiving the results request sent by the Gateway; and (*viii*) forwarding the results obtained to the Manager component. The operation of the Gateway module occurs aside of the remaining operations designed in the IEEE 802.22 Standard for BS to control the CPEs operation. Thus, the BS control operations are not interfered by the Gateway module. Moreover, due to the computational simplicity of the Gateway, the BS control operations are not compromised in terms of computational resources utilization, once the Gateway operation request few computational resources.

3.1.3 CPE Components

The IEEE 802.22 Standard defines a CPE as a generalized equipment set providing connectivity between a BS and a subscriber premise. In other words, a CPE is a physical entity responsible for providing network access to users. In *Kitsune*, the CPE entity is composed of three components: (*i*) MIB, (*ii*) Agent, and (*iii*) Cognitive Functions (CF). The MIB component is directly based on the IEEE 802.22 MIB, designed to the management of CR devices that compose infrastructured networks as defined in the IEEE 802.22 Standard. Using a MIB, the configuration of the CPEs can be defined and accessed in a simple and effective way. Moreover, the results of the cognitive functions are organized through the MIB, facilitating the access to this information.

We designed the MIB component by extending the one proposed in the IEEE 802.22 Standard. This extension was formulated to cover a gap of the original MIB, due to the fact that it cover only centralized network information, where the BS is responsible for the spectrum decision. By extending the MIB, we can apply this component to other types of CR networks, where the CPEs can perform the spectrum decision by themselves, enabling

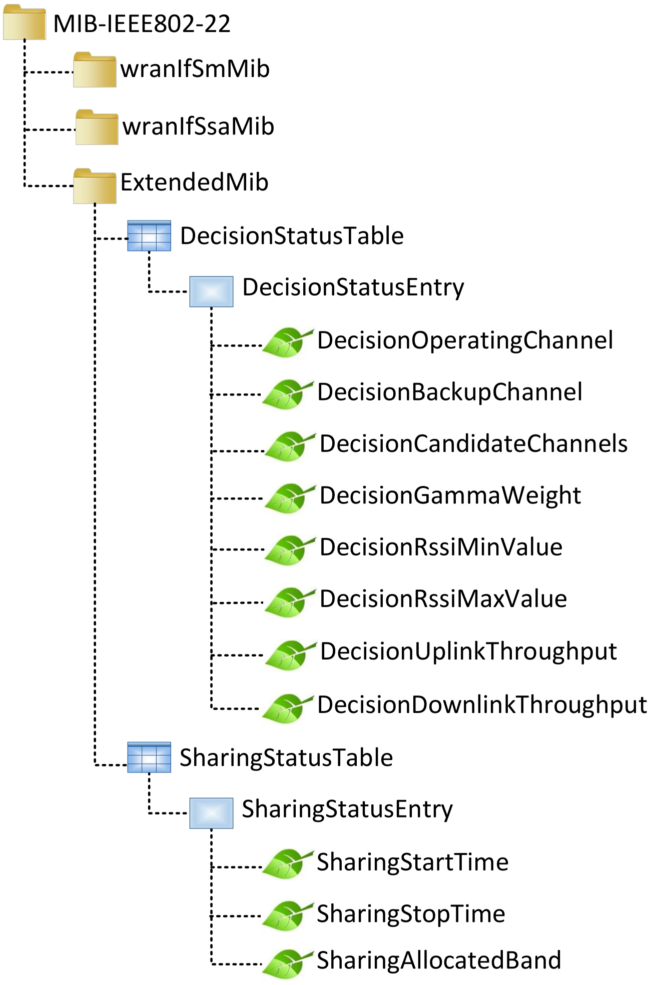
Kitsune to manage these networks. Moreover, this extension enables the management of the remaining of cognitive functions, *i.e.* the spectrum sharing and spectrum mobility, directly associated with the spectrum decision function. Moreover, we designed the extended MIB to operate in the CPEs, enabling the management of the cognitive functions performed by these devices. In Figure 3.4 can be observed the proposed extension to the IEEE 802.22 MIB.

The extension proposed is organized in a new group called *ExtendedMib*. In this group, we designed two tables: *DecisionStatusTable* and *SharingStatusTable*. The first one presents objects related to the spectrum decision function, while the second one brings objects referent to the spectrum sharing function. In the *DecisionStatusTable*, we created objects to the operating channel (*DecisionOperatingChannel*), backup channel (*DecisionBackupChannel*), and candidate channels (*DecisionCandidateChannels*). The values of these objects are resulting from the spectrum decision execution. The γ weight object (*DecisionGammaWeight*) is a specific value for the spectrum decision algorithm developed as proof of concept to *Kitsune* and will be explained in details in the next chapter. There are also objects representing the minimum and maximum values of the Received Signal Strength Indicator (RSSI) (*DecisionRssiMinValue* and *DecisionRssiMaxValue*), used in the spectrum decision algorithm to characterize the channels with better propagation conditions. The last objects in this table are the uplink and downlink throughput of the channel (*DecisionUplinkThroughput* and *DecisionDownlinkThroughput*), that represent the throughputs obtained by the CPE when the current operating channels was used in uplink and downlink directions. Additionally, in the *SharingStatusTable* we designed three objects: the timestamps when the CPE started and stopped the sharing over a operating channel (*SharingStartTime* and *SharingStopTime*), and the channel band allocated to transmit in this period of time (*SharingAllocatedBand*).

Although no specific objects to the spectrum mobility functions were designed, it does not mean that the management of this functions was neglected. The spectrum mobility function is directly associated with the spectrum decision function. Thus, by managing some objects of this function its is possible to indirectly manage the spectrum mobility. For example, by counting how many times the channels were selected as the operating channel, it is possible to make a visualization of the handoffs performed by each CPE, showing the mobility of this CR device over the channels. Moreover, another important visualization is how often a channel is selected to be the operating or the backup channel. It can demonstrate, for example, that some channels present good propagation conditions and/or low occupancy rate, making these channels suitable for transmission.

The Agent component is an intermediate between the MIB and CF components. The Agent receives messages with configurations from the Gateway and organizes them into the MIB. The configuration messages are sent through a control channel, defined at the beginning of the CR network operation. These configurations are used by the CF component to perform each cognitive function by the CF module. Moreover, the Agent collects cognitive functions results from the CF component and also organizes these results in the MIB. We designed an Agent that facilitates the addition of management capability to a device without impacting the operation of the CR device or its performance. The Agent may be added to a CR device slightly increasing the workload and demand on computational resources. Moreover, the Agent is able to communicate with a set of operations that remains the same on all CR devices. This component has the capability of supporting any type of information about any type of CR device that may be part of the CR network. It makes the Agent extensible, *i.e.*, new operations can easily be developed to the Agent.

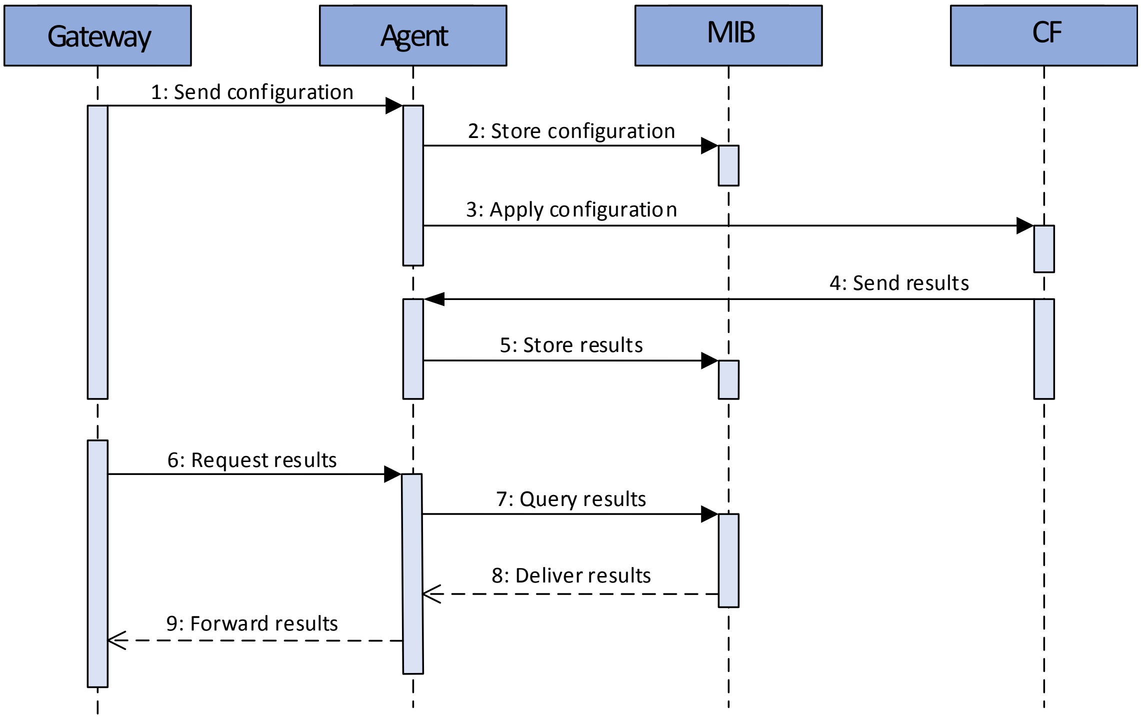
Figure 3.4: Proposed IEEE 802.22 MIB extension



CF component applies the new configurations to the cognitive functions. This component is responsible to directly communicate with the cognitive functions, working as a Hardware Abstraction Layer (HAL). The CF component executes the cognitive functions designed in each CPE, accessing specific directives of each hardware. For example, when a new configuration is set to the spectrum sensing, the CF component should call the hardware routines to perform this functions using the configuration organized in the MIB. Therefore, the cognitive functions should only be developed enabling its configuration by the CF component. This component is very important because it provides an abstraction level of the hardware used in the CR network, making *Kitsune* switchable to operate in different radio devices. It encompasses the CPE definition of the IEEE 802.22 Standard, that defines the CPE as a generic equipment *i.e.*, any hardware device able to provide the connectivity between the BS and the user. In Figure 3.5 we summarize the operation of the components present in the CPE entity.

The usual communication flow among the CPE components is performed exchanging nine messages, starting with the reception of a new configuration from the Gateway and finishing with the forward of the cognitive functions results to the Gateway. Each message is described in details above:

Figure 3.5: CPE components operation



1. **Gateway sends a new configuration**: when a new configuration for the CPEs is described by the network administrator, the Gateway receives this configuration and sends it to the Agents of the respective CPEs.

2. **Agent stores the configuration in the MIB**: after receiving a configuration, the Agent organizes it in the MIB.

3. **Agent applies the configuration to the CF**: the configuration is also applied to the CF component, responsible for performing the cognitive functions using this configuration.

4. **CF sends the results to the Agent**: once performed the cognitive functions, the CF should send to the Agent the results obtained.

5. **Agent stores the results in the MIB**: similarly to the configuration, the Agent organizes the results sent by the CF in the MIB.

6. **Gateway requests results to the Agent**: when the period defined for the Gateway request the cognitive functions results expires, it sends a request message to the Agents. Moreover, the Gateway can request the current configuration being used in the CR devices without a predefined interval, answering to a Manager request.

7. **Agent queries the results to the MIB**: after receiving the request message from the Gateway, the results stored in the MIB are queried by the Agent. These results

can be the cognitive functions results or the current configuration of the CPEs, according to the solicitation of the Gateway.

8. **MIB delivers the results to the Agent**: after be queried by the Agent, the MIB selects the respective results and delivers it to the Agent.

9. **Agent forwards the results to the Gateway**: finally, the Agent forward the results obtained from the MIB to the Gateway, finishing the results requests performed by the Gateway.

It is important to highlight that the operation of the CPE components can start in different modules. For example, the cognitive functions results can be forwarded to the Gateway using the same configuration used in the last forwarding, if no new configuration was received before the Gateway polling time to the Agent expires.

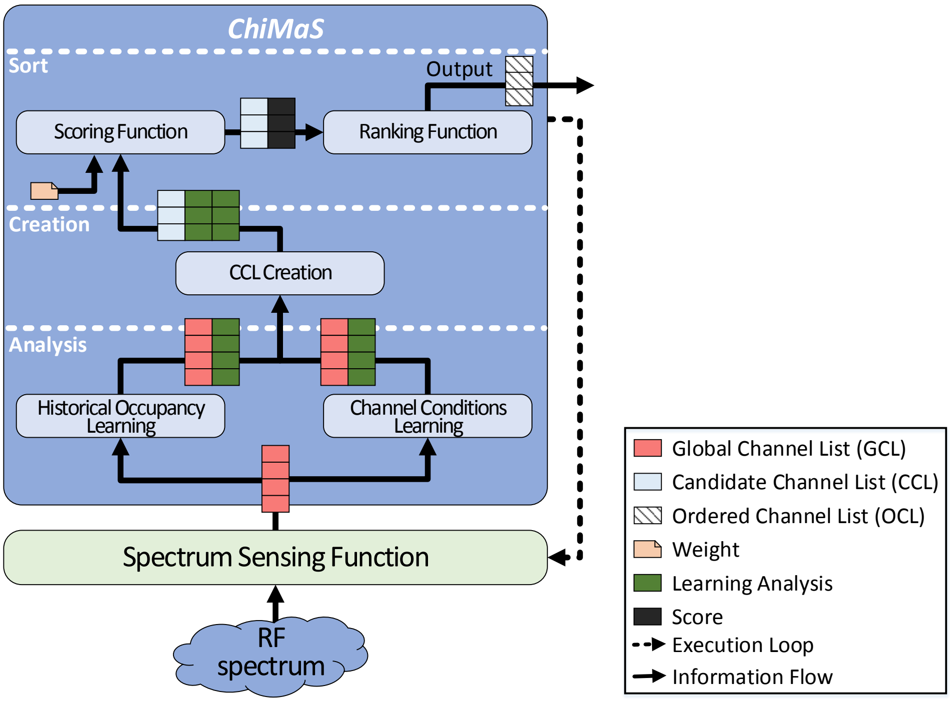
To complement the *Kitsune* system, we designed a channel list management system called *ChiMaS*. This system is used as the spectrum decision function, and indirectly performs the spectrum sharing and mobility functions, based on the spectrum sensing results. Once that *ChiMaS* represents these cognitive functions, we designed CF to comprise it, *i.e.*, execute and provide the results of *ChiMaS* execution. The objective using this system is to prove the proper operation of *Kitsune* in the management of the cognitive functions. In the next section, *ChiMaS* is described in details.

3.2 Channel List Management System (*ChiMaS*)

In decentralized CR networks, CR devices are able to take their own decisions, *i.e.* transmit without the direct intervention of a BS. As occur in centralized CR networks, after performing the spectrum sensing function, a CR device should define which channel will be used to transmit. The spectrum decision function is responsible for making this selection, by analyzing the results of the spectrum sensing. Finding the best channel in terms of quality is desirable to provide better transmission conditions to CR devices. However, there is a problem in finding high quality vacant channels in short periods of time, since the analysis of channel conditions demands more time than simply deciding whether or not the channel is vacant (AKYILDIZ et al., 2008). The amount of time spent to sense the spectrum is also related to the number of channels previously defined to be analyzed. Therefore, the IEEE 802.22 Standard (IEEE, 2011) defines that CR devices must keep a list of candidate channels to limit the duration of the spectrum sensing function. The existence of such list provides a reduction on the number of channels that should be sensed, since only the channels considered vacant in the last sensing must be in the list. This list can also be useful to decentralized CR networks. Taking into account decentralized CR networks scenarios and aiming to provide an effective way to find the best vacant channel we introduce *ChiMaS*, a solution to manage the Candidate Channels List (CCL) and find the best vacant channel. In Figure 3.6 we illustrate the *ChiMaS* components.

ChiMaS is divided into three management tasks, as can be seen in Figure 3.6. The first one, called Analysis, receives information from the spectrum sensing function regarding the occupancy status of a Global Channels List (GCL). This list comprises the group of channels previously defined to be analyzed by *ChiMaS*. The GCL is processed by two reinforcement learning algorithms to become aware about both the historical occupancy and conditions of each channel. Based on the results of such analysis, the second task, Creation, is responsible for the generation of the CCL. Finally, the Sort task uses a scoring

Figure 3.6: *ChiMaS* components



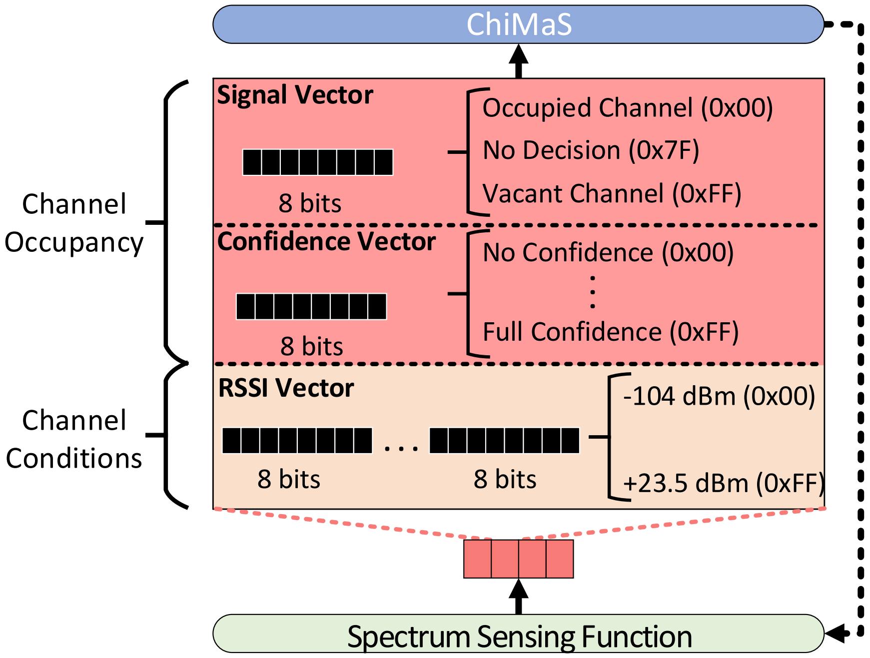
and a ranking function to obtain an Ordered Channels List (OCL), which is the output of *ChiMaS*. This output provides the best vacant channel to be used as operating channel. Moreover, based on this list, the backup channel may be selected, choosing the second channel in the list and the rest will stay in the CCL. Although *ChiMaS* enable CR devices to take their own decision, the *ChiMaS* results can be disseminated to other CR devices through a control channel, as defined by the IEEE 802.22 Standard (IEEE, 2011), allowing the CR devices to perform its transmissions according to the definitions of a BS. Thus, *ChiMaS* is suitable to be used in both decentralized and centralized CR networks, with some adaptations. In the next subsections, we explain in details the operations of the three tasks performed by *ChiMaS* during the spectrum decision.

3.2.1 Analysis Task

The Analysis task receives from the spectrum sensing function a GCL containing the channels to be analyzed by *ChiMaS* along with information regarding these channels. This information is composed of two types of data structures for each channel in the GCL, as defined by the IEEE 802.22 Standard and represented in the Figure 3.7.

The first data structure is a tuple composed of Signal Vector and Confidence Vector. The former contains information regarding the occupancy status of the channel. In this case, the spectrum sensing function must indicate if the channel is occupied (0x00), vacant (0xFF) or if it was unable to decide (0x7F). The latter carries information about the confidence of the spectrum sensing function in its current decision. The confidence level received by *ChiMaS* varies between 0 (0x00), indicating no confidence and 1 (0xFF), representing full confidence. The second data structure is a vector containing RSSI measurements. This vector ranges from -104 dBm (0x00) to +23.5 dBm (0xFF). Values outside this range shall be assigned to the closest extreme.

Figure 3.7: GCL data structures



GCL contains information about the occupancy status and conditions of each sensed channel. This information is analyzed by two reinforcement learning algorithms to update the knowledge about the channels. We chose reinforcement learning because it presents a good performance on centralized solutions (BKASSINY; LI; JAYAWEERA, 2012). The results of this analysis allow *ChiMaS* to evaluate the quality of the channels. The reinforcement learning algorithms were implemented adapting the specifications of Q-Noise+, proposed by Faganello *et al.* (FAGANELLO et al., 2013). Q-Noise+ is an evolution of the reward-based strategy adopted by the traditional reinforcement learning algorithm called Q-Learning (MENDES et al., 2011). The main adaptation we made was to eliminate the need for transmission to evaluate previous decisions, which is a drawback of Q-Noise+. Instead, we use information regarding the channel occupancy and conditions collected by the spectrum sensing function to make decisions.

The first learning algorithm of *ChiMaS* is called Historical Occupancy Learning, which is responsible for analyzing the usage profile of the channels. This analysis considers that the spectrum sensing function executions are performed in epochs (t). The goal of this feature is to use the spectrum sensing function information to assess the future occupancy of the channel. Towards this goal, a reward-based approach is applied considering two criteria to calculate Qh, which represents the results of the Historic Occupancy Learning. The criteria used to calculate Qh are: (*i*) the channel occupancy rate in the current epoch (r_t) and (*ii*) the weighted sum of this rate in a defined amount of past epochs (l). The former rate is defined by analyzing every partial analysis conducted by the spectrum sensing function to define whether or not the channel is occupied. This information is obtained in the Confidence Vector. Let G be the set of channels in the GCL, the Qh of a given channel c for the next epoch is then defined according to Equation 3.1.

$$\forall c \in G \Rightarrow Qh_t(c) = (1-\alpha)\sum_{i=1}^{l}[w_{t-i}r_{t-i}](c) + \alpha r_t(c) \tag{3.1}$$

where, $0 \leq \alpha \leq 1$ represents the weight of the reward (r_t) obtained in the last epoch. The number of past epochs to be considered for Qh calculation is defined by l. In this sense, w is the weight of each one of the last l epochs. This value is weighted by the weight of the past epochs, which is $(1 - \alpha)$.

Channel Conditions Learning is the second algorithm proposed in *ChiMaS*. This algorithm receives information about the mean RSSI level of a radio frequency channel to obtain knowledge about its conditions and calculate Qn, which represents the results of the algorithm. The criteria used to calculate the Qn are *(i)* the rate of RSSI in the current epoch (r_t) and *(ii)* the weighted sum of this rate in a defined amount of past epochs (l). It is important to highlight that RSSI measurements performed by the spectrum sensing function are considered by *ChiMaS* analysis task only in epochs where the channel is considered vacant, since in this case only noise is present. The Qn for a given channel c is calculated according to Equation 3.2.

$$\forall c \in G \Rightarrow Qn_t(c) = (1 - \beta) \sum_{i=1}^{l} [w_{t-i}\eta_{t-i}](c) + \beta\eta_t(c) \tag{3.2}$$

where $0 \leq \beta \leq 1$ is the weight of the current channel conditions and its complement is the weight of the conditions of past l epochs where the channel was considered vacant. Finally, η is a factor regarding to the channel conditions. This factor represents the reward of the Channel Conditions Learning. The better the channel, the higher η is.

3.2.2 Creation and Sort Tasks

In the Creation task, the CCL Creation function receives the GCL from the Historical Occupancy Learning and Channel Conditions Learning and creates the CCL taking into account the occupancy status of every channel in GCL. Only vacant channels are used to create the CCL. The results of the analysis of both historical occupancy and channel conditions of vacant channels are also part of the created list. It is important to emphasize that in the next execution all channels are sensed and analyzed, even those considered occupied by the spectrum sensing function in the current execution.

The Sort task is responsible for sorting the CCL using two functions, called Scoring and Ranking. The former receives the weight of both historical occupancy and channel conditions to calculate a score associated to each channel. The latter sorts the list according to the results of the Scoring Function. The obtained score is called Q-Value and indicates how suitable a channel is for opportunistic transmissions, considering its historical occupancy and conditions. In this sense, let C be the set of channels in the CCL. The Q-Value of a given channel $c \in C$ is obtained using Equation 3.3.

$$\forall c \in C \Rightarrow Q_{-Value} = \gamma * Qh_{t+1}(c) + (1 - \gamma) * Qn_{t+1}(c) \tag{3.3}$$

where γ is the weight of historical occupancy, and $(1-\gamma)$ represents the weight of channel conditions. The score of each channel is then processed by a Ranking function, which is responsible for finishing the sort and creating the OCL. Therefore, the most suitable channel for transmission, *i.e.*, the one with the highest Q-Value, will be placed in the beginning of the OCL, while the worst channel (with the lowest Q-Value) will be in the end of the same list.

All the configuration parameters of *ChiMaS* are managed by *Kitsune*, including the GCL, the γ weight, and the minimum and maximum RSSI values used in the Channels Condition Learning algorithm. Through the Manager GUI of *Kitsune*, the network administrator can configure the parameters values of *ChiMaS* in the CF component. Once finished the execution and obtained the OCL, the CPE is able to define its operating, backup, and candidate channels. The results are also managed by *Kitsune*, including the operating, backup and candidate channels in the MIB. These results are used to compose the visualizations of the spectrum decision, sharing, and mobility functions.

Once presented and explained the entities, components and modules that compose *Kitsune*, in the next chapter, we describe a prototype implementation of the proposed system based on the definitions presented in this chapter, aiming to prove the feasibility of *Kitsune* in the management of CR networks.

4 PROTOTYPE

Based on *Kitsune* system proposed in the previous chapter, we developed a prototype of *Kitsune* to prove its proper operation in the management of CR networks. In the next subsections, we detail the implementation of each component present in the *Kitsune* prototype, according to the respective entity where the component was deployed. First, in Section 4.1 we present an overview about the *Kitsune* prototype. Next, in Section 4.2, the Manager component placed in the Management Station is described. In Section 4.3 we present the Gateway component deployed in the BS. Finally, in Section 4.4 the Agent, MIB, and CF components operating in the CPE have their implementation and operation described in details.

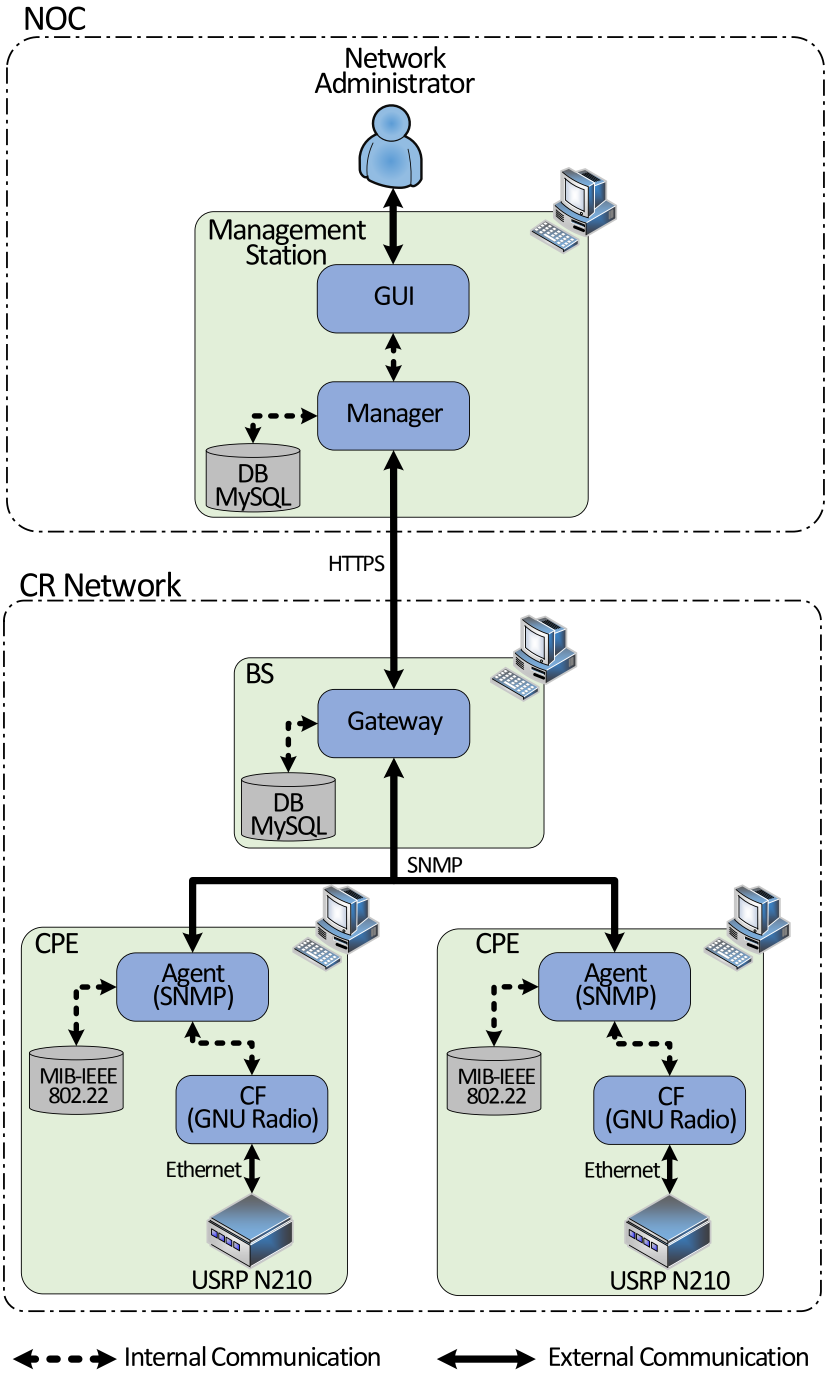
4.1 *Kitsune* Prototype Overview

Kitsune prototype provides the components present in the management entities in both the NOC and the CR network functional blocks, as previously defined, *i.e.*, the Manager component present in the NOC and the Gateway, Agent, MIB and CF components present in the CR network. In each block, specific entities are deployed and the information exchange among components of each entity may be performed through internal communication (inside the same device or computer) or external communication (among different devices and computers). In Figure 4.1, we illustrate the prototype implementation, according to the technologies used in each component.

We implemented the *Kitsune* prototype using four personal computers (PCs), one representing the Management Station, one for the BS, and two for the CPEs. These PCs are composed of Intel Core 2 Duo processors with 2.33 GHz of frequency, 2 GB of memory, and equipped with the operating system Ubuntu version 12.04 32 bits. In the CPEs, the PCs are connected to Ettus USRP N210 devices, which are computer-hosted Software Defined Radios (SDR)s that enable the development of radio applications using a specific toolkit. To implement software radio applications, we chose the GNU Radio, a free and open-source software development toolkit that provides signal processing blocks to use Universal Software Radio Peripheral (USRP) devices. In both the Management Station and BS, the caches are implemented using MySQL databases, due its rapid and simple development. The communication between the Management Station and BS, as the communication between the BS and the CPEs are cabled, using Ethernet connections. The Management Station and BS communication is performed over an encrypted connection using the Hypertext Transfer Protocol Secure (HTTPS), ensuring the security in the communication with low messages overhead, while the communication among BS and CPEs is performed using SNMP. The use of a secure protocol in the communication among the Management Stations and BSs was defined because messages over the Internet or

LANs can be monitored more simply than the message among BSs and CPEs, which use a dedicated control channel.

Figure 4.1: *Kitsune* system prototype



The Manager access the desired information using a set of services provided by the Gateway. The Manager should compose a request message with specific parameters to be sent to the Gateway, which processes the request and sends to the Manager a return message with the desired informations. The *Kitsune* services are based on the services proposed by Marotta *et al.* (MAROTTA et al., 2013), and are are presented in Table 4.1, with the respective parameters and returns.

Table 4.1: *Kitsune* Services

Service	Parameters	Return
beginSession	*user name, password*	*sessionID*
endSession	*sessionID*	-
setConfiguration	*sessionID, configuration*	*configStatus*
setCpeConfiguration	*sessionID, cpeConfiguration*	*cpeConfigStatus*
getConfiguration	*sessionID, configID*	*configuration*
getCpeConfiguration	*sessionID, configID*	*cpeConfiguration*
getCfResults	*sessionID*	*cfResults*

Accordingly to Table 4.1, when a Manager requests the service *beginSession*, the Gateway creates a session, authenticated by the *username* and *password* parameters. This service starts an authorized session, identified by the *sessionID*, that can be finished using the *endSession* service. To configure the Gateway, the *setConfiguration* service can be used, sending the *sessionID* and a *configuration*, which will return a *configStatus*. Moreover, the *setCpeConfiguration* service can be performed to directly configure each CPE and return a *cpeConfigStatus* with the actual configuration status. The *getConfiguration* service returns the information about the current configuration of the Gateway, returning a *configuration*. Similarly, the *getCpeConfiguration* service returns the information about a CPE configuration, identified by the *configID* parameter. The result of this service is a *cpeConfiguration*, with the configuration parameters and their actual values. Finally, the *getCfResults* service returns the cognitive functions results from all the CPEs monitored in the Gateway.

Each service should be requested by the Manager using a JavaScript Object Notation (JSON) message, a lightweight description language. When the request is received by the Gateway, it should process and return a JSON with the information requested by the Manager. In Section 4.2, we describe in details how this message is composed of the Manager. Moreover, in Section 4.3, we detail how the Gateway process the requests for each service and the respective returns. Finally, in Section 4.4, we explain the iteration of the CPE components with the services executed in the Gateway.

4.2 Manager

The Manager component is interested in the access different services from the Gateway. However, each service requires a specific request, *i.e.*, a JSON message with specific parameters. To solicit a specific service, the Manager should compose a JSON message and perform a requisition to the Uniform Resource Identifier (URI) of the service. In the next subsections, we describe in details how the requests are composed by the Manager.

4.2.1 beginSession Service Request

To access a service in the Gateway, the Manager must first validate itself through the *beginSession* service. It means that *beginSession* enables the Manager component to use other services from the Gateway, creating a session by validating parameters, which includes at least the service URI identifier *service_uri*, a *username* and a *password*. The *beginSession* request is depicted in the Listening 4.1.

```
        {
        "service_uri":"http://gateway_IP/services/beginSession.php
           ",
        "passwd":"secret",
        "owner":"kitsune-manager",
        "community":"adm",
        }
```

Listing 4.1: beginSession JSON request message

The *beginSession* request message is composed of different parameters. These parameters are the user name (*owner*) of the network administrator, the respective password (*passwd*) of the network administrator, and a community name (*community*), which delimits the privileges, capabilities, or permissions of the user to request the services.

4.2.2 endSession Service Request

Once validated, the Manager may use different services at the Gateway such as described in next subsections. However, the Manager may want to end a started session to start a new one with another user, capabilities, or permissions. A typical example of request message related to the *endSession* service can be seen at Listing 4.2.

```
        {
        "service_uri":"http://gateway_IP/services/endSession.php",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44"
        }
```

Listing 4.2: endSession JSON request message

The Manager sends a message to an URI that points to the *endSession* service on the Gateway. This message contains solely the service URI identifier *service_uri* and a valid session identification (*sessionID*) previously created using the *beginSession* service. Thus, the session identified by the session identification can be destroyed by the Gateway.

4.2.3 setConfiguration Service Request

After be validated in the Gateway, the Manager is able to send the configuration to the Gateway. Therefore, the Manager can request a *setConfiguration* service on the Gateway to deploy a new configuration. These parameters are provided by the Manager and can be better observed in the JSON request message depicted in Listing 4.3.

```
        {
        "service_uri":"http://gateway_IP/services/setConfiguration.
           php",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44",
        "configuration":{
                "pollInterval":"2",
                "requestTimeout":"5",
                "clearCache":"yes"
                }
        }
```

Listing 4.3: setConfiguration JSON request message

Different configurations can be applied in the Gateway, as can be seen in Listing 4.3. The Manager sends the request message with the service URI identifier *service_uri*, a valid session identification (*sessionID*), and the parameters with respective values to be configured. In Listing 4.3, we exemplify the configuration of the poll interval to the Gateway requests the results for the CPEs (*pollInterval* set to 2 seconds), the timeout of the requests performed by the Gateway to the CPEs (*requestTimeout* set to 5 seconds), and the command to clear all the registries in the cache (*clearCache* set to “yes”, *i.e.*, clear the cache). All these parameters can be selected using *Kitsune* configuration module, with the values desired by the network administrator.

4.2.4 setCpeConfiguration Service Request

The Manager is also able to send the configuration to the CR devices, after be validated in the Gateway. The Manager can request a *setCpeConfiguration* service on the Gateway to deploy a new configuration on the CR devices. This configuration is performed directly by selecting what CPEs and parameters will be configured. These parameters are provided by the Manager and can be better observed in the JSON request message in Listing 4.4.

```
        {
        "service_uri":"http://gateway_IP/services/
           setCpeConfiguration.php",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44",
        "cpeConfiguration":{
                "cpeIDs":"1,2,4",
                "silentTime":"2",
                "maxTxTime":"2",
                "sensingChannels":"1,3,4,5,6",
                "decisionRssiMinValue":"-104",
                "decisionRssiMaxValue":"+23.5",
                "decisionGammaWeight":"0.8"
                }
        }
```

Listing 4.4: setCpeConfiguration JSON request message

In this message, the Manager should define the list of CPEs to be configured (*cpeIDs*). Moreover, the set of parameters and respective values to be configured are also sent in this JSON message. For example, in Listing 4.4 we have the period in which CPEs are not able to transmit (*SilenceTime*), the maximum transmission time (*MaxTxTime*), the list of channels being sensed (*SensingChannels*), the minimum and maximum values to the RSSI used in *ChiMaS* (*DecisionRssiMinValue* and *DecisionRssiMaxValue*, respectively), and the γ weight for *ChiMaS* (*DecisionGammaWeight*).

4.2.5 getConfiguration Service Request

During the operation of the CR network, the current configuration in the Gateway can be collected by the Manager using the *getConfiguration* service. The current configuration corresponds to the last one described by the network administrator and sent to the Gateway from the Manager, using the *setConfiguration* service described previously. The *getConfiguration* service can be used to obtain the configuration parameters and respective values being used in the current Gateway operation. This service is required by the Manager through the Listing 4.5, outlined below.

```
        {
        "service_uri":"http://gateway_IP/services/getConfiguration.
           php",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44",
        "configID":"currentConf_INDEX",
        }
```

Listing 4.5: getConfiguration JSON request message

The Manager must send to the *getConfiguration* service a JSON request message with the service URI identifier *service_uri* and a session identification (*sessionID*). In addition, the Manager must inform the configuration identification *configID* which it desire to obtain the actual parameters values.

4.2.6 getCpeConfiguration Service Request

The *getCpeConfiguration* service enables the Manager to obtain the current configuration parameters and respective values from the CPEs. It is an important service, once that different CPEs can be configured with different configuration. Thus, the Manager can solicit to the Gateway what CPEs are using a specific configuration and which are the current values of the parameters used, as can be seen in the JSON Listing 4.6.

```
        {
        "service_uri":"http://gateway_IP/services/
           getCpeConfiguration.php",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44",
        "configID":"currentConf_INDEX",
        }
```

Listing 4.6: getCpeConfiguration JSON request message

The Listing 4.6 is composed of the service URI identifier *service_uri*, the session identifier (*sessionID*), and the configuration identifier (*configID*), like the 4.5 explained in the previous subsection. It is important to emphasize that, despite the 4.5 and the 4.6 are equals, the Gateway can identify what service is being requested due the specific URI requested by the Manager.

4.2.7 getCfResults Service Request

Finally, the *getCfResults* service is responsible to provide the cognitive functions results of the CPEs to the Manager. Using this service, the Manager must receive all the cognitive functions results, from all CPEs. This service is requested by the Manager using the JSON request message exemplified in the Listing 4.7.

```
        {
        "service_uri":"http://gateway_IP/services/getCfResults.php
           ",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44",
        }
```

Listing 4.7: getCfResults JSON request message

When the Manager requests the *getCfResults* service, it should only inform the service URI identifier *service_uri* and the session identifier (*sessionID*) in the JSON request message. The Gateway must identify this message using the URI of the *getCfResults*, process the service and return to the Manager the results of the cognitive functions solicited by the Manager. The Manager summarizes the cognitive functions results and creates a set of visualizations for the network administrator using a Web based Graphic User Interface (GUI). In the next subsection, we explain the GUI implemented in the *Kitsune* prototype.

4.2.8 *Kitsune* GUI

The configuration of the CR devices is possible using the Configuration module present in the Manager component of *Kitsune*. This module is directly accessed using the *Kitsune* GUI. Using the GUI, the network administrator can manage the entities of each CR network (BS and CPE), analyzing the cognitive functions results and changing the configurations applied to the CR devices. The GUI is directly associated with the Configuration and Visualization modules of the Manager, once that the configuration for the CR devices are described and the cognitive functions results are summarized in specific visualizations in the GUI. We developed three different menus in the GUI: Network, BS and CPE menu. These menus are very similar with each other, differing only in the specific parameters and visualizations available for the network administrator analysis. In Figure 4.2 we present the *Kitsune* GUI, showing the BS menu.

All the three menus presents similar options. As can be seen in Figure 4.2, at the left side, an options menu is provided with the main operations related to the selected entity. These options are: (*i*) List Base Stations, to list all registered BSs in the selected network domain; (*ii*) Create a Base Station, to register a new BS in the network; (*iii*) Update a Base Station, used to update the information about a specific BS, such as identifier, name, description, geolocation, etc.; (*iv*) Delete a Base Station, to remove a specific BS registry from the selected network; and (*v*) Manage a Base Station, which turns possible to visualize and modify the current configuration of the BS.

In the right side of Figure 4.2, the visualizations about the results collected by the BS are presented in collapsed menus, which can be expanded to show the summarizations about (*i*) CPEs information, (*ii*) channels occupation, (*iii*) channels RSSI, (*iv*) throughput and confidence average of each channel, and (*v*) the map with the BS and CPEs geolocations with the estimated coverage area. The visualizations are composed using two graphical Application Programming Interfaces (APIs) widely applied to compose dynamic charts: the Google Charts[1] and the D3.js[2] APIs.

The Manager was implemented using Yii[3], a high-performance Hypertext Preprocessor (PHP) framework best for developing Web 2.0 applications. The summarizations available in the GUI are stored in a MySQL database and queried by the Manager in a predefined time interval, *i.e.*, the network administrator can define how often the Manager request the results to the Gateway. This interval can be configured in the Manager and must be equal or longer than the interval in with the Gateway query the CPEs. The Manager and Gateway communication follow ROA because it enables a better utilization of the Internet infrastructure.

4.3 Gateway

The Gateway uses a MySQL database in combination with services provided to the Manager using the PHP language. The databases of both the Manager and the Gateway are very similar, differing only in the timestamps of the received information. The

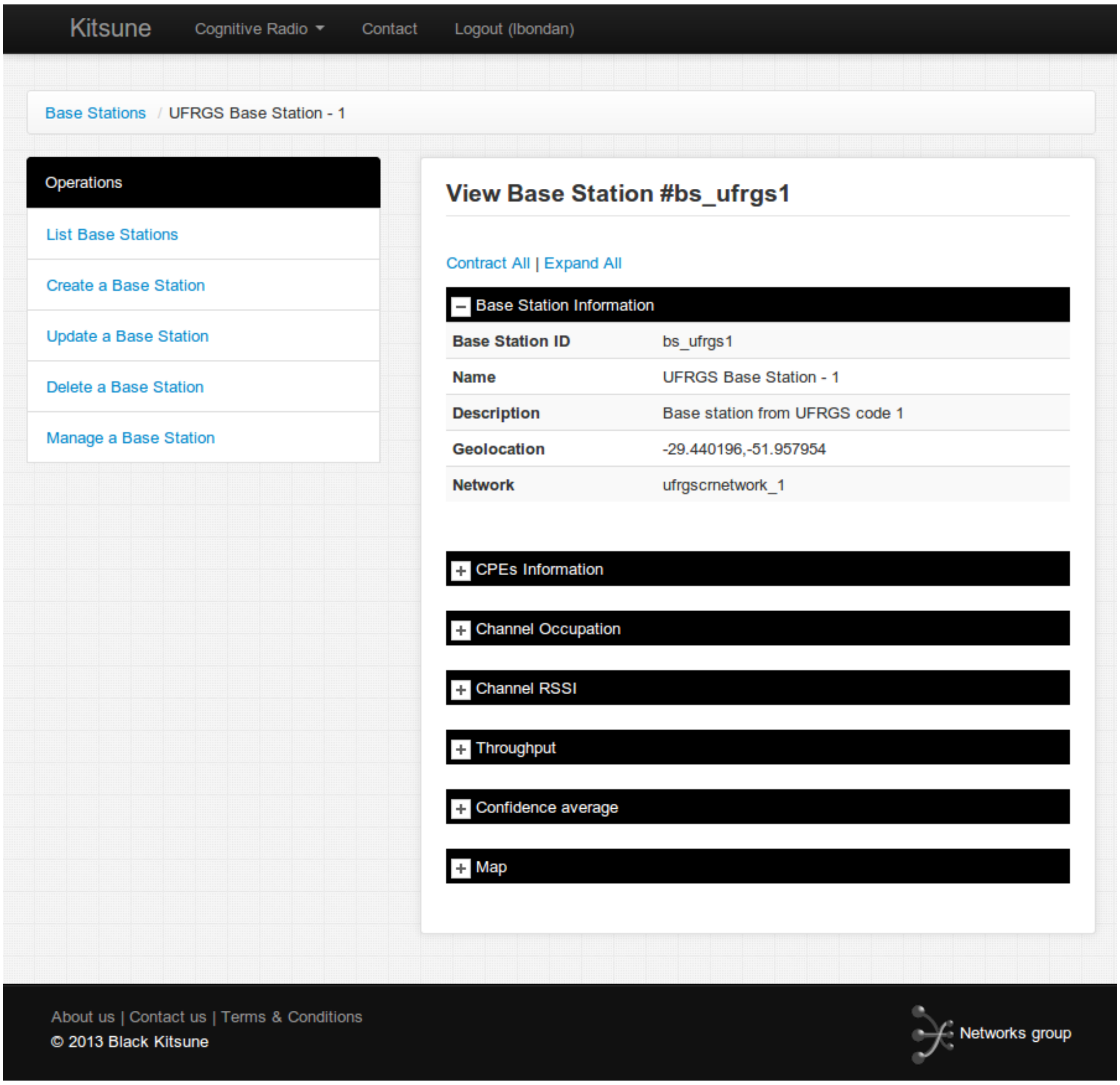
---

[1] https://developers.google.com/chart/

[2] http://d3js.org/

[3] http://www.yiiframework.com/

Figure 4.2: *Kitsune* GUI



View Base Station #bs_ufrgs1

Base Station Information

Base Station ID	bs_ufrgs1
Name	UFRGS Base Station - 1
Description	Base station from UFRGS code 1
Geolocation	-29.440196,-51.957954
Network	ufrgscrnetwork_1

Gateway operates following the time intervals configured in the Manager. When a new CPE configuration is received by the Gateway, it stores the configuration in the MySQL database and forwards it to CPEs. Similarly, the Gateway retrieves the cognitive functions results stored in the MIB into CPEs. When the results are received by the Gateway, they are also stored in the database and then, sent to the Manager. The Manager access the desired information through the set of services summarized in Table 4.1. Once received, the Gateway must process the request and send a return message to the Manager. In the next subsections, we describe in details how the Gateway process these services and the format of the JSON return messages.

4.3.1 beginSession Service Processing

Once received by the Gateway a *beginSession* service request sent by the Manager, the Gateway must validate the user and password received. This validation is performed by authenticating the received user and password with local information previously registered. If the user and password match with the local information, the Manager component become authorized to use other services. The Gateway responds to the Manager with a valid *sessionID*. Without a valid *sessionID* the Manager cannot use any of the other available services at the Gateway. The *beginSession* return is depicted in the Listening 4.8.

```
        {
        "status":"accepted",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44"
        }
```

Listing 4.8: beginSession JSON return message

The Gateway can validate each parameter and, in case of acceptance, it replies the first message with an operation status as accepted (*status*) and a valid unique session identification (*sessionID*). Otherwise, a simple message is replied indicating the failure attempt with a description containing an error message. Finally, the Manager authenticated during *beginSession* execution will be validated in the Gateway.

4.3.2 endSession Service Processing

An *endSession* service is designed to perform a session close in the Gateway. The Gateway processes the *endSession* request message sent from the Manager and destroys the session defined by the given session identification. A typical example of messages related to the *endSession* service return can be seen at Listing 4.9.

```
        {
        "status":"destroyed",
        "sessionID":"2f4d941a1359301b243ab41a3b405c30944c3f44"
        }
```

Listing 4.9: endSession JSON message exchange

When the session is successfully destroyed, a message from the Gateway to the Manager is sent, containing the status of the operation (*status*) and the session identification (*sessionID*) of the destroyed session. Otherwise, a message containing a status of "none" is replied indicating that the session identification provided is wrong. In addition, a status containing “error” may be returned when related to the Manager that is waiting for a service to be executed, therefore, the session is locked and cannot be destroyed.

4.3.3 setConfiguration Service Processing

After the validation process, the Manager can properly configure the Gateway, using the *setConfiguration* service. The Gateway process a *setConfiguration* request message by analyzing each parameter sent by the Manager and using these values to configure the Gateway operation. In return, the Gateway sends to the Manager a JSON message, as depicted in Listing 4.10.

```
        {
        "configID":"currentConf_INDEX",
        "configStatus":"applied",
        "errors":"none"
        }
```

Listing 4.10: setConfiguration JSON return message

Once processed, the Gateway sends to the Manager a status message with the configuration identifier (*configID*) and its status (*configStatus*). Finally, any possible errors occurred during the configuration are replied through an error field (*errors*). The Manager can also reset the configuration in the Gateway, requesting a *setConfiguration* service with a valid session identification (*sessionID*) and the option to reset the current configuration (*resetConfig*) set to “1”. In this case, the Gateway will restart its operation assuming default values to the configuration parameters.

4.3.4 setCpeConfiguration Service Processing

The *setCpeConfiguration* service allows the Manager to configure the CPEs of the CR network. When the Gateway receive a *setCpeConfiguration* request message, it must extract the values of each parameter in the message, save the values in its cache, and compose SNMP SET messages to send for each CPE informed by the Manager. The CPEs are identified by a *cpeID*, with is directly associated with the Internet Protocol (IP) address of the CPE. Thus, the Gateway sends one SNMP SET message for each parameter of each CPE informed in the *setCpeConfiguration* service request message sent by the Manager. In response, the Gateway sends to the Manager a return message, as depicted in Listing 4.11.

```
{
"cpeConfigID":"currentConf_INDEX",
"cpeConfigStatus":"applied",
"errors":"none"
}
```

Listing 4.11: setCpeConfiguration JSON return message

After processing the *setCpeConfiguration* service request, the Gateway sends to the Manager a status message with the configuration identification (*cpeConfigID*), its status (*cpeConfigStatus*), and provides a report if any errors occurred (*errors*). Similarly as occurs with the *setConfiguration*, the *setCpeConfiguration* also provides a way to reset the configuration of CPEs to its default, by setting to “1” the option to reset the current CPEs configuration (*resetConfig*).

4.3.5 getConfiguration Service Processing

The Gateway can return to the Manager all the current values of the configured parameters. The Manager must send a *getConfiguration* request message, which is processed by the Gateway that returns with Listing 4.12, outlined below.

```
{
"configuration":{
        "configID":"currentConf_INDEX",
        "status":"running",
        "pollInterval":"2",
        "requestTimeout":"100",
        "clearCache":"no"
        }
}
```

Listing 4.12: getConfiguration JSON return message

The Gateway sends back to the Manager the current information, related to: the status of the configuration (*status*), the poll interval (*pollInterval*), the timeout to the requests performed to the CPEs (*requestTimeout*), and the flag that identifies if the cache was cleared (*clearCache*).

4.3.6 getCpeConfiguration Service Processing

Once received a *getCpeConfiguration* service request message, the Gateway will summarize all the CPE configurations stored in its cache and send it back to the Manager, using a *getCpeConfiguration* return message, as can be seen in the JSON Listing 4.13.

```
        {
        "cpeConfiguration":{
                "cpeConfigStatus":"running",
                "cpeIDs":"1,2,4",
                "silentTime":"2",
                "maxTxTime":"2",
                "sensingChannels":"1,3,4,5,6",
                "decisionRssiMinValue":"-104",
                "decisionRssiMaxValue":"+23.5",
                "decisionGammaWeight":"0.8"
                }
        }
```

Listing 4.13: getCpeConfiguration JSON return message

All parameters and their configuration values requested must be send back to the Manager by the Gateway, including the period in which CPEs are not able to transmit (*SilenceTime*), the maximum transmission time (*MaxTxTime*), the list of channels being sensed (*SensingChannels*), the minimum and maximum values to the RSSI used in *ChiMaS* (*DecisionRssiMinValue* and *DecisionRssiMaxValue*, respectively), and the γ weight for *ChiMaS* (*DecisionGammaWeight*), among others parameters previously configured by the network administrator using the Manager.

4.3.7 getCfResults Service Processing

The *getCfResults* is responsible for processing the request sent by the Manager to obtain the cognitive functions results of the CPEs. The Gateway process the *getCfResults* service request message by selecting in its cache all the results that were not sent to the Manager yet. The results are queried and stored in the cache periodically, and the results of all cognitive functions are returned to the Manager when the *getCfResults* service is requested. We decided to return all the results due to the fact that the results are obtained through polling, *i.e.*, at some time, all the cognitive functions must have their values obtained. Thus, requesting all, no additional messages must be exchanged to make each cognitive function visualization, reducing the management messages overhead and improving the time expended to make the visualizations. The JSON return message format is illustrated in Listing 4.14.

The JSON message returned by the Gateway to the Manager is structured with the time when the cognitive functions results were collected by the Gateway (*timestamp_d-hh:mm:ss*). This time stamp has a day index and the hour, minutes, and seconds when the cognitive functions results were received. Moreover, this time stamp is organized by the results returned by each CPE (*cpe_1, cpe_n*). The results returned by CPEs include: list of occupied channels (*occupiedChannelSet*); time in which the last spectrum sensing was performed (*timeLastSensing*); operating channel (*operatingChannel*); backup channel (*backupChannel*); candidate channels (*candidateChannels*); sharing start and stop times (*sharingStartTime* and *sharingStopTime*); allocated band in the transmission (*sharingAllocatedBand*); channel throughput obtained (*channelThroughput*); among others results organized in the MIB component present in the CPE.

```
        {
        "timestamp_d-hh:mm:ss":{
                "cpe_1":{
                        "configID":"currentConf_INDEX",
                        "cpeConfigStatus":"running",
                        "cpeID":"1",
                        "occupiedChannelSet":"1,3,5,7",
                        "timeLastSensing":"12:34:55",
                        "operatingChannel":"2",
                        "backupChannel":"4",
                        "candidateChannels":"6,8,9,10",
                        "sharingStartTime":"12:34:44",
                        "sharingStopTime":"12:34:45",
                        "sharingAllocatedBand":"2",
                        "channelThroughput":"110"
                        }
                ...
                "cpe_N":{
                        ...
                        }
                }
        ...
        "timestamp_d-hh:mm:ss":{
                ...
                }
        }
```

Listing 4.14: getCfResults JSON return message

4.4 MIB, Agent and CF

We designed an extension to the IEEE 802.22 MIB, aiming to cover information that is not directly covered by the default version of this MIB. The extended MIB was implemented using the Net-SNMP framework[4], a suite of applications used to implement solutions for SNMP communication. The MIB structure is described using the Structure of Management Information (SMI) format. Using SMI, it is possible to describe all the managed objects using different definitions for each object. For example, in the SMI file, the objects related to the configuration of the CR devices (*e.g. wranIfSmSsaChAvailabilityCheckTime* and *DecisionOperatingChannel*) are defined as "read-write", *i.e.*, the object values can be read (using SNMP GET) and written (SNMP SET). On the other hand, objects related to the cognitive functions results (*e.g. wranIfSsaTimeLastSensing* and *DecisionOperatingChannel*) are not writable, because are defined as "read-only". It avoids situations in which, for some reason, a SNMP SET message was performed to change a result value, invalidating the real information of the object.

The detailed SMI description of the MIB used in *Kitsune* prototype can be seen in Appendix A. Once described the MIB structure in the SMI file, we implemented the Agent component to handle the MIB using C programing language. The Agents are deployed in personal computers with an Ettus USRP N210 device connected. To create

---

[4] http://www.net-snmp.org/

the radio applications, we chose the GNU Radio, a toolkit widely used in the academic and industrial areas, due to its flexibility in terms of applications and supported hardware. In summary, the USRP device combined with the GNU Radio represent the radio frontend of the CPE, providing some level of abstraction to access the RF spectrum, avoiding the need of precise information about the hardware configuration.

The USRP device used in the prototype operates according to the algorithms implemented in a PC, connected through Gigabit Ethernet interface. The algorithms represent the CF component, developed as Python scripts that interacts with the Agent component to obtain its parameters values and return its results. These scripts implement the cognitive functions, *i.e.* a spectrum sensing technique based on energy detection and a spectrum decision function based on a channel list management system. The scripts use the parameters configured by the network administrator, such as the list of channels to be used for transmissions, sensing period, sensing window, and maximum bandwidth per channel. Finally, results of the CF are stored in the MIB and both the configurations and results are manipulated by the Agents.

To complete the *Kitsune* prototype implementation, we designed and implemented the channel list management system, or *ChiMaS*. This system is used as the spectrum decision function, and indirectly performs the spectrum sharing and mobility functions. The objective using this system is to prove the proper operation of *Kitsune* in the management of the cognitive functions. The CF component is responsible for the execution of *ChiMaS*, using the configuration defined in the Manager and forwarding the results to the Agent.

Once defined the *Kitsune* system and the prototype implementation, in the next chapter we present in details how we deployed *Kitsune* system prototype in an experimental RF scenario to be evaluated. In addition, we describe the experiments performed and the scenario where *Kitsune* was evaluated, discussing in details the results obtained.

5 EXPERIMENTAL EVALUATION

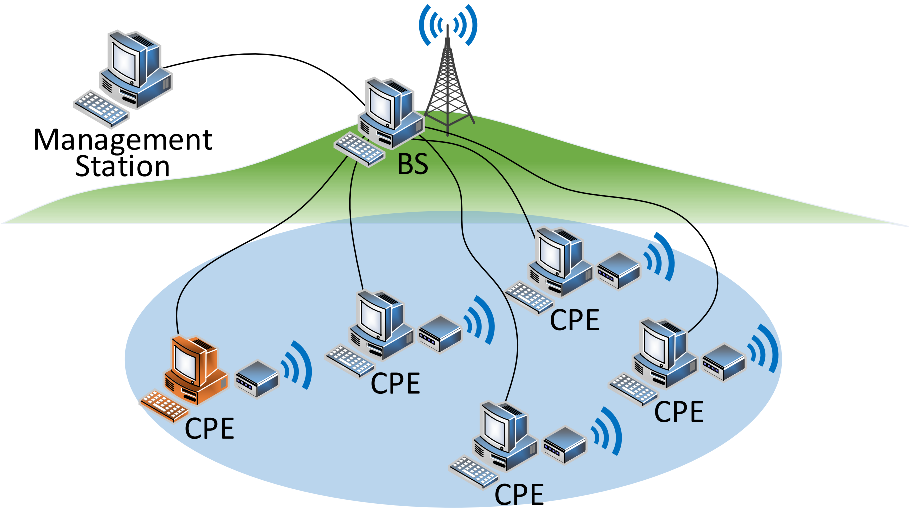
In this chapter we present and discuss the results obtained in the evaluation of *Kitsune* in an experimental RF scenario. In Section 5.1, we describe the evaluation scenario used. In Section 5.2, the results of the spectrum sensing visualizations are presented. In Section 5.3, we describe and discuss the spectrum decision results. We present the results of the spectrum sharing visualization in Section 5.4. The visualizations resulting of the spectrum mobility are presented in Section 5.5. We also present the results obtained by changing the configuration of the CR devices, in Section 5.6. Finally we compare *Kitsune* with the main proposals found in the literature to CR networks management in Section 5.7.

5.1 Evaluation Scenario

Kitsune prototype was deployed according to a scenario based on IEEE 802.22 Standard, where analogical TV channels are used to provide Internet access for CPEs (IEEE, 2011). In all the experiments CPEs should avoid interfere in the transmissions performed by licensed users in the same region. In some experiments, the CPEs should operate according to BS decision, *i.e.* under the control of BS. However, to explore situations where CPEs have their operation apart from BS decisions, some experiments where performed considering that CPEs are able to decide which are the best channels to be used, without BS intervention. We defined a CR network containing a BS and five CPEs in a same region, with a radius not greater than 10 meters. In some experiments, one CPE was configured to represent a licensed user using RF spectrum channels. This configuration was done to evaluate if the visualizations are properly representing the periods where licensed users are using the RF spectrum. Moreover, a Management Station was connected to BS, responsible for configure, monitoring, and composing the visualizations of the information obtained from the CR devices. An illustration of the proposed scenario can be seen in Figure 5.1.

Five experiments were designed in order to evaluate the management of the cognitive functions supported by *Kitsune*, each one exploring the experimental scenario using different configurations. Each experiment was performed during 60 seconds, considering an USRP sample rate of 10^6 samples per second. In our evaluations, until 5 channels and a BS with until 5 associated CPEs. For each channel, its status was modeled through a Poisson distribution with a mean and variance (λ), such as presented by Ghosh *et al.* (GHOSH et al., 2010). This distribution was used to model the licensed user behavior over the channels. In addition, these channels were sensed 0.1 seconds. Afterwards, the interval between each spectrum sensing function execution was set to be a different period for each experiment. The maximum bandwidth per channel was set to 6 MHz, according to the IEEE 802.22 Standard (IEEE, 2011). The polling waiting time from Manager to Gateway (P_M) was set to 30 seconds and from Gateway to Agents (P_G) was set to 2 seconds. We defined this value to P_G because it represents the minimum interval that the devices should perform the spectrum sensing. Moreover, P_M is higher than the P_G on purpose to verify that the Gateway summarizes properly the information obtained in a smaller interval.

Figure 5.1: Evaluation Scenario



We also configured *ChiMaS* with a history of 3 epochs (l) for both Historical Occupancy Learning and Channel Conditions Learning. The current epoch weight for both α and β was set to 0.5 and the past epochs weights (w), defined from the most recent to the oldest one, are 0.45, 0.35, and 0.2, respectively. These values were chosen to give a higher importance to more recent epochs because it is important to highlight the current state of the channel from the past ones. γ was set to 0.5, while the Channel Conditions Learning weight is $1-\gamma$. We selected this value to consider the same importance for both learning algorithms. The parameters used in the experiments are summarized in Table 5.1.

Table 5.1: Evaluation parameters

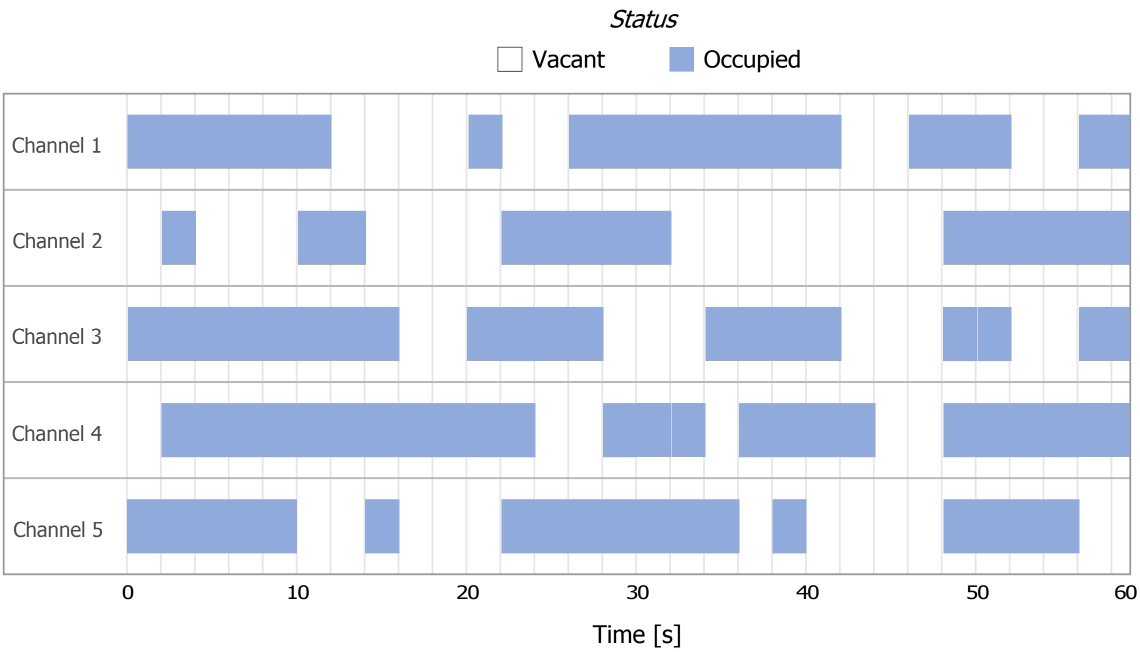
Parameter	**Value**
Execution time per experiment	60 s
USRP sample rate	10^6 samples/s
Number of channels	5
Number of CPEs	5
λ	$[1-5]$ s
Sensing Duration	0.1 s
Sensing Period	$[0.1, 2]$ s
Maximum bandwidth per channel	6 MHz
P_M	30 s
P_G	2 s
Current *ChiMaS* epoch weight (α and β)	0.5
ChiMaS Past epochs (l)	3
ChiMaS Past epochs weights (w)	[0.45, 0.35, 0.2]
ChiMaS Historical Occupancy Learning weight (γ)	0.5

Among the parameters configured in the evaluation, *Kitsune* was used to configure the number of channels, sensing duration, sensing period, P_M, P_G, and the *ChiMaS* current occupancy learning weight (γ). *Kitsune* visualizations of the four cognitive functions are presented in the next subsections.

5.2 Spectrum Sensing Results

The main objective of the spectrum sensing visualization is the improvement of the network administrator knowledge about users behavior in the sensed channels. In this experiment, CPEs are not able to transmit, only periodically performing the spectrum sensing with a period of 2 seconds. All the five CPEs were configured to execute the spectrum sensing function. BS requests the spectrum sensing results from CPEs and determine the current status of the channels occupancy. These results are summarized in a visualization by *Kitsune*, allowing the network administrator to analyze the occupancy status of each channel. The occupancy status of the sensed channels, during a time period of 60 seconds, can be observed in Figure 5.2. In this figure, the y-axis represents each channel sensed, while in the x-axis the occupancy status of each channel over the time can be observed. The colored squares represents an occupied channel, while the white squares indicates the vacancy of the channel. The current status of the channels is obtained using a energy detection algorithm. The spectrum sensing function based in energy detection compares the energy sensed in the channel with a threshold to define the channel status as vacant or occupied.

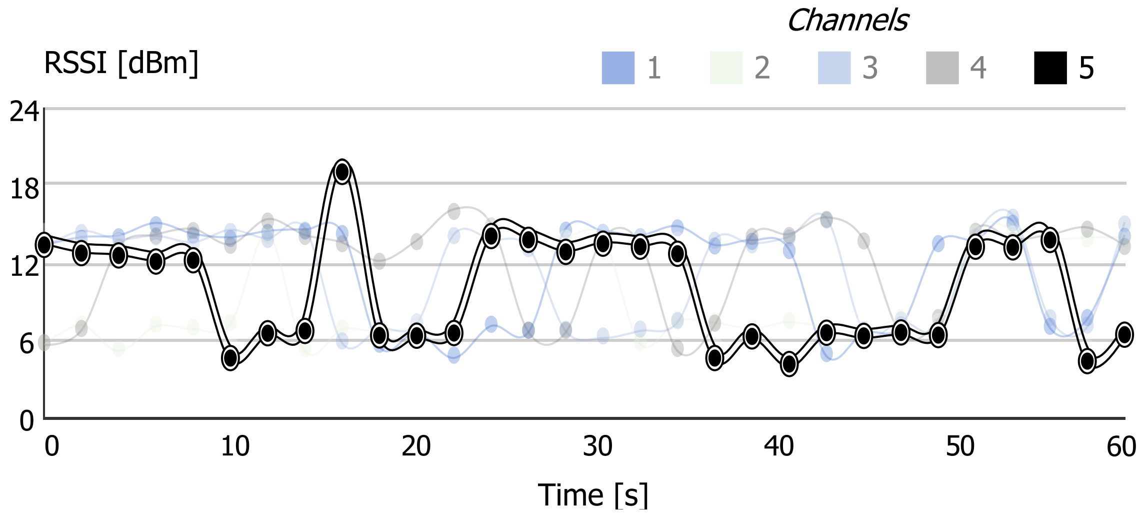
Figure 5.2: Channels occupancy status



Through this visualization, a network administrator may observe the user behavior in the sensed channels and make some considerations. For example, the network administrator may conclude that Channel 4 is a bad option to be selected for transmission, because it has more periods of occupancy when compared to others. It means that a transmissions performed in Channel 4 should be often stopped due to the heavy usage of the channel, decreasing its overall throughput. Different from Channel 4, Channel 2 presents a lower usage, becoming a potential option to be selected for transmissions.

Kitsune also provides a visualization of the RSSI in each sensed channel. This is an important information that can be used to analyze the quality of the sensed channels. This visualization can be seen in Figure 5.3. Associating this visualization with Figure 5.2 it is possible to analyze the sensibility of the spectrum sensing, *i.e.*, the minimum RSSI necessary to consider a channel as occupied. In this visualization, the y-axis represents the RSSI detected in the spectrum sensing of each channel, while the x-axis shows the variation of the RSSI over the time. The network administrator may use this information to adjust the sensing sensibility to a desired level. A vacant channel with high RSSI may indicate a bad channel, with high noise level. On the other hand, an occupied channel with high RSSI may indicate a good channel, with good received signal strength. In addition, the network administrator may highlight the a specific channel information to analyze, as exemplified in Figure 5.3, where the highlighted line represents the RSSI of Channel 5.

Figure 5.3: Channels RSSI

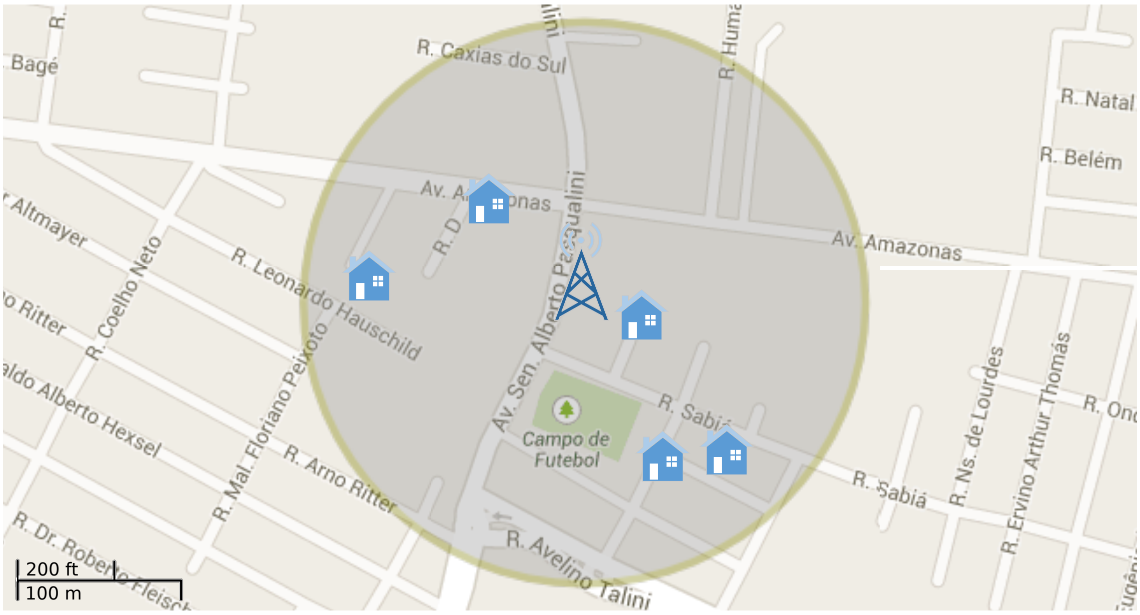


Associating Figure 5.2 and Figure 5.3 we can also observe the spectrum sensing defining Channel 5 as occupied, when the sensed channels present a high RSSI (*e.g.* about 16 dBm). On the other hand, when a low RSSI is observed in the channel (*e.g.* about 6 dBm), it will be defined as vacant. It is a characteristic of the energy detector spectrum sensing technique used in this paper. This information can be used to analyze the proper operation of the spectrum sensing algorithm, adjusting its parameters if necessary.

Location awareness is an essential characteristic of CR devices (CELEBI; ARSLAN, 2007). Considering the importance of this characteristic, *Kitsune* provides the geolocation of BS and CPEs, which may be used by *Kitsune* to display a map of the network. This visualization can be seen in Figure 5.4 In this map, the BS is represented by a tower icon and its coverage area is delimited by the colored circle. Its associated CPEs are represented by house icons.

This visualization can be combined with the channel vacancy status to provide the network administrator an overview about the network. The combined use of these visualizations is a powerful tool to achieve a better understanding about the sensed channels. Is important to highlight that the geolocation visualization is obtained using predefined values of latitude and longitude of BS and CPEs. This functionality can be improved by using mechanisms to automatically obtain the geolocation of the CR devices, such as the Global Positioning System (GPS). However, the CR devices should be able to provide this information, *i.e.*, have a GPS module.

Figure 5.4: Geolocation Map

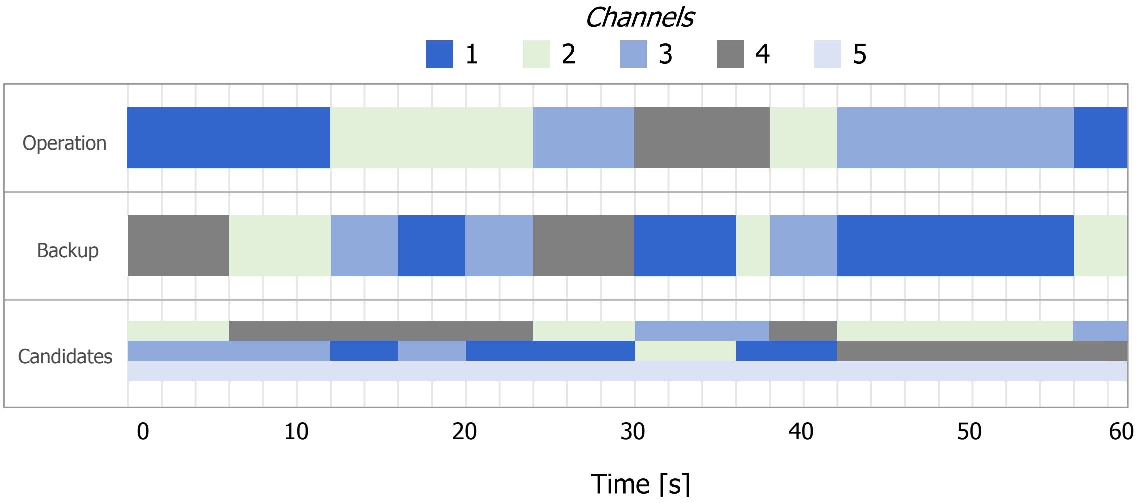


5.3 Spectrum Decision Results

The second experiment is based on the result of the spectrum decision function for each CPE. With the analysis over this visualization, the network administrator may determine which are the channels that are most frequently selected as the operating channel, indicating that they may present more opportunities for transmissions. This information can be used to analyze the operation of the spectrum decision function and change its configuration, if the network administrator deems necessary. In this experiment, *Kitsune* shows the information about the decisions taken by the CPEs and the transmission result over the operating channel. Moreover, with the analysis of the transmissions, administrators can reconfigure the network to maximize the throughput obtained by a specific CPE. Different from the IEEE 802.22 Standard, in this experiment all CPEs are able to perform their own decisions. However, we take advantage of the definition presented by the standard about three classes of channels, enabling CPEs to select the operating, backup and candidate channels. In this experiment, we configured one CPE to operate as a licensed user, using the channels according to a Poisson distribution. The remaining CPEs should classify the channels according to their availability, *i.e.*, when the licensed user was not using the channel. In Figure 5.5 is shown a visualization of the channels analyzed by a specific CPE, classifying in the y-axis the operating, backup, and candidate channels using a color scheme over the time. It is important to remember that despite a CPE selects the operating channel, it may not transmit over this channel in this moment, for example, because there is no data to transmit.

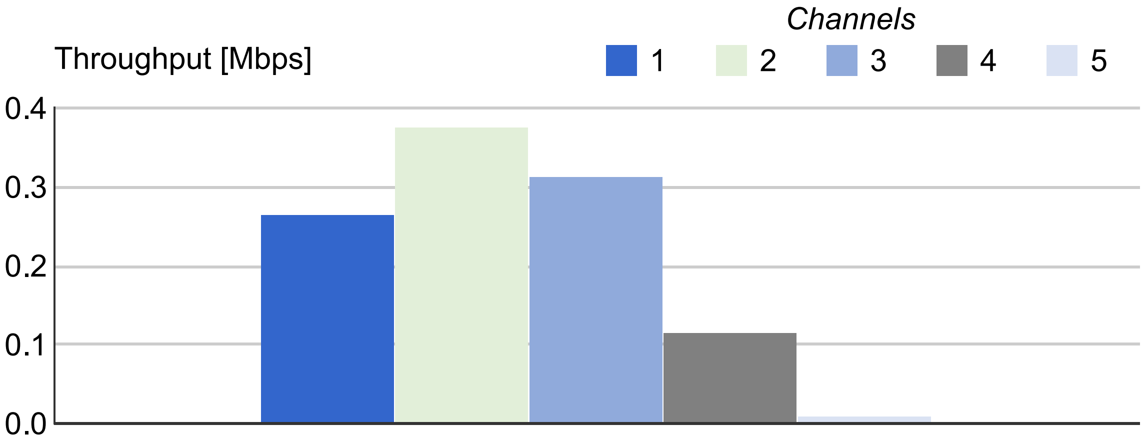
Once verified the status of the channel and selected the operating channel, the CPE may start its transmission over it until a new spectrum sensing is performed. The interval between each spectrum sensing is given by the sensing period. In the case of 5.5, between 30 and 38 seconds, when CPE transmitted, it was performed over the Channel 4, which was the operation channel at that period of time.

Figure 5.5: Channels classification



In Figure 5.6, another important information about the transmissions performed by the CPEs is presented. This information is related to the average uplink throughput obtained in the transmissions. We can associate this throughput directly to the classification shown in Figure 5.5, identifying which is the channel with higher throughout. On the other hand, the worst channel, in terms of throughput can be identified by the network administrator, that may configure the network to ignore this channel, avoiding wasting time by sensing a bad channel for transmission. In Figure 5.6, the y-axis represents the average throughput in megabits per second, while the squares are the average throughput obtained by the channels, identified with different colors.

Figure 5.6: Uplink Throughput



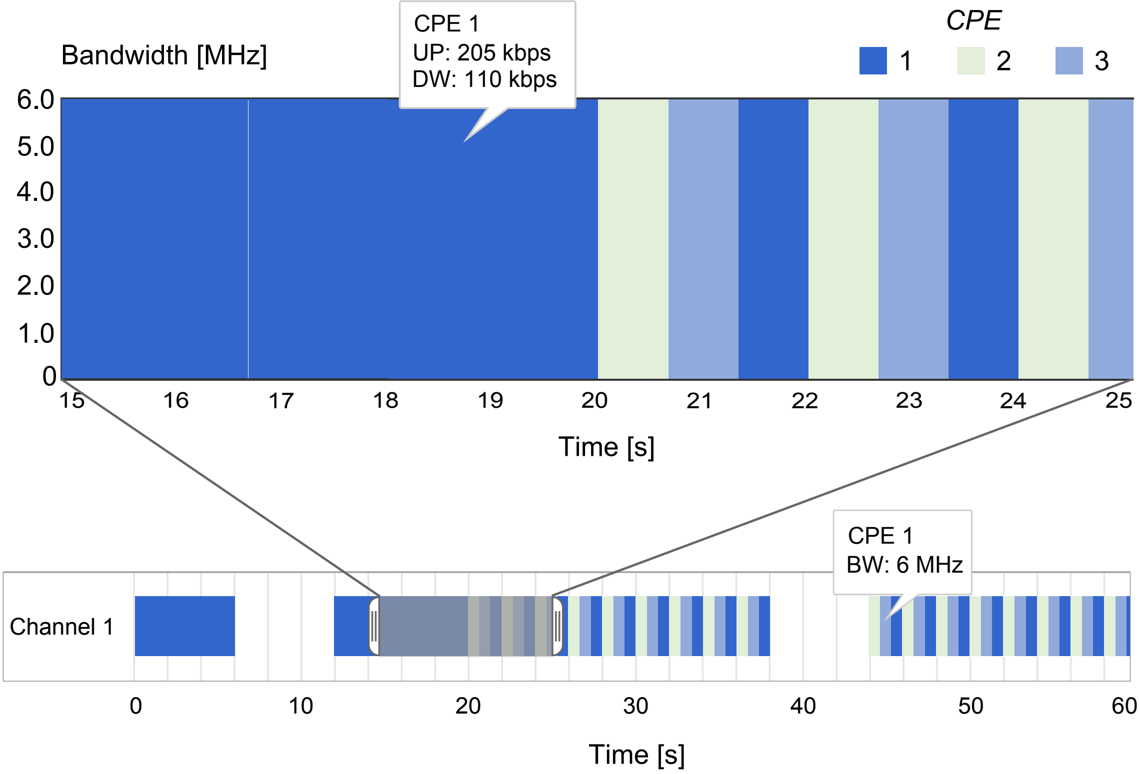
In Figures 5.5 and 5.6, we can observe that the Channel 2 presents higher throughput than the others, being a better choice to be used for transmissions. However, Channel 5 presents the lower average throughput, because it was not selected to be the operating channel during the evaluation. It is important to emphasize that a channel often selected as operating channel not necessarily will present the higher throughput, because CPEs cannot be interested in transmitting all the time. The network administrator can analyze the occupancy, throughput, and the RSSI of channels in the visualizations as a tool to analyze the proper operation of the spectrum decision function, defining which is the best channel and analyze the proper operation of this cognitive function. For example, a channel with 95% of occupancy, 6 dB of RSSI and 30 kbps is a bad channel while a channel with 15% of occupancy, 18 dB of RSSI and 1 Mbps is a good channel. Moreover, a vacant channel with high RSSI may represents a bad channel, since that no signal is detected in the channel, the RSSI represents the noise over the channel. All these considerations can be used also to evaluate the efficacy of the spectrum decision algorithm. If channels with low throughput are often selected as operating channel, the network administrator may consider to change the decision algorithm to a better one.

5.4 Spectrum Sharing Results

The spectrum sharing visualization is a powerful tool, as it provides a very complete analysis of the spectrum sharing status. This analysis provides an overview about the channels division over the time, and can be used to improve the configuration of the spectrum sharing function, for example, increasing the transmission time for each CPE. In this experiment we evaluated the spectrum sharing by the CPEs. The CPEs share the same channel using the Time Division Duplex (TDD) scheme, where the total allocation time of a channel is divided for the CPEs. Moreover, each CPE can request a bandwidth of the operating channel to satisfy its Quality of Service (QoS) requirements. The BS should coordinate the spectrum sharing, providing the division of the vacant channels in the best way. For this evaluation, we defined 3 CPEs to share one channel, considering a bandwidth of 6 MHz to the operating channel. In this experiment, the sharing strategy was emulated, by dividing the total transmission time equally by all the CPEs configured to transmit. However, the sharing strategy may be defined by the management system, according to the network administrator definition. The results of the spectrum sharing are summarized in the visualization illustrated in the Figure 5.7. The channel division for each CPE are identified by different colors along the time. By clicking in a channel in the visualization presented in the Figure 5.7, we can select a specific period of time to analyze in details the sharing status of a specific channel. In this specific visualization, it is possible to see the channel bandwidth allocation in the y-axis.

We can observe that initially the Channel 1 is only used by the CPE 1. It is important to highlight that the squares in the channel represent both the uplink and downlink utilization of the CPE. An important behavior can be observed at the 20 seconds, where other CPEs start sharing the Channel 1 with CPEs 2 and 3. It may occur because the BS operation determines that the Channel 1 can satisfy the bandwidth requested by the CPEs, sharing this channel among the CPEs in order to maintain the bandwidth requested by the CPE 2 and provide the bandwidth requested by the other two CPEs. Each CPE is able to transmit during a specific period of time, determined by BS. In this evaluation, we defined the same period for each CPE, dividing the total transmission time (2 seconds) by all CPEs requesting the channel. When this period finishes, other CPE can start its transmission, until the transmission period finishes. In the case shown in the Figure 5.7, we selected a period of 10 seconds, starting in 15 seconds and finishing in 25, to show the sharing status in the moment where more CPEs starts to share the same channel of the CPE 1, *i.e.*, in 20 seconds. Another important information carried in this visualization is the uplink and downlink throughput average of each CPE that occupies the channel (dialog box). This specific visualization complements the general channels sharing visualization, showing in details the bandwidth allocated for each CPE using the channel and the average throughput obtained by the CPE in its transmissions.

Figure 5.7: Channels sharing visualization

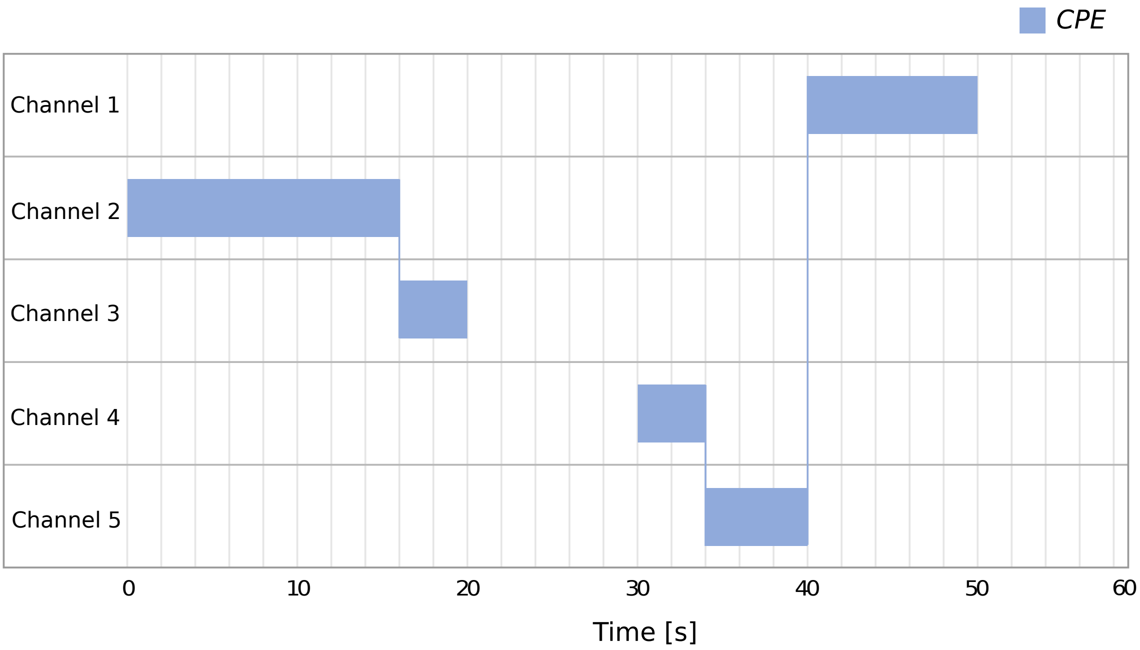


5.5 Spectrum Mobility Results

Through the spectrum mobility visualization, the network administrator can analyze the spectrum handoffs performed during a specific period of time. With this visualization the network administrator can analyze which are the channels most frequently allocated, as well as how often is performed the spectrum handoff. This evaluation considers a CR device configured to perform the spectrum handoff over 5 different channels. The spectrum handoff is only possible through the cognitive function called spectrum mobility. In *Kitsune* prototype, this functions is performed by *ChiMaS* when a different operating channel is selected. When a user is detected in the current operating channel, the CR device should release it and another channel should be selected to continue the transmission. The CR device verifies if a backup channel previously defined is vacant, performing the handoff to the backup channel in positive case. In Figure 5.8, we show the visualization obtained with this evaluation, where the channels are represented the y-axis and the colored rectangles represents the allocation of a given channel.

We can observe that all the configured channels (1, 2, 3, 4, and 5) were used during the evaluation, performing the first handoff at 16 seconds. Another two handoffs were performed, in 34 and 40, respectively. Another important behavior occurs at 20 seconds, where the current transmission is stopped, probably because no vacant channels were found by the spectrum sensing. The transmission restarts at 30 seconds, in the Channel 4. The white squares in the visualization represents the moments where the channel was not allocated, possibly because the channel was defined as occupied by the spectrum sensing function. Therefore, the network administrator may conclude some important facts, such the channels most used and how often no transmissions are performed in each chan-

Figure 5.8: Mobility over the channels



nel, *i.e.*, no channel could be allocated. With these analysis, the network administrator can reconfigure some parameters of the cognitive functions, increasing or decreasing the transmission time over the channels, adjusting the bandwidth allocated in each channel, or removing the channels often occupied from the channels list, for example.

5.6 Configuration Analysis

The last evaluation experiment is also based on the transmissions performed in the CR network. However, in this evaluation we are interested in analyzing the impact of a new configuration settled by the network administrator after analyzing an initial configuration. To perform this evaluation, we stored the results of the spectrum sensing function to apply the same channels occupancy after and before the configuration analysis. Then, we set an initial sensing period and obtained the average throughput by a CPE in each channel. We selected the sensing period to be configured, that is a exclusive parameter of radio networks which must periodically perform the spectrum sensing function. We defined two different configurations for the sensing period: 1 second and 2 seconds. These values are selected to show the impact of the sensing period in the transmissions performed, considering that the maximum interval between spectrum sensing executions is 2 seconds, according to the IEEE 802.22 Standard. As result, we obtained the average throughput, its variation using each sensing period, and the number of transmissions performed. Results were gathered and visualized through *Kitsune* system. Afterwards, a new configuration for the sensing period was set to monitor the results, shown in Table 5.2.

Using both sensing periods, the average throughput obtained for each channel varied. This variation is given by the transmission behavior of the network users, which directly impacts in the results. Moreover, one important conclusion from the analysis of the generated visualizations: the sensing period parameter of a CPE impacts in the average throughput obtained during the transmissions. In the fourth column of the Table 5.2

Table 5.2: Throughput results with different configurations

Channel	Sensing Period [s]	Throughput [Mbps]	Variation [%]	Transmissions
1	1	0.3182	42.04	18
	2	0.5490		25
2	1	0.2267	51.84	15
	2	0.4708		24
3	1	0.4016	18.42	22
	2	0.4923		24
4	1	0.1803	42.54	12
	2	0.3138		17
5	1	0.4027	17.25	21
	2	0.4867		23

is shown the variation of the average throughput obtained in each sensing period. For all the channels, the average throughput increases when the sensing period is higher. This can be explained due to the fact that a CPE must interrupt its transmissions to perform the sense with less frequency, allowing a higher transmission time. An important trade-off must be considered in the adjustment of the sensing period. By increasing the sensing period, the number of spectrum sensing execution decreases. Thus, a licensed user can start its transmission over the current operating channel without the CR device detect it. The CR device will cause interference in the licensed user operation, which is completely prohibited. Thereby, despite algorithms be able to autonomically adjust the sensing period to optimize the throughput, an more detailed analysis may be performed only by the network administrator.

Another important observation that can be highlighted is that the throughput variation is different for each channel. For example, the difference of the average throughput obtained in the Channel 1 was 42.04%, while the Channel 3 presented a variation of 18.42%. To understand these variations, we should observe the number of transmissions performed in each channel. These numbers varies according to the behavior of the users on the channel. For example, in Channel 1, with a sensing period of 1 second, 18 transmissions were performed. Meanwhile, 22 transmissions were performed in Channel 3, with the same sensing period. Increasing this period to 2 seconds, the number of transmissions increased by 7 of the former channel, meanwhile it increased only by 2 the number transmissions of Channel 3. Thus, the number of transmission directly impacts in the average throughput of the CPEs. Therefore, *Kitsune* system is able to provide an overview about throughput variations, their cause, and their consequence, which combined with former scenarios results from previous sections become powerful tools for the network administrator to understand the behavior of the users in a CR network and to participate in all three stages of the continuous knowledge building process.

5.7 Management Tools Comparison

Finally, we perform a comparison about the features of the management tools found in the literature. We summarized these features according to the characteristics of each tool: architecture type, protocol used in the management communication (*i.e.*, to the Management Station obtain the information), and which cognitive functions are managed or proposed by the proposed solution, *i.e.*, spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility. The results of this analysis are summarized in the Table 5.3.

Table 5.3: Management proposals comparison

Proposals	**Architecture**	**Protocol**	**Sensing**	**Decision**	**Sharing**	**Mobility**
CRABSS	Distributed	802.11 MAC	Yes	–	Yes	Yes
Naganawa *et al.*	–	HTTP	Yes	–	–	
Google Spectrum Database	Centralized	–	Yes	–	–	–
WhiteFiService	–	–	Yes	–	–	
CogNet	Centralized	CogNet Bus	–	–	Yes	–
Knowledge Management Toolbox	Distributed (self-management)	–	–	Yes	–	Yes
Kitsune	Hierarchical	SNMP and HTTP	Yes	Yes	Yes	Yes

The CRABSS system (MANFRIN; ZANELLA; ZORZI, 2010) advances in direction to the development of spectrum sensing devices for the Industrial, Scientific and Medical (ISM) 2.4-2.499 GHz band using IEEE 802.22 Medium Access Control (MAC) layer to obtain information about the usage of specific bands in different periods of time. A next steep based on the spectrum sensing is explored in the solution proposed by Naganawa *et al.*, in which heterogeneous devices are used to perform a distributed spectrum sensing. After it, the information is transfered using HTTP to compose a database that enables the visualization of the spectrum usage in specific areas. Based in the advances of the spectrum sensing, the Google Spectrum Database uses the spectrum allocation information provided by the FCC to show the spectrum occupancy over the USA territory. The WhiteFiService, in its turn, provides an API for spectrum sensing implementation with a specific hardware device.

The CogNet (MANOJ; RAO; ZORZI, 2008) presents a centralized knowledge management architecture for CR networks, collecting the results of learning algorithms through a special protocol called CogNet Bus. These results are used to compose a shared knowledge database that can be used by the cognitive functions to improve the RF spectrum utilization. In the same way, the Knowledge Management Toolbox (STAVROULAKI et al., 2012) refers to a decentralized knowledge management architecture that enables the usage of policies by CR devices to take the decisions over the channels, performing the spectrum mobility when necessary. These policies are used by the learning algorithms to improve the RF spectrum usage by the CR devices. Finally, we feature the *Kitsune* system as a hierarchical management system, able to manage the CR networks by configuring, monitoring, and visualizing the cognitive functions performed by the CR devices. It is important to highlight that *Kitsune* does not implement the cognitive functions, but functions *Kitsune* is able to monitor, configure, and visualize all the four cognitive functions. We characterize *Kitsune* system as the most complete management system for CR networks, due to its integration with all the cognitive functions performed by the CR devices.

Once proved the proper operation of the *Kitsune* system and explored its advantages when compared with the main solutions found in the literature, in the next section we present the conclusions obtained through the development of this dissertation and discuss future work related to the improvement of *Kitsune*.

6 CONCLUSION AND FUTURE WORK

In this dissertation, we presented the problem of the spectrum scarcity and underutilization in today wireless communication systems and networks. The increasing usage of the RF spectrum associated with the actual policy to explore this resource are forcing the creation of new solutions to better explore the RF spectrum resource to perform communication. A widely explored solution today is the cognitive radio concept, which aims to explore unused channels of the RF spectrum and opportunistically perform transmission, introducing cognitive functions to the radio devices.

These cognitive functions enable CR devices to analyze the RF spectrum looking for opportunities to transmit. Based on this analysis, the CR devices decide what are the best available channels to be used for transmission. Moreover, these devices can share the selected channels with other devices and leave a channel when the user licensed to transmit in the channel is detected performing its transmission. The cognitive functions allow an improved spectrum management by the devices that compose a cognitive radio network, which may reduce the spectrum underutilization problem. Despite many research efforts be concerned in the best ways to perform each one of the cognitive functions in the last years, few researches were conducted to provide the management of CR networks focused on the configuration, monitoring, and visualization of the cognitive functions.

In this Master dissertation it was proposed a new management system for cognitive radio networks called *Kitsune*. The proposed system is based on the four cognitive functions, allowing the configuration of their parameters, monitoring the execution of the cognitive functions in each radio device, and making the results visualizations for the network administrator analysis. Therefore, *Kitsune* allows administrators to perform in a continuous knowledge building process considering the cognitive functions results with the use of an intuitive user interface.

6.1 Main Contributions and Results Obtained

Along with the proposed management system this dissertation contributes with a prototype implementation of *Kitsune*, showing the viability in the deployment of the proposed system on cognitive radio networks. Moreover, an extension to the IEEE 802.22 MIB is other important contribution of *Kitsune*. This extended MIB can be used to the management of different cognitive functions that are not mapped by the IEEE 802.22 Standard, such as the spectrum decision function being performed by CPEs, and the spectrum sharing function. Thus, this extension enable the management of another types of CR networks by *Kitsune*.

Another important contribution of this dissertation is the proposal and implementation of a channel list management system called *ChiMaS*, used to prove the management of the spectrum decision, spectrum sharing and spectrum mobility functions by *Kitsune*. This system can be used in different types of CR networks, including those where the CR devices have autonomy to perform the spectrum decision locally and select the best channel for transmission. *ChiMaS* is able to define an operating, a backup, and the candidate channels. Such system is directly managed by *Kitsune*, showing the operation of the proposed management system.

Kitsune also provides a set of visualizations of the cognitive functions results that is the last step of a continuous knowledge building process for the network administrator. The configuration, monitoring, and visualization of the cognitive functions allows the network administrator to achieve a new level of knowledge about the CR devices operation in the network. The visualization was designed to ease the analysis of the cognitive functions results by the network administrator, showing relevant information simply.

Kitsune prototype was evaluated in an experimental RF scenario and the results shown that the proposed system allows network administrators to improve the throughput obtained in transmissions performed by CPEs. This is possible by gathering results, applying visualizations techniques, and finally, changing the configurations of these devices. *Kitsune* enables administrators to interact actively in the configuration, monitoring and visualizations of the cognitive function. Such interaction allows *Kitsune* to readapt a CR network to achieve a better average throughput.

6.2 Final Remarks and Future Work

There are several trends for future research. Once proved the proper operation of *Kitsune*, we intend to further extend *Kitsune* functionalities, allowing the configuration, monitoring and visualization of the cognitive functions in different network architectures, such as CR mesh and ad-hoc networks. To enable it, *Kitsune* can be modified to implement some concepts of the management by delegation model.

Another improvement to *Kitsune* can be performed in terms of functionalities, using new mechanisms that improve the network operation. In this case, different algorithms designed for each cognitive function may be managed through *Kitsune*. For example, changing the spectrum decision criteria or using different spectrum decision algorithms, based on the current network status. Moreover, *Kitsune* can be extended to comprise a set of algorithms and select the best one depending on the network performance. Changing the spectrum sensing algorithm, for example, it is possible to improve the accuracy in the detection of licensed users, or decrease the time expended with this function. Furthermore, the use of policies in the configuration of CR devices should be considered. The usage of policies may facilitate the configuration of the CR devices by the network administrator, providing some level of abstraction and ensuring that restrictions will be respected.

About the *Kitsune* interface in the Manager component, the feedback of expert network administrators may be obtained. This information may be used to improve the *Kitsune* interface, turning it more intuitive and useful to network administrators. The visualizations present in the *Kitsune* interface can be also extended in the future, by making different visualization for the same cognitive functions, according to the necessities of the network administrators. We consider that *Kitsune* may be an important alternative to spectrum visualizations tool, such as the Google Spectrum Database, once that this tool use static database with information about the spectrum allocation, while *Kitsune* uses more precise information about the spectrum utilization.

REFERENCES

AKYILDIZ, I. F.; LEE, W.-Y.; CHOWDHURY, K. R. CRAHNs: cognitive radio ad hoc networks. **Ad Hoc Networks**, Amsterdam, The Netherlands, The Netherlands, v.7, n.5, p.810–836, July 2009.

AKYILDIZ, I. F.; LEE, W.-Y.; VURAN, M. C.; MOHANTY, S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. **Computer Networks**, [S.l.], v.50, n.13, p.2127–2159, 2006.

AKYILDIZ, I.; LEE, W.-Y.; VURAN, M.; MOHANTY, S. A survey on spectrum management in cognitive radio networks. **IEEE Communications Magazine**, [S.l.], p.40–48, 2008.

BIAN, K.; PARK, J.-M.; DU, X.; LI, X. Enabling fair spectrum sharing: mitigating selfish misbehaviors in spectrum contention. **IEEE Network**, [S.l.], v.27, n.3, p.16–21, 2013.

BICEN, A.; PEHLIVANOGLU, E.; GALMES, S.; AKAN, O. Dedicated Radio Utilization for Spectrum Handoff and Efficiency in Cognitive Radio Networks. **IEEE Transactions on Wireless Communications**, [S.l.], v.PP, n.99, p.1–9, 2013.

BKASSINY, M.; LI, Y.; JAYAWEERA, S. A Survey on Machine-Learning Techniques in Cognitive Radios. **IEEE Communications Surveys Tutorials**, [S.l.], n.99, p.1–24, 2012.

BOX, D.; EHNEBUSKE, D.; KAKIVAYA, G.; LAYMAN, A.; MENDELSOHN, N.; NIELSEN, H.; THATTE, S.; WINER, D. Simple object access protocol (SOAP) 1.1. **World Wide Web Consortium (W3C)**, [S.l.: s.n.], 2000.

CELEBI, H.; ARSLAN, H. Utilization of Location Information in Cognitive Wireless Networks. **IEEE Wireless Communications**, [S.l.: s.n.], p.6–13, 2007.

CHEN, T.; ZHANG, H.; MAGGIO, G. M.; CHLAMTAC, I. CogMesh: a cluster-based cognitive radio network. In: IEEE INTERNATIONAL SYMPOSIUM ON NEW FRONTIERS IN DYNAMIC SPECTRUM ACCESS NETWORKS (DYSPAN), 2007. **Proceedings...** [S.l.: s.n.], 2007. p.168–178.

CHRISTIAN, I.; MOH, S.; CHUNG, I.; LEE, J. Spectrum mobility in cognitive radio networks. **IEEE Communications Magazine**, [S.l.], v.50, n.6, p.114–121, 2012.

CLANCY, C.; HECKER, J.; STUNTEBECK, E.; O'SHEA, T. Applications of Machine Learning to Cognitive Radio Networks. **IEEE Wireless Communications**, [S.l.: s.n.], v.14, n.4, p.47–52, 2007.

CORDEIRO, C.; CHALLAPALI, K.; BIRRU, D.; SAI SHANKAR, N. IEEE 802.22: the first worldwide wireless standard based on cognitive radios. In: NEW FRONTIERS IN DYNAMIC SPECTRUM ACCESS NETWORKS (DYSPAN), 2005. **Proceedings...** [S.l.: s.n.], 2005. p.328–337.

DARDAILLON, M.; MARQUET, K.; RISSET, T.; SCHERRER, A. Software defined radio architecture survey for cognitive testbeds. In: INTERNATIONAL WIRELESS COMMUNICATIONS AND MOBILE COMPUTING CONFERENCE (IWCMC), 2012. **Proceedings...** [S.l.: s.n.], 2012. p.189–194.

DE DOMENICO, A.; CALVANESE STRINATI, E.; DI BENEDETTO, M.-G. A Survey on MAC Strategies for Cognitive Radio Networks. **IEEE Communications Surveys & Tutorials**, [S.l.: s.n.], v.14, n.1, p.21–44, 2012.

FAGANELLO, L.; KUNST, R.; BOTH, C.; GRANVILLE, L.; ROCHOL, J. Improving reinforcement learning algorithms for dynamic spectrum allocation in cognitive sensor networks. In: IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (WCNC), 2013. **Proceedings...** [S.l.: s.n.], 2013. p.35–40.

FIELDING, R. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado em Ciência da Computação) — University of California.

GHOSH, C.; PAGADARAI, S.; AGRAWAL, D.; WYGLINSKI, A. A framework for statistical wireless spectrum occupancy modeling. **IEEE Transactions on Wireless Communications**, [S.l.: s.n.], v.9, p.38–44, 2010.

IEEE. Standard for Information Technology - Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN) - Specific requirements Part 22. **IEEE Std 802.22**, [S.l.: s.n.], p.1–680, 2011.

LEINWAND, A.; CONROY, K. **Network management**: a practical perspective. [S.l.]: Addison-Wesley, 1996. (UNIX and open systems series).

MANFRIN, R.; ZANELLA, A.; ZORZI, M. CRABSS: calradio-based advanced spectrum scanner for cognitive networks. **Wireless Communication & Mobile Computing**, Chichester, UK, v.10, n.12, p.1682–1695, 2010.

MANOJ, B. S.; RAO, R.; ZORZI, M. CogNet: a cognitive complete knowledge network system. **IEEE Wireless Communications**, [S.l.: s.n.], v.15, n.6, p.81–88, 2008.

MAROTTA, M. A.; CARBONE, F. J.; SANTANNA, J. J. C.; TAROUCO, L. M. R. Through the Internet of Things - a Management by Delegation Smart Object Aware System (MbDSAS). In: COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), 2013. **Proceedings...** [S.l.: s.n.], 2013. p.732–741.

MARTIN-FLATIN, J.-P.; ZNATY, S.; HUBAUX, J.-P. A Survey of Distributed Enterprise Network and Systems Management Paradigms. **Journal of Network and Systems Management**, [S.l.: s.n.], v.7, p.1–11, 1999.

MASONTA, M. T.; MZYECE, M.; NTLATLAPA, N. Spectrum Decision in Cognitive Radio Networks: a survey. **IEEE Communications Surveys & Tutorials**, [S.l.: s.n.], v.15, n.3, p.1088–1107, 2013.

MENDES, A.; AUGUSTO, C.; SILVA, M. da; GUEDES, R.; REZENDE, J. de. Channel sensing order for cognitive radio networks using reinforcement learning. In: IEEE CONFERENCE LOCAL COMPUTER NETWORKS (LCN), 2011. **Proceedings...** [S.l.: s.n.], 2011. p.546–553.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. 432p. v.4.

MITOLA J., I.; MAGUIRE G.Q., J. Cognitive radio: making software radios more personal. **IEEE Personal Communications**, [S.l.: s.n.], v.6, p.13–18, 1999.

MOORE, E. B. Policy Core Information Model (PCIM) Extensions. In: **Request For Comments (RFC) 3460**. [S.l.]: IETF, 2003.

MOURA, G.; SILVESTRIN, G.; SANCHEZ, R.; GASPARY, L.; GRANVILLE, L. On the Performance of Web Services Management Standards - An Evaluation of MUWS and WS-Management for Network Management. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT (IM), 2007. **Proceedings...** [S.l.: s.n.], 2007. p.459–468.

NAGANAWA, J.; KIM, H.; SARUWATARI, S.; ONAGA, H.; MORIKAWA, H. Distributed spectrum sensing utilizing heterogeneous wireless devices and measurement equipment. In: IEEE SYMPOSIUM ON NEW FRONTIERS IN DYNAMIC SPECTRUM ACCESS NETWORKS (DYSPAN), 2011. **Proceedings...** [S.l.: s.n.], 2011. p.173–184.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. "big"’ web services: making the right architectural decision. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2010. **Proceedings...** New York: [s.n], 2010. p.805–814.

PEHA, J. Sharing Spectrum Through Spectrum Policy Reform and Cognitive Radio. **Proceedings of the IEEE, Special Issue in Cognitive Radios**, [S.l.: s.n.], v.97, n.4, p.708–719, 2009.

POTIER, P.; QIAN, L. Network management of cognitive radio ad hoc networks. In: INTERNATIONAL CONFERENCE ON COGNITIVE RADIO AND ADVANCED SPECTRUM MANAGEMENT (COGART), 2011, 2011. **Proceedings...** New York: [s.n], 2011. p.1–5.

QIN, H.; CUI, Y. Spectrum Coordination Protocol for reconfiguration management in cognitive radio network. In: IEEE YOUTH CONFERENCE ON INFORMATION, COMPUTING AND TELECOMMUNICATION, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.407–410.

SCHONWALDER, J.; QUITTEK, J.; KAPPLER, C. Building distributed management applications with the IETF Script MIB. **IEEE Journal on Selected Areas in Communications**, [S.l.: s.n.], v.18, n.5, p.702–714, 2000.

SHERMAN, M.; COMBA, A.; HE, D.; MCDONALD, H. A cognitive policy management framework for DoD. In: MILITARY COMMUNICATIONS CONFERENCE (MILCOM), 2010. **Proceedings...** [S.l.: s.n.], 2010. p.1875–1880.

SLOMAN, M. Policy Driven Management For Distributed Systems. **Journal of Network and Systems Management**, [S.l.: s.n.], v.2, p.333–360, 1994.

STAVROULAKI, V.; BANTOUNA, A.; KRITIKOU, Y.; TSAGKARIS, K.; DEMESTICHAS, P.; BLASCO, P.; BADER, F.; DOHLER, M.; DENKOVSKI, D.; ATANASOVSKI, V.; GAVRILOVSKA, L.; MOESSNER, K. Knowledge Management Toolbox: machine learning for cognitive radio networks. **IEEE Vehicular Technology Magazine**, [S.l.: s.n.], p.91–99, june 2012.

STOTAS, S.; NALLANATHAN, A. Enhancing the Capacity of Spectrum Sharing Cognitive Radio Networks. **IEEE Transactions on Vehicular Technology**, [S.l.: s.n.], v.60, n.8, p.3768–3779, 2011.

THOMAS, R. W.; FRIEND, D. H.; DASILVA, L. A.; MACKENZIE, A. B. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. **IEEE Communications Magazine**, [S.l.: s.n.], v.44, n.12, p.51–57, 2006.

TRAGOS, E.; ZEADALLY, S.; FRAGKIADAKIS, A.; SIRIS, V. Spectrum Assignment in Cognitive Radio Networks: a comprehensive survey. **IEEE Communications Surveys Tutorials**, [S.l.: s.n.], v.15, n.3, p.1108–1135, 2013.

VANDERHORN, N.; HAAN, B.; CARVALHO, M.; PEREZ, C. Distributed policy learning for the Cognitive Network Management System. In: MILITARY COMMUNICATIONS CONFERENCE (MILCOM), 2010. **Proceedings...** [S.l.: s.n.], 2010. p.435–440.

WANG, C.-X.; CHEN, H.-H.; HONG, X.; GUIZANI, M. Cognitive radio network management. **IEEE Vehicular Technology Magazine**, [S.l.: s.n.], v.3, n.1, p.28–35, 2008.

YUCEK, T.; ARSLAN, H. A survey of spectrum sensing algorithms for cognitive radio applications. **IEEE Communications Surveys Tutorials**, [S.l.: s.n.], v.11, p.116–130, 2009.

ZUBAIR, S.; FISAL, N.; ABAZEED, M.; SALIHU, B. A proposed network management protocol for Cognitive Radio Sensor Networks. In: INTERNATIONAL SYMPOSIUM ON TELECOMMUNICATION TECHNOLOGIES (ISTT), 2012. **Proceedings...** [S.l.: s.n.], 2012. p.88–93.

APPENDIX A SMI DESCRIPTION OF THE MIB

This appendix refers to the SMI description of the MIB used in the implementation of *Kitsune* prototype.

- **File name:**
  MIB-KITSUNE.smi
- **Size:**
  9.4 kbits
- **Last modification:**
  January, 21, 2014

```
MIB-KITSUNE DEFINITIONS ::= BEGIN
IMPORTS
MODULE-IDENTITY, OBJECT-TYPE, Integer32, mib-2 FROM SNMPv2-SMI;

mibIEEE80222 MODULE-IDENTITY
LAST-UPDATED "200602020000Z"
ORGANIZATION "UFRGS"
CONTACT-INFO "lbondan@inf.ufrgs.br"
DESCRIPTION "An extended implementation of IEEE 802.22 MIB for cognitive radio networks"
::= { mib-2 1300 }

-- ********************************* DEV *********************************
wranDevMib OBJECT IDENTIFIER ::= { mibIEEE80222 1 }
-- ********************************* BS **********************************
wranIfBsMib OBJECT IDENTIFIER ::= { mibIEEE80222 2 }
-- **************************** BS Management ****************************
wranIfBsSfMgmt OBJECT IDENTIFIER ::= { mibIEEE80222 3 }
-- ********************************* CPE *********************************
wranIfCpeMib OBJECT IDENTIFIER ::= { mibIEEE80222 4 }
-- ********************************* SM **********************************
wranIfSmMib OBJECT IDENTIFIER ::= { mibIEEE80222 5 }

wranIfSmConfigTable OBJECT-TYPE
SYNTAX SEQUENCE OF WranIfSmConfigTableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object contains information about the SM configuration.
It is made of multiple entries, one for each CPE, as defined by wranIfSmConfigTableEntry."
::= {wranIfSmMib 1}

wranIfSmConfigTableEntry OBJECT-TYPE
SYNTAX WranIfSmConfigTableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object represents the entry in wranIfSmConfigTableTable."
INDEX {wranIfSmSsaChAvailabilityCheckTime, wranIfSmSsaNonOccupancyPeriod,
wranIfSmSsaChannelOpeningTxTime}
::= {wranIfSsaSensingRecordTable 1}

WranIfSmConfigTableEntry::=
SEQUENCE{
wranIfSmSsaChAvailabilityCheckTime OCTET STRING,
wranIfSmSsaNonOccupancyPeriod OCTET STRING,
wranIfSmSsaChannelOpeningTxTime OCTET STRING,
}

wranIfSmSsaChAvailabilityCheckTime OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS current
DESCRIPTION "Sensing window to check if a channel is vacant or occupied."
::= {wranIfSmConfigTableEntry 1}

wranIfSmSsaNonOccupancyPeriod OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS current
DESCRIPTION "Period in which the CR cannot perform transmissions."
::= {wranIfSmConfigTableEntry 2}

wranIfSmSsaChannelOpeningTxTime OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS current
DESCRIPTION "Maximum transmission time"
::= {wranIfSmConfigTableEntry 3}

wranIfSmChClassificationStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF WranIfSmConfigTableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object contains information about the SM configuration.
It is made of multiple entries, one for each CPE, as defined by
```

```
wranIfSmChClassificationStatusEntry."
::= {wranIfSmMib 2}

wranIfSmChClassificationStatusEntry OBJECT-TYPE
SYNTAX WranIfSmConfigTableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object represents the entry in wranIfSmChClassificationStatusTable."
INDEX {wranIfSmManagedChannel, wranIfSmManagedChannelStatus,
wranIfSmWranOccupiedChannelSet}
::= {wranIfSmChClassificationStatusTable 1}

wranIfSmChClassificationStatusEntry::=
SEQUENCE{
wranIfSmManagedChannel OCTET STRING,
wranIfSmManagedChannelStatus OCTET STRING,
wranIfSmWranOccupiedChannelSet OCTET STRING,
}

wranIfSmManagedChannel OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS current
DESCRIPTION "Managed channel identification."
::= {wranIfSmChClassificationStatusEntry 1}

wranIfSmManagedChannelStatus OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS current
DESCRIPTION "Status of the managed channels that can be used in the CR network."
::= {wranIfSmChClassificationStatusEntry 2}

wranIfSmWranOccupiedChannelSet OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS current
DESCRIPTION "Channels evaluated as occupied during the spectrum sensing."
::= {wranIfSmChClassificationStatusEntry 3}

-- ******************************** SSA ********************************
wranIfSsaMib OBJECT IDENTIFIER ::= { mibIEEE80222 6 }

wranIfSsaSensingRecordTable OBJECT-TYPE
SYNTAX SEQUENCE OF WranIfSsaSensingRecordEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object contains information the sensing status of each channel.
It is made of multiple entries, one for each channel, as defined by
wranIfSsaSensingRecordEntry."
::= {wranIfSsaMib 1}

wranIfSsaSensingRecordEntry OBJECT-TYPE
SYNTAX WranIfSsaSensingRecordEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object represents the entry in wranIfSsaSensingRecordTable."
INDEX {wranIfSsaSensingChannel, wranIfSsaTimeLastSensing, wranIfSsaSensingPathRssi}
::= {wranIfSsaSensingRecordTable 1}

wranIfSsaSensingRecordEntry::=
SEQUENCE{
wranIfSsaSensingChannel OCTET STRING,
wranIfSsaTimeLastSensing OCTET STRING,
wranIfSsaSensingPathRssi OCTET STRING,
}

wranIfSsaSensingChannel OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS current
DESCRIPTION "Channel that sensing has been conducted on, or not
conducted on if in IPC-UPD."
```

```
::= {wranIfSsaSensingRecordEntry 1}

wranIfSsaTimeLastSensing OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS current
DESCRIPTION "Last time that this channel was sensed."
::= {wranIfSsaSensingRecordEntry 2}

wranIfSsaSensingPathRssi OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS current
DESCRIPTION "RSSI on sensing path."
::= {wranIfSsaSensingRecordEntry 3}

-- ********************************* Database *********************************
wranIfDatabaseServiceMib OBJECT IDENTIFIER ::= { mibIEEE80222 7 }

-- *************************** Decision and Sharing ***************************
ExtendedMib OBJECT IDENTIFIER ::= { mibIEEE80222 8 }

DecisionStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF DecisionStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Objects that refers to the spectrum decision function, as defined by
DecisionStatusEntry."
::= {ExtendedMib 1}

DecisionStatusEntry OBJECT-TYPE
SYNTAX DecisionStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object represents the entry in DecisionStatusTable."
INDEX {DecisionOperatingChannel, DecisionBackupChannel, DecisionCandidateChannels,
DecisionGamaWeight, DecisionRssiMinValue, DecisionRssiMaxValue, DecisionUplinkThroughput,
DecisionDownlinkThroughput}
::= {DecisionStatusTable 1}

DecisionStatusEntry::=
SEQUENCE{
DecisionOperatingChannel OCTET STRING,
DecisionBackupChannel OCTET STRING,
DecisionCandidateChannels OCTET STRING,
DecisionGamaWeight OCTET STRING,
DecisionRssiMinValue OCTET STRING,
DecisionRssiMaxValue OCTET STRING,
DecisionUplinkThroughput OCTET STRING,
DecisionDownlinkThroughput OCTET STRING,
}

DecisionOperatingChannel OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Operating Channel selected by the Decision Function"
    ::= {DecisionStatusEntry 1}

DecisionBackupChannel OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Backup Channel selected by the Decision Function"
    ::= {DecisionStatusEntry 2}

DecisionCandidateChannels OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Candidate Channels selected by the Decision Function"
    ::= {DecisionStatusEntry 3}
```

```
DecisionGamaWeight OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "Gamma Weight to be used by the Decision Function"
    ::= {DecisionStatusEntry 4}

DecisionRssiMinValue OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "Minimum RSSI level to the RSSI corresponding table"
    ::= {DecisionStatusEntry 5}

DecisionRssiMaxValue OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "Maximum RSSI level to the RSSI corresponding table"
    ::= {DecisionStatusEntry 6}

DecisionUplinkThroughput OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The max uplink Throughput obtained in Mbits per second"
::= { DecisionStatusEntry 7}

DecisionDownlinkThroughput OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The max downlink Throughput obtained in Mbits per second"
::= { DecisionStatusEntry 8}

SharingStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF SharingStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Objects that refers to the spectrum sharing function, as defined by
SharingStatusEntry."
::= {ExtendedMib 2}

SharingStatusEntry OBJECT-TYPE
SYNTAX SharingStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object represents the entry in SharingStatusTable."
INDEX {SharingStartTime, SharingStopTime, SharingAllocatedBand}
::= {SharingStatusTable 1}

SharingStatusEntry::=
SEQUENCE{
SharingStartTime OCTET STRING,
SharingStopTime OCTET STRING,
SharingAllocatedBand OCTET STRING,
}

SharingStartTime OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Timestamp when the CPE started the sharing over the operating channel."
    ::= {SharingStatusEntry 1}

SharingStopTime OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Timestamp when the CPE stoped the sharing over the operating channel."
    ::= {SharingStatusEntry 2}
```

```
SharingAllocatedBand OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "Channel band allocated to transmit in the last transmission."
    ::= {SharingStatusEntry 3}
END
```

APPENDIX B ACCEPTED PAPER – NOMS 2014

In this paper we published advances in the management of CR networks, proposing the configuration, monitoring, and visualization of the spectrum sensing function using *Kitsune* system.

- **Title:**
  Kitsune: A Management System for Cognitive Radio Networks Based on Spectrum Sensing
- **Conference:**
  Mini-conference of IEEE/IFIP Networks and Operation Management Symposium (NOMS 2014)
- **URL:**
  `http://noms2014.ieee-noms.org/`
- **Date:**
  May 5-9, 2014
- **Held at:**
  Radisson Park Inn, Krakow, Poland
- **Digital Object Identifier (DOI):**
  To appear

Kitsune: A Management System for Cognitive Radio Networks Based on Spectrum Sensing

Lucas Bondan, Marcelo Antonio Marotta, Maicon Kist, Leonardo Roveda Faganello,
Cristiano Bonato Both, Juergen Rochol, Lisandro Zambenedetti Granville
Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Porto Alegre, Brazil
Email: {lbondan, mamarotta, maicon.kist, lrfaganello, cbboth, juergen, granville}@inf.ufrgs.br

Abstract—**Software defined radio enables the improvement of the radio-frequency spectrum utilization through the design of cognitive radio devices. The implementation of these devices must be based on spectrum sensing function searching for vacant channels and, opportunistically, transmit over these channels in a cognitive radio network. Therefore, the configuration, monitoring and visualization of the spectrum sensing function are fundamentals to the continuous learning process of the network administrator. In this paper we propose *Kitsune*, a management system based on a hierarchical model allowing to manage summarized information about the spectrum sensing function in a cognitive radio networks. Moreover, a *Kitsune* prototype was developed and evaluated through a real IEEE 802.22 scenario using TV channels to Internet access. Results shown that *Kitsune* allows network administrator to achieve a higher knowledge about behavior of the users and improve the average throughput for each channel.**

Index Terms—**software defined radio, cognitive radio, network management, spectrum sensing**

I. INTRODUCTION

Cognitive Radio (CR) is an emerging technology, based on Software Defined Radio (SDR), that enables the improvement of the Radio-Frequency (RF) spectrum usage [1]. CR devices can analyze the RF spectrum, searching for channels that are not being used and opportunistically transmit over these channels. To allow opportunistic transmissions, a CR device needs to implement four main functions: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility [2]. Among these functions, the spectrum sensing is considered the most important because it is responsible for analyzing the RF spectrum and evaluate whether or not a channel is vacant [3]. Moreover, the behavior of users directly impact in the spectrum sensing evaluation, since these users may transmit in the sensed channel.

The spectrum sensing function must be implemented in each device of a CR network. As defined by the IEEE 802.22 standard, a CR network is composed of two types of devices: Base Stations (BS) and Customer-Premises Equipment (CPE) [4]. In these networks, the spectrum sensing function can be dynamically configured to minimize the sensing time and maximizing both the transmission time and the reliability of spectrum sensing results. This configuration contains a set of parameters, such as the number of sensed channels, sensing time, and sensing window [5]. The dynamic configuration of these parameters can be improved through a management system. In such system the spectrum sensing function can be monitored and visualized, allowing the administrator to continuously learn how to properly configure the spectrum sensing function of each CR to obtain the best network performance.

In the last decade, much has been discussed about CR devices and networks [6], but there are few discussions about the management of CR networks. This lack of discussions may be due to the fact that the deployment of CR networks has begun only in the last years [7]. Therefore, CR network management has the same classical problem of traditional networks management, *i.e.*, when a new technology is introduced, the management is usually neglected, until eventually it becomes necessary. Currently, researches on CR networks management mainly addressed the usage of management protocols [8], autonomic systems [9], policies [10], spectrum visualization [11], and machine learning algorithms [2]. However, to the best of our knowledge, no management system for CR networks based on the spectrum sensing function has been proposed. Such management system should offer summarized information about the spectrum sensing function, retrieved from each CR device, allowing the administrators to acquire better knowledge about the managed CR network.

In this paper we propose *Kitsune*, a management system for CR networks designed to enable the administrator to continuously learn how to better configure the spectrum sensing function. Management through *Kitsune* is separated into three stages: configuration, monitoring, and visualization. *Kitsune* system follows a hierarchical management model, composed of a Manager, Gateways, and Agents. The main advantage of the proposed system is to enable dynamic configuration and monitoring through an intuitive visualization interface. The *Kitsune* management system is evaluated using a prototype implemented in a experimental scenario based on the IEEE 802.22 standard. We demonstrated that the proposed system enables the network administrator to achieve a better knowledge about the network, improving the throughput obtained in the transmissions performed by the CPEs.

The rest of this paper is organized as follows. In Section II, we present a background on CR network management and

978-1-4799-0913-1/14/$31.00 © 2014 IEEE

related work in the area. In the Section III, the *Kitsune* management system is explained in details. In the Section IV, the evaluation methodology is explained and results obtained are discussed. Finally, conclusions and future work are provided in Section V.

II. BACKGROUND ON COGNITIVE RADIO MANAGEMENT

In this section, we present a brief background and related work on CR and its management. In the Subsection II-A, the four main cognitive functions, the traditional hierarchical management model, and the Management Information Base (MIB) of the IEEE 802.22 standard are presented. Afterwards, the related work on CR networks management is discussed in the Subsection II-B.

A. Cognitive Radio and Network Management

CR devices are foreseen as the future of radio devices, greatly improving the utilization of the RF spectrum. In a CR, almost every aspect of its operation can be dynamically configured [1]. The most important characteristic of the CR is the ability to search the RF spectrum for vacant channels and learn about the RF environment to opportunistically perform transmission over these channels. The literature in CR separates the functionality of these devices in four functions: spectrum sensing, spectrum decision, spectrum sharing and spectrum mobility [12]. Through the spectrum sensing analysis, the CR can identify and use the vacant channels to transmit [3]. Spectrum decision refers to the ability of selecting the best vacant channel, according to some criteria, such as historical availability, channel quality, among others [5]. Spectrum sharing comprises the coordinated access to a vacant channel by two or more CR devices [13]. Finally, spectrum mobility function enables a CR device to leave the channel when another transmission is detected [14].

Among the four functions of a CR device, the spectrum sensing is considered the most important [3]. The main reason for this is because all the other functions can use the spectrum sensing results as input to perform their operations [3]. In addition, the spectrum sensing capability to analyze the RF spectrum and evaluate whether or not a channel is vacant is fundamental to the proper operation of the CR network. Therefore, considering the spectrum sensing importance in the design of CR devices, such as BSs and CPEs, we argue that a CR network management system must be based on the spectrum sensing function.

When a CR network comprises a large number of BSs and CPEs, its management may become complex, given that each BS and CPE can be dynamically configured, monitored and visualized. To facilitate the management of such complex systems, three management entities may be used according to a hierarchical model: (*i*) Managers are responsible for the management tasks, such as controlling the transmission of each CR device in the network or generating summarized reports about the overall network, (*ii*) Gateways are capable of executing ordinary actions, such as collecting information about the CR devices in the network and (*iii*) Agents are placed within CR devices and are responsible for responding the requests performed by Gateways and applying new configurations to these devices. In the literature, the Agent usually operates the IEEE 802.22 MIB [4], as explained below.

The IEEE 802.22 Standard specifies a MIB module [4] which provides the most relevant information for CR networks management systems. The MIB module is organized in seven groups. These groups are presented in Fig. 1. The *wranDevMib* group informs the software version of the device and which SNMP traps can be configured. The *wranIfBsMib* group presents objects related to the BS operation. The *wranIfBsSfMgmt* group refers to the management items associated with service flow configuration, instantiation, and management. The *wranIfCpeMib* group has objects related to operation of CPEs. The importance assigned to the spectrum sensing function is noticeable due to the *wranIfSmMib* and the *wranIfSsaMib* groups. The first group presents objects associated with the spectrum management, while the second group offers objects related to the spectrum sensing function management. Finally, the *wranIfDatabaseServiceMib* group has objects for the database service considering network general information.

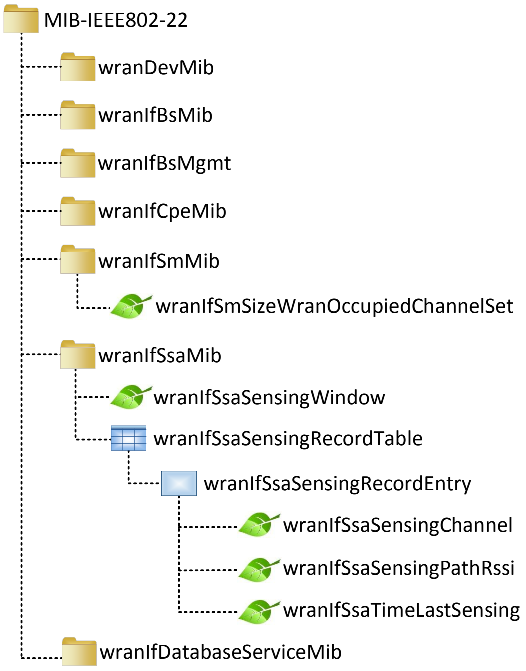


Fig. 1. IEEE 802.22 MIB groups structure

Considering the IEEE 802.22 MIB module, we can highlight some important objects for a CR network management system. These objects provide useful information regarding the spectrum sensing in each CR device. An important object is the occupied channel list (*wranIfSmWranOccupiedChannelSet*), which identifies all channels evaluated as occupied in the spectrum sensing result. The object associated with the sensing window configuration (*wranIfSsaSensingWindow*) determines how long each channel must be sensed to obtain the evaluation result. The sensed channel number (*wranIfSsaSensingChannel*), the Received Signal Strength Indication (RSSI) during the sensing (*wranIfSsaSensingPathRssi*), and the timestamp of most recent sensing (*wranIfSsaTimeLastSensing*) are also fundamental information to develop a CR network management system.

The communication between the Agent and the Gateway is realized through the Simple Network Management Protocol (SNMP). This protocol is particularly interesting to be used among BS and CPEs entities because all the messages exchanged among them are usually transmitted over a dedicated control channel [4], avoiding overhead of control messages in the communication. This control channel is selected before the initialization of the network operation and presents the best propagation conditions. Usually, this channel comprises a 6 MHz bandwidth channel, with transfer rate of $18Mbps$ and can be used only for control and management data between BSs and CPEs. Moreover, considering an IEEE 802.22 network with a maximum of 512 Agents and 2 seconds as the smallest time interval between two consecutive requests performed by the Gateway, the SNMP traffic will correspond to less than 5% of the control channel capacity. However, SNMP is not suitable to realize communications over the Internet, mainly due to the absence of a control channel and also because firewalls may block SNMP messages. In addition, SNMP security mechanisms causes network overhead [15].

One way to avoid SNMP over the Internet is using management approaches based on Web Services (WS) [16]. These approaches may be implemented through two architectures: Service Oriented Architecture (SOA) [17] and Resource Oriented Architecture (ROA) [18]. Both architectures may use Hypertext Transfer Protocol (HTTP) to transmit data, avoiding firewalls. Moreover, SOA and ROA take advantage of HTTP Secure (HTTPS) protocol to easily secure the messages being exchanged. However, Pautasso *et al.* [19] proved that ROA is better than SOA to achieve a higher performance when implementing a WS based approach.

ROA is a loosely coupled approach to the client-server model that uses a Uniform Resource Identifier (URI) to directly access devices resources. In general, this approach follows the Representational State Transfer (REST) architectural style [18]. This style defines HTTP as the only application protocol and standardizes the access interface through methods of this protocol (*i.e.*, GET, PUT, POST, and DELETE). Each REST message represents a state of the accessed resource, *i.e.*, the current collection of meaningful information (*e.g.* sensed channel list, sensing window, and sensing evaluation). Other features of REST includes the support to cache messages and resources inheritance. Despite all advantages, ROA was not used, at our best knowledge, to implement a sophisticated management application to CR networks. Therefore, we investigate and explain the related work about CR networks management in the next subsection.

B. Related Work

Much has been discussed about CR in the last decade. The architecture proposed by Wang *et al.* [9] uses the information provided by the CR devices to automatically configure the network access. The architecture was developed on a multiprocessor System-on-Chip but the authors highlight the need of a CR network management system. Potier and Quian [8] addressed the problem of management for CR ad-hoc networks and presented how the management of these networks is different than the traditional management for wired and wireless networks. Moreover, the authors proposed a cognitive network management protocol for ad-hoc networks. Wang *et al.* [10] provided an overview of the essential functionalities of the spectrum sensing and discussed its impact on making business level policies for CR networks management. Manfrin, Zanella, and Zorzi [11] proposed CRABSS[1], an open platform to monitor unlicensed spectrum bands. The proposed system export information about the spectrum sensing function to management tools, facilitating the development of spectrum utilization maps. Stavroulaki *et al.* [2] presented an overview about machine learning mechanisms to perform automatic decisions about configuration in CR networks. The authors also presented basic learning functionalities for the identification and processing of information that can lead to exploitable knowledge in CR networks.

The researches described above are specific solutions to CR management, *e.g.* the automatic configuration to access the CR network, an ad-hoc protocol, the application of policies and machine learning in these networks. In a CR network, CR devices should operate automatically, but we argue that this operation can be improved by combining the analysis of spectrum sensing results and the administrator intervention. An administrator may configure the cognitive functions parameters such as the range of channels to be sensed, sensing window, list of sensing channels, etc. However, to the best of our knowledge, no management system for CR networks based on the spectrum sensing function has been proposed. Therefore, there is a lack of management systems that enable the configuration, monitoring, and visualization of CR networks. To minimize this lack, we propose a management system called *Kitsune*, which allows administrators to acquire a better knowledge about a managed CR network. In the next section, we present the *Kitsune* system, the modules of its architecture, and the prototype developed and deployed in a real CR network.

III. KITSUNE MANAGEMENT SYSTEM

In this section, *Kitsune* management system is described in details. First, in the Subsection III-A we present and discuss the functional blocks of *Kitsune*, detailing each component of the system. Next, we describe the details related to the prototype developed to validate the operation of the proposed system in the Subsection III-B.

---

[1] ARAGORN Proejct – http://www.ict-aragorn.eu/

A. *Kitsune Functional Blocks*

Kitsune system aims to manage CR networks based on spectrum sensing. This system follows a hierarchical management model composed of one functional block for the Network Operation Center (NOC) and another block representing the CR network. These two functional blocks exchange information through a backhaul, which may be a private network infrastructure, *e.g.* a cellular operator network, or a shared infrastructure, such as the Internet. Fig. 2 illustrates in details *Kitsune* proposed system using a top-down approach.

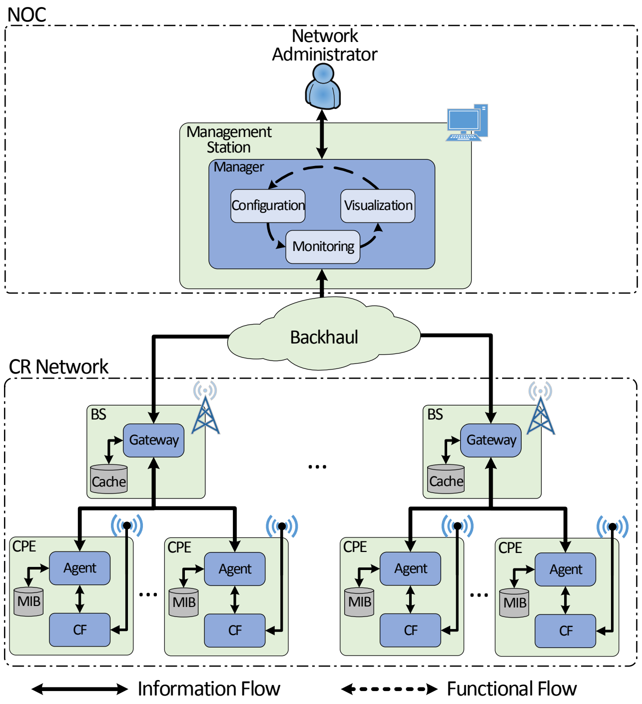


Fig. 2. *Kitsune* management system

The NOC comprises a physical entity called Management Station, as shown in Fig. 2. This Management Station presents a Manager component composed of three modules, Configuration, Monitoring, and Visualization. Through these three modules, a network administrator can improve the configuration of the spectrum sensing function of each CR device by monitoring and analyzing its results. Therefore, the functional flow inside the Manager represents a continuous learning process to the network administrator.

In the first stage of this learning process, the network administrator defines the configurations, using the Configuration module. In this sense, the network administrator defines which parameters of the spectrum sensing function will be managed (*e.g.* sensed channel list, sensing window, and sensing evaluation). In the second stage of the learning process, the Monitoring module will monitor the spectrum sensing results in each device of the CR network. Finally, the third stage is provided by the Visualization module which will show a composition of the historical information about the spectrum sensing results of each CR device. Since the learning process exhibits a cyclic behavior, the Visualization may be used as a feedback by the network administrator to perform new configurations on the CR network.

As defined by IEEE 802.22 standard, a centralized CR network is composed by two physical entities: BS and CPE [4]. BS is characterized by concentrating all the network information in an infrastructural wireless communication system. In this BS, we design the second component of *Kitsune*, called Gateway. This component is responsible for four main tasks: (*i*) delivering the configuration messages sent from the Manager to the spectrum sensing of each CPE, (*ii*) requesting the spectrum sensing results of each CPE, (*iii*) caching these results, and (*iv*) forwarding the results to the Manager component in a predefined time interval. In addition, the caching of the spectrum sensing results is an important task in the hierarchical management model, because it enables the Manager to obtain results from the Gateway without major overheads caused by messages exchanged directly with CPEs.

CPE is a physical entity responsible for providing network access to users. This entity is composed of three components: (i) MIB, (ii) Agent and (iii) Cognitive Functions (CF). A MIB is a conceptual database that organizes the information about the current settings and results of the spectrum sensing function. The Agent component receives configuration messages from the Gateway and organizes them into the MIB. Moreover, the Agent collects spectrum sensing results from the CF component and also organizes these results in the MIB. Finally, the CF component performs the spectrum sensing function through the use of SDR. In the next subsection, we describe the prototype implementation of the *Kitsune* system.

B. *Prototype Implementation*

Kitsune prototype implements both the NOC and the CR network functional blocks, as previously defined. The information exchange among system components may be performed through internal communication (*i.e.* inside the same device or computer) or external communication (*i.e.* among different devices and computers), as shown in Fig. 3.

The Manager summarizes the spectrum sensing results and creates a visualization for the network administrator through a Web based Graphic User Interface (GUI). These summarizations are stored in a MySQL database and queried by the Manager in a predefined time interval. The Manager and Gateway communications follows ROA because it enables a better usage of the Internet infrastructure. In addition, Managers and Gateways communication is performed over an encrypted connection using HTTPS. Finally, communication among Gateways and CPEs is typically performed over LANs, making SNMP preferable in this case. As described previously, the Gateway possesses four tasks, which are implemented through a MySQL database in combination with services provided to the Manager. In addition, services and parameters supported by a Gateway are presented in Table I.

Accordingly to Table I, when a Manager requests the service *beginSession*, the Gateway creates a session, authenticated by

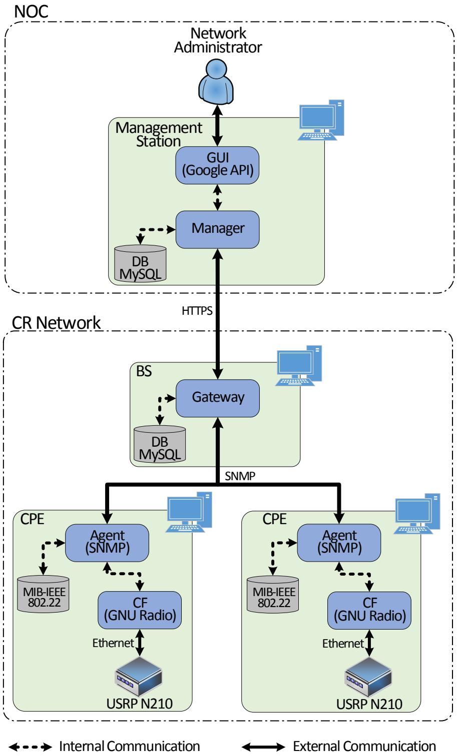


Fig. 3. *Kitsune* system prototype

TABLE I
GATEWAY SERVICES

Service	**Parameter**	**Return**
beginSession	*username, password*	*sessionID*
endSession	*sessionID*	-
setConfiguration	*sessionID, configuration*	*configStatus*
setCpeConfiguration	*sessionID, cpeID, cpeConfiguration*	*cpeConfigStatus*
setCpeList	*sessionID, newCpe*	*cpeList*
getCpeList	*sessionID*	*cpeList*
getCpe	*sessionID, cpeID*	*cpeSensingInfo*
getSensingInfo	*sessionID*	*sensingInfo*

the *username* and *password* parameters. This service starts an authorized session that can be finished using the *endSession* service. To configure the Gateway, the *setConfiguration* service can be used. Moreover, the *setCpeConfiguration* service can be performed to directly configure each CPE. The *setCpeList* and *getCpeList* services are implemented to directly access the list of Agents presented in the Gateway. The *getCpe* service returns the spectrum sensing results about a CPE, identified by the *cpeID* parameter. The results of this service may be an occupied channel list (*wranIfSmWranOccupiedChannelSet*), the sensing window configuration (*wranIfSsaSensingWindow*), the received signal intensity (*wranIfSsaSensingPathRssi*), among others objects of the IEEE 802.22 MIB. The MIB presented in each Agent is based on the IEEE 802.22 MIB, as shown in Fig 1. Finally, the *getSensingInfo* service returns the spectrum sensing results about all the CPEs monitored in the Gateway.

The Agents are deployed in Ettus USRP N210 devices with a computer-hosted SDR kit that enable the fast development of radio applications [20] through an SDR framework called GNURadio [21]. In addition, these devices perform the spectrum sensing function. The USRP device operates according to the algorithms implemented in the personal computer connected with the USRP through a Gigabit Ethernet interface. We developed a Python script to perform the spectrum sensing function that interacts with the MIB. This script has implemented an energy detector based spectrum sensing technique, given its wide adoption [3]. The script uses the parameters configured by the network administrator, such as the list of channels to sense, sensing period, and sensing window. Finally, results of the SSF are stored in the MIB and both the configurations and results are manipulated by the Agents.

Once defined the *Kitsune* system and the prototype implementation, in the next section we present in details how we deployed *Kitsune* system prototype in three different scenarios to be evaluated. In addition, we describe the experiments performed over each scenario where *Kitsune* was evaluated and discussed in details according to the obtained results.

IV. KITSUNE EVALUATION

Kitsune prototype was deployed and evaluated according to three scenarios. Each scenario was based on the IEEE 802.22 standard, where analogic TV channels are used to provide Internet access [4]. In Table II, parameters and their respect values are representing hardware and software configurations used for these three scenarios during all experiments.

TABLE II
KITSUNE EVALUATION SCENARIO PARAMETERS AND VALUES

Parameter	**Value**
Execution time	60s
USRP sample rate	10^6 samples/s
Number of sensed channels	5
λ	$[1, 2, 3, 4, 5]$s
Sensing Window	.11s
Sensing Period	$[0.1, 2]$s
P_M	30s
P_G	2s

According to Table II, for each scenario experiments were performed during 60 seconds, considering a USRP sample rate of 10^6 samples/second. We defined 5 different channels to be sensed by 5 CPEs. In addition, users behavior in the sensed channels was modeled through a Poisson distribution [22] that changes the vacancy status of the channels with a mean and variance (λ) for each channel. These channels were sensed during a sensing window of 0.1 second and the interval between each spectrum sensing function execution was set to be a different period for each scenario. Another two important parameters are the polling waiting time for Managers requests to Gateway (P_M), setted to 30 seconds, and the polling waiting time for Gateway requests to Agents (P_G), setted to 2 seconds. We defined the P_M higher than the

P_G on purpose to verify that the Gateway summarizes properly the informations obtained in a smaller interval.

In the Subsection IV-A, the first experimental scenario used to visualize the channel evaluation is described and results obtained are discussed. Similarly, in the Subsection IV-B we described the second scenario and discussed the results gathered during the evaluation of transmissions performed by the CPEs. Finally, in Subsection IV-C we evaluate how the network performance can be improved by a network administrator that analyzes the visualizations and decide new configurations for the spectrum sensing function.

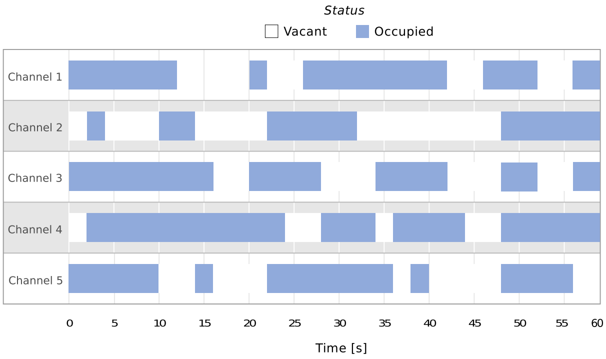
A. *Channel Status Evaluation*

In this scenario, the BS defines a silent phase where CPEs are not able to perform transmission. In this phase, CPEs should periodically perform the spectrum sensing over the channels defined by the BS, using a sensing period setted to 2 seconds. The spectrum sensing results are requested from CPEs by a BS to determine the actual status of the channels occupancy. The main goal of this scenario is to verify that *Kitsune* allows the network administrator to analyze results of the spectrum sensing performed by CPEs, enabling the improvement of the knowledge about users behavior in the sensed channels. Fig. 4 shows three BS visualizations through the *Kitsune* interface.

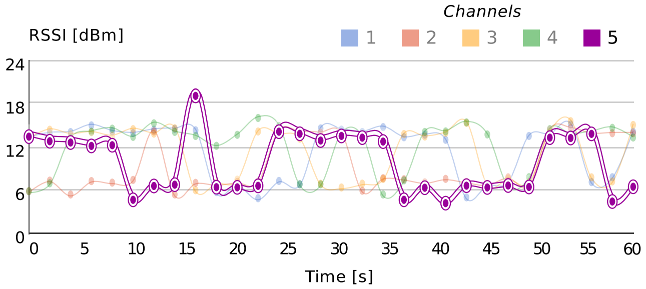
In Fig. 4, we can observe three visualizations provided by *Kitsune* interface from the BS perspective. The occupancy status of the sensed channels, during a time period of 60 seconds, can be observed in Fig 4(a). Through this visualization, a network administrator may observe the user behavior in the sensed channels and make some considerations. For example, the administrator may conclude that channel 4 is a bad option to be selected for transmission, because it has longer periods of occupancy when compared to others. It means that a CPE in channel 4 would have to often stop its transmission due to the heavy usage of the channel, decreasing its overall throughput. Different from channel 4, channel 2 presents a lower usage, becoming a potential option to be selected for transmissions.

Kitsune also provides a visualization of the RSSI in each sensed channel. This is an important information that can be used to analyze the quality of the sensed channels, as can be seen in Fig. 4(b). Associating this visualization with Fig. 4(a) it is possible to stipulate the sensibility of the spectrum sensing, *i.e.* , the minimum RSSI necessary to consider a channel as occupied. The network administrator may use this information to adjust the sensing sensibility to a desired level. In addition, the administrator may highlight the information of a specific channel to analyze, as exemplified in Fig. 4(b), where the highlighted line represents the RSSI of channel 5.

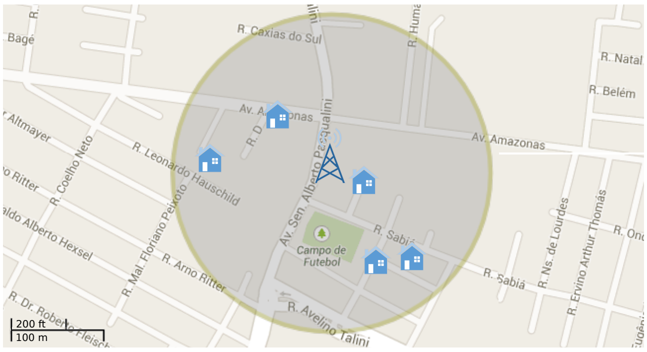
Associating Fig. 4(a) and Fig. 4(b) we can observe the spectrum sensing defining channel 5 as occupied, when the sensed channel presents a high RSSI (*e.g.* about 16 dBm). On the other hand, when a low RSSI is observed in the channel (*e.g.* about 6 dBm), it will be defined as vacant. It is a characteristic of the energy detector spectrum sensing technique used in this paper, which compares the energy sensed in the channel with a threshold to define the channel status as vacant or occupied.



(a) Channels status



(b) Channels RSSI



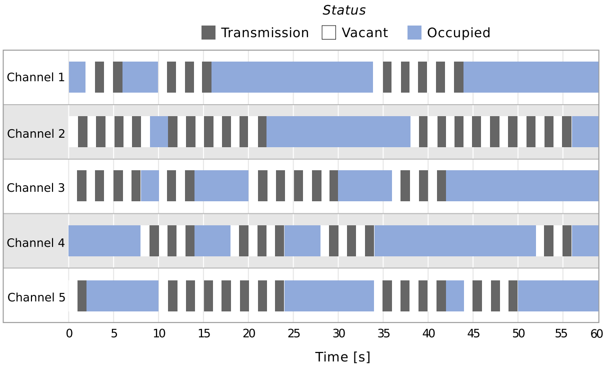
(c) Basestation and CPEs geolocation

Fig. 4. BS visualizations

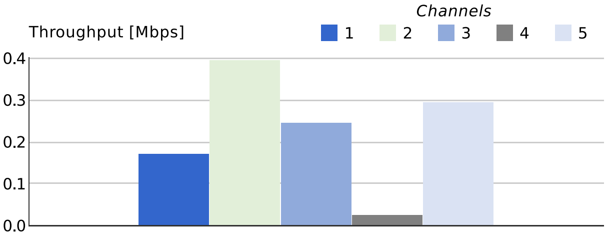
The geolocation of BS and CPEs may be used by *Kitsune* to display a map of the network, as can be seen in Fig. 4(c), based in an interpolation method of geolocation information associated to Okumura-Hata propagation model to define coverage area. In this map, the BS is represented by a tower icon and its coverage area is delimited by the colored circle. Its associated CPEs are represented by house icons. This visualization can be combined with the channel vacancy status to provide the network administrator an overview about the network. The combined use of these visualizations is a powerful tool to achieve a better understanding about the sensed channels.

B. *CPE transmissions*

The second scenario is based on the transmission performed by each CPEs that compose the CR network. In this case, CPEs should perform the spectrum sensing to verify the status of the channel and transmit only if the channels status are defined as vacant. Once verified the status of the channel, the CPE starts its transmission over it until a new spectrum sensing is performed. The interval between each spectrum sensing is given by the sensing period. In this scenario, *Kitsune* shows the information about CPEs transmissions over the channel configured for each CPE. Moreover, with the analysis of the transmissions, administrators can reconfigure the network to maximize the throughput obtained by a specific CPE. Results of the transmission analysis are shown in Fig. 5.



(a) Channels tranmissions status



(b) CPEs throughput

Fig. 5. CPE transmissions visualization

As can be seen in the Fig. 5, we designed two different visualizations of the transmissions performed by CPEs.. Fig. 5(a) shows a visualization similar to Fig. 4(a). However, in addition to showing the results of spectral sensing, this new visualization enable an important analysis not available through Fig. 4(a). This analysis is related to transmissions performed by CPEs over the channels, highlighted with a different color for vacant and occupied status. With an analysis over this visualization, the network administrator can determine which is the channel that presents fewer opportunities for transmission. Once the network administrator knows which is the worst channel, in terms of number of transmissions, this channel can be avoided, and another channel can be defined to be analyzed by a CPE.

Complementary to the visualization shown in Fig. 5(a), Fig. 5(b) allows another important analysis about the transmissions performed by the CPEs. This information is related to the total throughput obtained in the transmissions. We can associate this throughput directly to the transmissions shown in Fig. 5(a), identifying which is the channel with higher throughout. On the other hand, the worst channel, in terms of throughput and number of transmission can be identified by the administrator that may configure the network to ignore this channel, avoiding wasting time by sensing a bad channel for transmission. In the example of Fig. 5, we can observe that the channel 2 presents a higher throughput than the others, being a better choice to be used for transmissions. However, channel 4 presents the lower total throughput. The administrator can analyze the occupancy, throughput, and the RSSI of channels in the visualization and define which is the best channel. For example, A channel with 95% of occupancy, 6dB of RSSI and 30Kbps is a bad channel while a channel with 15% of occupancy, 18dB of RSSI and 1Mbps is a good channel.

C. *Configuration Analysis*

Finally, the last scenario is also based on the transmissions performed by CPEs. However, in this evaluation we are interested in analyzing the impact of a new configuration settled by the network administrator after analyzing the initial configuration. To perform this evaluation, we stored the results of the spectrum sensing function to apply the same channel occupancy for both configurations. Then, we setted an initial sensing period and obtained the average throughput by a CPE in each channel. Results were gathered and visualized through *Kitsune* system. Afterwards, a new configuration for the sensing period was set to monitor the results. These results are shown in Table III.

TABLE III
THROUGHPUT RESULTS WITH DIFFERENT CONFIGURATIONS

Channel	**Sensing Period [s]**	**Throughput [Mbps]**	**Variation [%]**	**Transmissions**
1	1	0.3182	42.04	18
	2	0.5490		25
2	1	0.2267	51.84	15
	2	0.4708		24
3	1	0.4016	18.42	22
	2	0.4923		24
4	1	0.1803	42.54	12
	2	0.3138		17
5	1	0.4027	17.25	21
	2	0.4867		23

According to Table III, we defined two different configurations for the sensing period: 1 second and 2 seconds. In both cases, the throughput obtained for each channel varied. This variation is given by the behavior of the network users, which directly impacts in the results. Moreover, one important conclusion from the analysis of the generated visualizations: the sensing period parameter of a CPE impacts in the average throughput obtained during the transmissions. In the fourth column of the Table III is shown the variation of the average throughput obtained in each sensing period. For all the channels, the average throughput increases when the sensing period is higher. This can be explained due the fact that a CPE will interrupt their transmissions to perform the sense with less frequency, allowing a higher transmission time.

Another important observation that can be highlighted is that the variations between channels varies for each channel. For example, the difference of the average throughput obtained in the channel 1 was 42.04%, while the channel 3 presents a variation of 18.42%. To understand these variations, we should observe the number of transmissions performed in each channel. These numbers varies according to the behavior of the users on the channel. For example, in channel 1, with a sensing period of 1 second, 18 transmissions were performed. Meanwhile, 22 transmissions were performed in channel 3, with the same sensing period. Increasing this period to 2 seconds, increased by 7 the number of transmissions of the former channel, meanwhile it increased only by 2 the number transmissions of channel 4. Thus, the number of transmission directly impacts in the average throughput of the CPEs. Therefore, *Kitsune* system is able to provide an overview about throughput variations, their cause and consequence, which combined with former scenarios results from subsection IV-A and IV-B become powerful tools for the administrator to understand the behavior of the users in a CR network and to participate in all three stages of the continuous learning process.

Once proved the proper operation of the *Kitsune* system, in the next section we present the final conclusions obtained through the development of this work and discuss future work related to the improvement of *Kitsune*.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed *Kitsune*, a management system for CR networks that helps administrators to learn how to improve the configuration of the spectrum sensing function. *Kitsune* allows administrators to participate in a continuous learning process based on the configuration, monitoring and visualization of the spectrum sensing function through a Graphical User Interface. A prototype of *Kitsune* was implemented and evaluated through a real RF environment based on IEEE 802.22 WRAN scenario, showing the operation of the proposed management system.

Results show that *Kitsune* system allows network administrators to improve the throughput obtained in transmissions performed by CPEs by gathering results (monitoring), applying visualizations techniques (visualization), and finally, changing the configurations of these devices (configuration). In summary, *Kitsune* enables administrators to participate actively in the spectrum sensing monitoring, controlling, and visualization. Such a participation allows *Kitsune* to readjust a CR network to achieve a better throughput, in average, as described in Section IV.

There are several avenues for future research. We intend to further extend *Kitsune* operation, allowing the configuration, monitoring and visualization of the rest of cognitive functions performed by cognitive devices. Moreover, *Kitsune* can be improved in terms of functionalities through new mechanisms that improve the network operation. For example, machine learning algorithms may be implemented in *Kitsune* to identify the best vacant channels in terms of historical availability, propagation conditions, and also select channels to be used by CPEs according to Quality of Service (QoS) policies.

ACKNOWLEDGEMENTS

The authors thanks to Digitel S.A., Fundação de Apoio a Pesquisa do Estado do Rio Grande do Sul (FAPERGS), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Nível Superior (CAPES) for the financial support.

REFERENCES

[1] I. Mitola, J. and J. Maguire, G.Q., “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, pp. 13–18, 1999.

[2] V. Stavroulaki, A. Bantouna, Y. Kritikou, K. Tsagkaris, P. Demestichas, P. Blasco, F. Bader, M. Dohler, D. Denkovski, V. Atanasovski, L. Gavrilovska, and K. Moessner, “Knowledge Management Toolbox: Machine Learning for Cognitive Radio Networks,” *IEEE Vehicular Technology Magazine*, pp. 91–99, june 2012.

[3] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Communications Surveys Tutorials*, vol. 11, pp. 116–130, 2009.

[4] IEEE, “Standard for Information Technology - Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN) - Specific requirements Part 22,” *IEEE Std 802.22*, pp. 1–680, 2011.

[5] M. T. Masonta, M. Mzyece, and N. Ntlatlapa, “Spectrum decision in cognitive radio networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1088–1107, 2013.

[6] M. Dardaillon, K. Marquet, T. Risset, and A. Scherrer, “Software defined radio architecture survey for cognitive testbeds,” *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 189–194, 2012.

[7] J. Wang, M. Ghosh, and K. Challapali, “Emerging cognitive radio applications: A survey,” *IEEE Communications Magazine*, pp. 74–81, march 2011.

[8] P. Potier and L. Qian, “Network management of cognitive radio ad hoc networks,” *International Conference on Cognitive Radio and Advanced Spectrum Management (CogART)*, pp. 1–5, 2011.

[9] S. Wang, L. Xie, H. Liu, B. Zhang, and H. Zhao, “Acra: An autonomic and expandable architecture for cognitive radio nodes,” *International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, october 2010.

[10] C.-X. Wang, H.-H. Chen, X. Hong, and M. Guizani, “Cognitive radio network management,” *IEEE Vehicular Technology Magazine*, vol. 3, no. 1, pp. 28–35, 2008.

[11] R. Manfrin, A. Zanella, and M. Zorzi, “CRABSS: CalRAdio-Based advanced Spectrum Scanner for cognitive networks,” *Wireless Communication & Mobile Computing*, vol. 10, no. 12, pp. 1682–1695, 2010.

[12] I. Akyildiz, W.-Y. Lee, M. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” *IEEE Communications Magazine*, pp. 40–48, 2008.

[13] J. Peha, “Sharing spectrum through spectrum policy reform and cognitive radio,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 708–719, 2009.

[14] I. Christian, S. Moh, I. Chung, and J. Lee, “Spectrum mobility in cognitive radio networks,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 114–121, 2012.

[15] G. Pavlou, P. Flegkas, S. Gouveris, and A. Liotta, “On management technologies and the potential of web services,” *IEEE Communications Magazine*, vol. 42, no. 7, pp. 58–66, 2004.

[16] G. Moura, G. Silvestrin, R. Sanchez, L. Gaspary, and L. Granville, “On the Performance of Web Services Management Standards - An Evaluation of MUWS and WS-Management for Network Management,” *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 459–468, 2007.

[17] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Winer, “Simple object access protocol (SOAP) 1.1,” 2000.

[18] R. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, 2000.

[19] C. Pautasso, O. Zimmermann, and F. Leymann, “Restful web services vs. ”big”’ web services: making the right architectural decision,” *Proceedings of the 17th international conference on World Wide Web*, pp. 805–814, 2008.

[20] Ettus Research, “USRP2,” http://www.ettus.com.

[21] Free Software Foundation, “GNU radio,” www.gnuradio.org, 2004.

[22] C. Ghosh, S. Pagadarai, D. Agrawal, and A. Wyglinski, “A framework for statistical wireless spectrum occupancy modeling,” *IEEE Transactions on Wireless Communications*, vol. 9, pp. 38–44, 2010.

APPENDIX C PUBLISHED PAPER – SBRC 2013

In this paper a first investigation about the management of CR networks is presented, proposing an initial management system to improve the throughput obtained in the transmission, by adjusting the sensing time and period (in Portuguese).

- **Title:**
  Uma Solucão para Gerenciamento de Dispositivos de Rádio Cognitivo Baseada na MIB IEEE 802.22
- **Conference:**
  XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2013)
- **URL:**
  `http://sbrc2013.unb.br/`
- **Date:**
  May 6-10, 2013
- **Held at:**
  Royal Tulip Brasília Alvorada, Brasília, Distrito Federal, Brazil
- **Digital Library of SBC:**
  `http://sbrc2013.unb.br/files/anais/trilha-principal/artigos/artigo-46.pdf`

Uma Solução para Gerenciamento de Dispositivos de Rádio Cognitivo Baseada na MIB IEEE 802.22

Lucas Bondan[1], **Maicon Kist**[1], **Rafael Kunst**[1],
Cristiano B. Both[1], **Juergen Rochol**[1], **Lisandro Z. Granville**[1]

[1]Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brasil

`{lbondan, maicon.kist, rkunst, cbboth, juergen, granville}@inf.ufrgs.br`

Abstract. *Cognitive Radio technology allows wireless devices to transmit information while the channels are not in use at time. However, it is necessary a system that helps the dynamic configuration process of the cognitive devices for proper operation of cognitive radio networks. In this paper, a management system for cognitive radio devices is proposed based on the IEEE 802.22 MIB. The system's main objective is to manage and to monitor the spectrum sensing process. The results obtained in simulations show that the system enables to maximize the throughput of the devices by setting the sensing window according to the confidence on the state of a channel, i.e., free or busy.*

Resumo. *A tecnologia de Rádio Cognitivo permite que dispositivos sem fio transmitam informações enquanto os canais não estiverem em uso em um determinado instante de tempo. Entretanto, para o correto funcionamento das redes de rádios cognitivos, é necessário um sistema que auxilie no processo de configuração dinâmica dos dispositivos cognitivos. Neste trabalho é proposto um sistema de gerenciamento especializado para dispositivos de rádio cognitivo baseado na MIB IEEE 802.22. O sistema tem como objetivo principal o gerenciamento e a monitoração do processo de sensoriamento espectral. Os resultados, obtidos através de simulações, mostram que o sistema possibilita maximizar a vazão na transmissão de informações dos dispositivos, ajustando a janela de sensoriamento de acordo com os níveis de confiança para percepção de que um determinado canal está ocupado.*

1. Introdução

Técnicas de acesso dinâmico ao espectro de frequências (*Dynamic Spectrum Access* - DSA) estão sendo utilizadas para melhorar a eficiência de modernos dispositivos de redes sem fio [Zhao e Sadler 2007]. Os dispositivos que implementam essas técnicas, possuem a capacidade de acessar dinamicamente uma determinada frequência do espectro que não está em uso em um determinado instante de tempo. Atualmente, a principal tecnologia que provê a implementação de técnicas de acesso dinâmico ao espectro é chamada de Rádio Cognitivo [Wang *et al.* 2011]. Essa tecnologia permite que os dispositivos ajustem o seu funcionamento para maximizar suas taxas de transmissão e minimizar a interferência em relação a outros dispositivos sem fio [Khalid e Anpalagan 2010]. Entretanto, para o correto funcionamento das redes de rádios cognitivos é necessário um sistema de gerenciamento que auxilie no processo de configuração dinâmica dos dispositivos cognitivos.

As características dinâmicas dos dispositivos de rádio cognitivo exigem a configuração de vários parâmetros que devem ser gerenciados e monitorados. Neste sentido, o primeiro grupo de trabalho IEEE a padronizar a utilização da tecnologia de rádio cognitivo foi o 802.22, através da definição de uma *Management Information Base* (MIB) para gerenciar e monitorar dispositivos de uma rede de rádios cognitivos [IEEE 802.22 2011]. Um exemplo da necessidade de um sistema de gerenciamento em redes de rádios cognitivos está no processo conhecido como sensoriamento espectral. Neste processo, o dispositivo cognitivo deve analisar, durante um intervalo de tempo, se um determinado canal está em uso ou não [Yucek e Arslan 2009]. A confiabilidade do resultado dessa análise está diretamente relacionada à duração do sensoriamento, ou seja, quanto mais tempo de sensoriamento, maior o nível de confiança. Considerando apenas esses dois parâmetros de configuração, tempo e confiabilidade do sensoriamento, já percebe-se a necessidade de um sistema especializado de gerenciamento para a configuração do dispositivo e posterior monitoramento da suas funcionalidades, a fim de garantir o correto funcionamento dos dispositivos.

A comunidade científica em gerenciamento de redes apresenta poucos trabalhos sobre redes de rádios cognitivos. Neste artigo, as poucas propostas encontradas na literatura são classificadas de acordo com dois escopos: Global e Local. No primeiro escopo, encontram-se os trabalhos publicados por Wang *et al.* [Wang *et al.* 2008] e Stavroulaki *et al.* [Stavroulaki *et al.* 2012], nos quais o enfoque é no gerenciamento global de redes de rádios cognitivos. No segundo escopo, os trabalhos publicados por Wang *et al.* [Wang *et al.* 2010] e por Potier e Quian [Potier e Qian 2011] investigam como devem ser gerenciadas as informações e os parâmetros em um dispositivo de rádio cognitivo em específico. Entretanto, não se tem conhecimento de um sistema de gerenciamento especializado para a configuração e monitoração de dispositivos de rádio cognitivo, nem mesmo como esse sistema deve ser projetado, considerando a dinamicidade do sensoriamento espectral.

A principal contribuição deste artigo é propor um sistema de gerenciamento para dispositivos de rádio cognitivo considerando um escopo Local. O sistema especializado baseia-se nas informações definidas na MIB IEEE 802.22, sendo o objetivo principal deste artigo o gerenciamento e a monitoração do processo de sensoriamento espectral. Para atender à dinamicidade dos parâmetros de configuração de dispositivos de rádio cognitivo, o sistema foi modelado considerando um gerenciamento automático. Desta forma, o sistema proposto possibilita a configuração automática dos dispositivos baseada em regras para adequação de parâmetros de configuração. Os resultados obtidos através de simulações mostram que o sistema possibilita maximizar a vazão na transmissão dos dispositivos, ajustando a janela de sensoriamento e garantindo a confiabilidade na detecção de canal ocupado.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta a fundamentação teórica sobre redes de rádios cognitivos, bem como o estado da arte sobre as pesquisas em gerenciamento de redes de rádios cognitivos. Na Seção 3, é descrita a arquitetura da solução proposta e o protótipo desenvolvido como prova de conceito. Os detalhes sobre o modelo do sistema e os resultados obtidos são apresentados na Seção 4. Por fim, a Seção 5 conclui o presente trabalho e apresenta perspectivas para trabalhos futuros.

2. Fundamentação Teórica

Nesta seção são abordados os elementos que definem os conceitos sobre dispositivos de rádios cognitivos e seu gerenciamento. Na Subseção 2.1, são descritos os principais mecanismos presentes em rádios cognitivos, que tornam possível a comunicação entre estes dispositivos. Na Subseção 2.2, são abordados os trabalhos relacionados ao gerenciamento de redes compostas por dispositivos de rádio cognitivo.

2.1. Gerência em Rádios Cognitivos

O rádio cognitivo pode ser definido como um dispositivo capaz de mudar seus parâmetros dinamicamente, através da análise do ambiente em que se encontra, sem interferir na operação dos demais dispositivos. As características fundamentais do rádio cognitivo são a capacidade cognitiva e a reconfigurabilidade [Akyildiz *et al.* 2008]. Através da capacidade cognitiva, o dispositivo pode selecionar o melhor canal disponível para transmissão em um determinado instante de tempo. Além disso, a reconfigurabilidade permite que o rádio cognitivo ajuste os parâmetros mais adequados para a utilização do espectro selecionado.

Com a mudança dinâmica dos parâmetros nas redes de rádios cognitivos, os dispositivos dotados da capacidade cognitiva devem ser capazes de modificar suas configurações para se adaptar às mudanças no ambiente em que se encontram, por exemplo ajustando a frequência em que estão trabalhando. Isso gera a necessidade de gerenciar essas configurações, a fim de manter a rede de comunicação operando de maneira satisfatória, sem interferir em transmissões de outros dispositivos que eventualmente utilizem o mesmo canal [Coutinho *et al.* 2012]. Dentre os trabalhos para gerenciamento das informações, encontra-se a MIB IEEE 802.22 [IEEE 802.22 2011]. A organização das informações no formato de uma MIB facilita o acesso e manipulação dessas informações, possibilitando o desenvolvimento de sistemas de gerenciamento para redes de rádios cognitivos. A MIB IEEE 802.22 é organizada em sete grupos, separados de acordo com a função exercida por cada elemento que compõe a rede: estação radio-base (*Base Station* - BS) ou dispositivo nas dependências do usuário (*Customer Premisse Equipment* - CPE). Esses grupos são descritos a seguir:

- *wranDevMib*: apresenta objetos contendo informações comuns a todos os dispositivos da rede;
- *wranIfBsMib*: apresenta objetos referentes exclusivamente à operação da BS;
- *wranIfBsSfMgmt*: apresenta objetos contendo informações sobre o fluxo de serviços de configuração, instanciação e gerência da BS;
- *wranIfCpeMib*: apresenta objetos referentes à operação móvel ou fixa dos CPEs;
- *wranIfSmMib*: apresenta objetos relacionados ao gerenciamento da utilização do espectro de frequências;
- *wranIfSsaMib*: apresenta objetos relacionados ao processo de sensoriamento espectral;
- *wranIfDatabaseServiceMib*: apresenta objetos relacionados ao serviço de banco de dados com informações gerais da rede.

Alguns dos objetos presentes na MIB IEEE 802.22 fornecem informações relevantes para maximizar o desempenho da rede. Dentre esses objetos, pode-se citar o grupo que provê informações sobre o sensoriamento do espectro (*wranIfSsaMib*), bem como objetos que provêm informações sobre a vazão obtida na transmissão de dados pelos dispositivos (*wranCpeTxThroughput*, por exemplo). A Figura 1 ilustra a estrutura dos sete grupos da MIB IEEE 802.22, apresentando alguns dos objetos que foram analisados no desenvolvimento deste artigo.

Dentro desses sete grupos, pode-se destacar alguns objetos importantes para o desenvolvimento deste artigo, visto que impactam diretamente na configuração e na aplicação do sensoriamento espectral. Por exemplo, os que se referem a configuração da janela

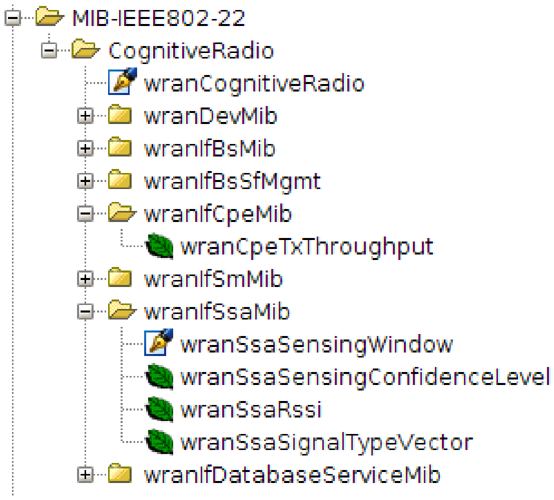


Figura 1. MIB IEEE 802.22

de sensoriamento (*wranSsaSensingWindow*). Esta janela determina por quanto tempo o dispositivo de rádio cognitivo irá analisar o canal, a fim de descobrir se o canal está ocupado. A MIB apresenta também o objeto que provê o resultado da vazão obtida por cada dispositivo na transmissão de dados (*wranCpeTxThrougput*). O nível de confiança na detecção de canais ocupados também está presente (*wranSsaSensingConfidendeLevel*), assim como o objeto que define a intensidade do sinal reconhecido no sensoriamento (*wranSsaRssi*) e o vetor que identifica o tipo de sinal recebido (*wranSsaSignalTypeVector*), como por exemplo *Orthogonal Frequency Division Multiplexing* (OFDM) ou microfone sem fio.

Além da MIB IEEE 802.22, a literatura sobre gerência de redes apresenta propostas para o gerenciamento de redes de rádios cognitivos. A Subseção 2.2 apresenta os principais trabalhos sobre esse tópico de pesquisa na literatura.

2.2. Trabalhos Relacionados

Neste artigo, o gerenciamento de dispositivos de rádio cognitivo foi classificado em dois escopos. O primeiro, chamado escopo Global, tem como objetivo gerenciar as informações pertinentes às redes de rádios cognitivos. Essas informações precisam ser compartilhadas por todas as redes, por exemplo, as frequências em uso nas redes adjacentes e geolocalização dos dispositivos. O segundo escopo é definido como Local, sendo responsável por gerenciar informações presentes em cada dispositivo que compõe a rede. Este escopo é composto por informações armazenadas localmente em cada dispositivo, como os objetos presentes na MIB IEEE 802.22.

A literatura referente ao escopo Local, foco de investigação deste artigo, apresenta poucos trabalhos publicados. Um desses trabalhos é o artigo de Potier e Quian [Potier e Qian 2011], que apresenta um protocolo para gerência de dispositivos de rádio cognitivo, definindo a atuação do rádio cognitivo nas diferentes camadas de protocolos. O protocolo proposto pelos autores é elaborado de forma que as decisões tomadas não afetem a operação dos demais dispositivos que compõem a rede de rádios cognitivos. Entretanto, a solução não apresenta análises sobre a correlação entre os parâmetros de configuração, como por exemplo o impacto da alteração do tempo de sensoriamento no desempenho da rede.

Outro trabalho sobre gerenciamento local foi publicado por Wang *et al.* [Wang *et al.* 2010], que propõe uma arquitetura para manipulação local das informações dos dispositivos. Essa arquitetura possibilita que o dispositivo de rádio cognitivo configure automaticamente seu acesso à rede. Baseado na arquitetura proposta, os autores elaboraram um sistema multiprocessado em um único chip (*Multiprocessor System-on-Chip* - MPSoC). Embora a arquitetura proposta no trabalho considere que os dispositivos necessitam de gerenciamento, o trabalho não indica como esse gerenciamento deve ser realizado.

Baseado na literatura sobre gerenciamento de rádio cognitivo, pode-se perceber a necessidade de um sistema de gerência Local, que considere a dinamicidade do sensoriamento espectral. Dessa forma, este artigo propõe um sistema de gerenciamento capaz de ajustar automaticamente os parâmetros de configuração necessários para maximizar a vazão na transmissão de dados do dispositivo de rádio cognitivo, enquanto o nível de confiabilidade é mantido. Os detalhes do sistema proposto são descritos na Seção 3.

3. Sistema de Gerenciamento para Dispositivo de Rádio Cognitivo

Nesta seção estão descritos os detalhes do sistema de gerenciamento proposto para dispositivos de rádio cognitivo. Inicialmente, na Subseção 3.1 é apresentada e discutida a arquitetura do sistema. Em seguida, na Subseção 3.2, são descritos detalhes relativos ao desenvolvimento do protótipo utilizado para comprovar o funcionamento da solução proposta.

3.1. Arquitetura

O sistema de gerenciamento proposto neste artigo foi projetado dentro do escopo Local de redes de rádio cognitivos. Uma rede de rádios cognitivos é composta por dois tipos de dispositivos: BS e CPE. Cada CPE possui um conjunto de informações gerenciáveis, isso é, informações que representam a configuração ativa no dispositivo, como por exemplo o tamanho da janela de sensoriamento. Além disso, o dispositivo armazena as informações obtidas após o processo de sensoriamento, como a taxa de acerto na identificação dos canais livres.

A BS caracteriza-se por concentrar todas as informações da rede em um sistema de comunicação sem fio. Essa característica torna a BS o melhor elemento para implantação do sistema de gerenciamento. Dessa forma, o sistema de gerenciamento proposto opera na BS. Assim, o sistema é capaz de gerenciar todos os dispositivos que fazem parte da rede de rádios cognitivos, bem como processar as informações obtidas e realizar os ajustes de configuração necessários.

O sistema de gerenciamento proposto baseia-se nas configurações estabelecidas no início da operação de gerência, realizadas pelo administrador do sistema. Posteriormente, todo o processo de gerência das operações é realizado de maneira automática. A operação automática do sistema de gerenciamento é proposta devido a grande variabilidade dos parâmetros de configuração que os dispositivos cognitivos possuem. Apesar de não necessitar da intervenção do administrador da rede durante sua execução, o sistema de gerenciamento pode ter suas operações paralisadas a qualquer momento, permitindo que o administrador configure os parâmetros do sistema de acordo com a necessidade. A Figura 2 ilustra em detalhes a arquitetura do sistema de gerenciamento proposto.

A operação do sistema de gerenciamento é formada por quatro módulos: (i) Interface, (ii) Regras, (iii) Coleta e (iv) Configuração. O sistema inicia sua operação pelo

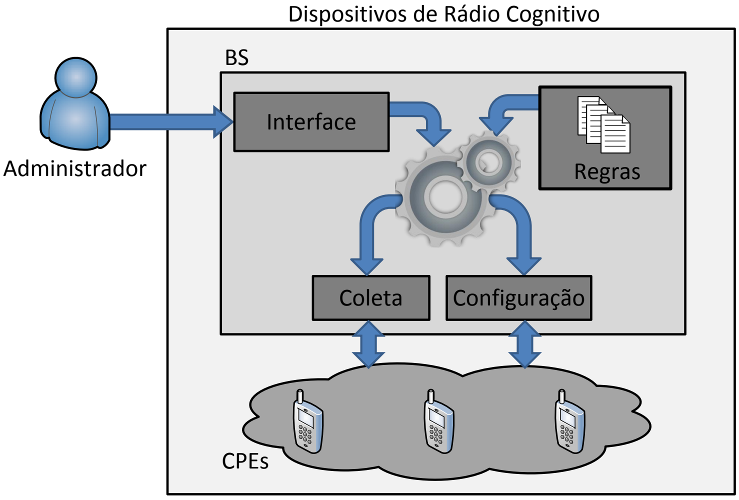


Figura 2. Arquitetura do sistema de gerência proposto

módulo de Interface. Nesse módulo, o administrador configura os valores desejados para os fatores utilizados nas Regras de configuração do rádio cognitivo. Essas regras utilizam limiares para ajustar os valores dos parâmetros de configuração dos dispositivos, podendo ser descritas pelo próprio administrador da rede. Os módulos de Coleta e Configuração têm como objetivo obter informações e aplicar ajustes na configuração dos dispositivos de rádio cognitivo. Baseado na arquitetura proposta, foi desenvolvido um protótipo do sistema de gerenciamento. Os detalhes dos protótipo são descritos na subseção a seguir.

3.2. Protótipo

Para comprovar o funcionamento do sistema de gerenciamento proposto, foi desenvolvido um protótipo em linguagem de programação C baseado na arquitetura descrita na Subseção 3.1. As informações são obtidas por um gerente através de um agente para dispositivos de rádio cognitivo, desenvolvido através do *Simple Network Management Protocol* (SNMP) [IETF 1988]. Este protocolo provê um meio de comunicação simples e eficiente entre os dispositivos gerenciados e o sistema de gerenciamento. As informações no agente são organizadas em uma MIB, que possui objetos baseados na MIB IEEE 802.22, de acordo com a representação da Figura 1. As informações do processo de sensoriamento espectral são obtidas através da simulação de um detector de energia no *software* Matlab.

O sistema de gerenciamento precisa ser configurado apenas durante sua inicialização, onde são informados os valores de ajuste dos parâmetros de configuração. Esses valores são utilizados pelas regras de configuração, respeitando os limiares descritos nestas regras. Para validação do sistema de gerenciamento, o protótipo desenvolvido apresenta atualmente duas regras de configuração, voltadas para o sensoriamento espectral: (i) baseada nos limiares da vazão do dispositivo e (ii) baseada nos limiares do nível de confiança para identificação de ocupação dos canais de rádio frequência. Entretanto, novas regras de configuração podem ser adicionadas facilmente. Após a coleta das informações, o sistema de gerenciamento aciona o módulo das regras de configuração. Esse módulo calcula qual a melhor configuração para os parâmetros do agente de acordo com os limiares previamente definidos.

Os limiares estabelecidos consideram os valores máximos e mínimos para os parâmetros analisados. Por exemplo, de acordo com a norma IEEE 802.22, o nível mínimo de confiança no sinal recebido para identificação de canal ocupado é de 67% [IEEE 802.22 2011]. Assim, existe uma relação direta entre a vazão obtida na transmissão e o nível de confiança para a percepção se determinado canal está livre ou não. Esse nível de confiança está relacionado com o tamanho da janela de sensoriamento. Por exemplo, ao diminuir a janela de sensoriamento para maximizar a vazão, deve-se considerar que essa diminuição afetará a confiança do resultado do sensoriamento espectral.

A partir das especificações do protótipo, na próxima seção são descritos os detalhes sobre a modelagem do ambiente de simulação para o sistema de gerenciamento. Além disso, são apresentados e discutidos os resultados obtidos através das simulações realizadas dentro do ambiente definido.

4. Avaliação e Resultados

Esta seção apresenta os detalhes sobre as simulações realizadas para validar o funcionamento do sistema. Além disso, os resultados obtidos são apresentados e discutidos. A Subseção 4.1 apresenta os detalhes da modelagem do sistema de gerenciamento para rádio cognitivo. Na Subseção 4.2 são discutidos os resultados obtidos pela simulação, através da análise de desempenho do sistema.

4.1. Modelagem do Sistema de Gerenciamento

As simulações realizadas neste trabalho têm como objetivo principal avaliar o funcionamento do sistema de gerenciamento proposto, utilizando duas regras de configuração. A primeira regra visa maximizar a vazão, ajustando a janela de sensoriamento em relação ao nível de confiança para detectar canais ocupados. Por outro lado, a segunda regra objetiva maximizar o nível de confiança do sensoriamento espectral. Sendo assim, nesta seção, é apresentada a modelagem do sistema de gerenciamento utilizada para a avaliação de desempenho.

Inicialmente, é necessário modelar um sinal de transmissão sem fio para ser utilizado na simulação como prova de conceito. Dessa forma, é importante ressaltar que qualquer tipo de sinal pode ser utilizado no sistema proposto. A Equação 1 apresenta o sinal modelado $S(t)$, de acordo com a definição de Liang *et al.* [Liang *et al.* 2008].

$$S(t) = \cos\left(2\pi \int_{0}^{t} [f_c + f_{\Delta} s(n)] dn\right) \tag{1}$$

onde, t é um instante de tempo, f_c é a frequência portadora, f_{Δ} é o desvio de frequência e $s(n)$ é o sinal de origem. A detecção de $S(t)$ pelo receptor é realizada de acordo com uma taxa de amostragem N, definida pela a Equação 2.

$$N = \tau.2.f_{max} \tag{2}$$

onde, τ representa a duração do sensoriamento espectral, em milissegundos, e f_{max} indica a frequência máxima do sinal amostrado. Considerando a amostragem realizada, a capacidade de transmissão de um determinado canal $C(t)$, em Mbit/s, pode ser obtida através do teorema de Shannon:

$$C(t) = B.log_2\left(1 + \frac{S(t)}{R(t)}\right) \tag{3}$$

onde B representa a largura de banda do canal e $\frac{S(t)}{R(t)}$ a relação entre o nível do sinal e do ruído, medida em um instante de tempo t, sendo que $R(t) \neq 0$. Essa relação é obtida com base no indicador de potência do sinal recebido (*Received Signal Strength Indication* - RSSI). Entretanto, para a obtenção da taxa efetiva de transmissão da rede de rádios cognitivos, é necessário considerar a duração da janela sensoriamento espectral (τ), para identificar se um determinado canal está livre ou ocupado. Durante o tempo de sensoriamento, não é possível realizar transmissões, portanto a duração de uma transmissão de dados (T_D), em milissegundos, pode ser definida como:

$$T_D = Q_D - \tau \tag{4}$$

onde Q_D é a duração de um quadro, em milissegundos e $Q_D > \tau$. Dessa forma, a capacidade de transmissão de dados em cada quadro (C_Q) pode ser obtida através da Equação 5.

$$C_Q = C(t).T_D \tag{5}$$

Assim, a capacidade total de transmissão (C'), em Mbit/s, descontando-se a janela de sensoriamento, é definida na Equação 6. Através desta equação, o sistema de gerenciamento proposto neste trabalho analisa a capacidade de transmissão do dispositivo em relação ao tempo da janela de sensoriamento.

$$C' = C_Q.\frac{10^3}{Q_D}, \quad (Q_D > 0) \tag{6}$$

Após a definição da capacidade total de transmissão, é necessário relacioná-la com a confiança na detecção do sinal. Essa confiança é baseada no nível de similaridade da energia do sinal antes da transmissão e após a amostragem realizada pelo dispositivo receptor do sinal. O nível de energia médio do sinal amostrado é calculado através da Equação 7.

$$M(y) = \frac{1}{N}\sum_{n=1}^{N} |y(n)|^2 \tag{7}$$

onde, $y(n) = s(n) + u(n)$ é o somatório do sinal transmitido ($s(n)$) com os ruídos e interferências que o compõe ($u(n)$). Como o ruído analisado no sinal recebido é do tipo *Additive White Gaussian Noise* (AWGN), a relação não resultará em um valor de sinal recebido muito acima do transmitido. Dessa forma, considerando-se que $M(y)'$ é a energia medida antes da transmissão e $M(y)''$ é a energia medida após, tem-se o nível de confiança (ς), dado pela Equação 8.

$$\varsigma = 1 - \left|1 - \frac{M(y)''}{M(y)'}\right|, \quad (M(y)' > 0) \tag{8}$$

No sistema de gerenciamento proposto neste artigo, τ pode ser alterado automaticamente. O valor desta alteração automática é chamado de ajuste da janela de sensoriamento e o intervalo entre cada ajuste é definido como período de *polling*. A relação entre a vazão e o nível de confiança do sensoriamento pode ser obtida pela normalização dos valores obtidos através das regras de configuração. Esses valores são representados pela tupla $v = [C', \varsigma]$, onde $v \in V$ e V é um conjunto de tuplas. A primeira regra prioriza maximizar a vazão, onde o parâmetro secundário é o nível de confiança. Dessa forma, a tupla resultante da regra da vazão (RC') é definida por:

$$RC' = V_k \rightarrow \forall i \neq k, V_{C',k} > V_{C',i} \quad \wedge \quad V_{\varsigma,k} > \varsigma_{\min} \tag{9}$$

onde, $\varsigma_{\min}$ é o menor nível de confiança tolerado pelo sistema de gerenciamento. Por outro lado, a segunda regra prioriza obter o maior nível de confiança, deixando a vazão como parâmetro secundário. Relacionando-se os valores obtidos em RC', pode-se obter uma taxa relativa à regra da vazão ($R'C'$):

$$R'C' = \frac{RC'_{C'}}{RC'_{\varsigma}} \tag{10}$$

De forma análoga à regra de priorização da vazão, pode-se obter a tupla ótima para a regra de priorização da confiança ($R\varsigma$). Tem-se que $R\varsigma$ é representada pela Equação 11:

$$R\varsigma = V_k \rightarrow \forall i \neq k, V_{\varsigma,k} > V_{\varsigma,i} \tag{11}$$

Baseando-se novamente na tupla v, pode-se calcular a taxa relativa à regra do nível de confiança ($R'\varsigma$). Essa taxa é definida pela Equação 12:

$$R'\varsigma = \frac{R\varsigma_{\varsigma}}{R\varsigma_{C'}} \tag{12}$$

Finalmente, pode-se encontrar a melhor relação entre $R'C'$ e $R'\varsigma$. Essa relação é obtida pelo valor mínimo da diferença entre o resultado das duas regras de configuração, como pode ser observado na Equação 13. Dessa forma, a menor discrepância entre os valores da tupla v representa a melhor configuração possível entre a vazão e o nível de confiança do sensoriamento.

$$R = \min\{R'C', R'\varsigma\} \tag{13}$$

A modelagem do sistema descrita nesta subseção é validada em um ambiente de simulação. Esse ambiente e a avaliação de desempenho do sistema de gerenciamento são apresentados e discutidos na próxima subseção.

4.2. Avaliação de Desempenho

A avaliação de desempenho do sistema de gerenciamento proposto foi realizada em um ambiente de simulação. A Tabela 1 apresenta os principais parâmetros do sistema, bem como os limiares utilizados nas regras de configuração. Os parâmetros e limiares utilizados neste trabalho são baseados na norma IEEE 802.22, que define o padrão para redes sem fio regionais (*Wireless Regional Networks* - WRAN) [IEEE 802.22 2011]. Estes parâmetros e limiares também caracterizam as suposições iniciais para a operação do sistema de gerenciamento. A janela de sensoriamento foi configurada entre 50μs e 5ms. O valor mínimo do nível de confiança para a detecção de um canal ocupado foi de 67$\%$, considerando um ruído AWGN de 20dB. A duração do quadro foi definida em 10^{-3}s, o ajuste da janela de sensoriamento variando de 10^{-6} até 10^{-5}s e o período de *polling* foi ajustado entre 10^{-1}s e 1s. Além disso, os experimentos foram repetidos até alcançar um intervalo de confiança de 95$\%$.

Parâmetro	**Valor**
Janela de sensoriamento	$[5.10^{-5}$:$5.10^{-3}]$s
Nível de confiança mínimo	67$\%$
Ruído	AWGN
Relação Sinal/Ruído	20dB
Duração do Quadro	10^{-3}s
Ajuste da janela de sensoriamento	$[10^{-6}$:$10^{-5}]$s
Período de *Polling*	$[10^{-1}$:1]s
Intervalo de confiança	95$\%$

Tabela 1. Limiares considerados para simulação

O sistema de gerenciamento foi analisado considerando quatro abordagens: (i) o comportamento da tupla v em função do tamanho da janela de sensoriamento, (ii) regras de configuração em relação ao ajuste da janela de sensoriamento, (iii) sobrecarga do tráfego SNMP para o ajuste da janela de sensoriamento e (iv) o tempo de convergência entre as regras de configuração, considerando o período de *polling*.

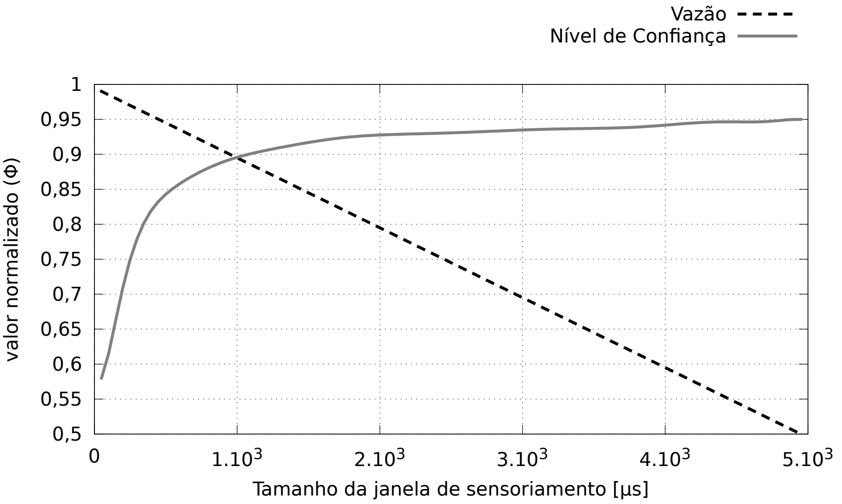


Figura 3. Relação da vazão e do nível de confiança

A primeira análise refere-se ao comportamento da tupla v em função do tamanho da janela de sensoriamento e da relação normalizada entre o percentual da vazão e do nível de confiança. Esta relação normalizada é definida como Φ. Na Figura 3, observa-se a tendência dos valores normalizados da vazão, onde a medida em que o tamanho da janela de sensoriamento aumenta, a vazão máxima obtida é reduzida linearmente e os valores normalizados do nível de confiança apresentam um crescimento logarítmico. De acordo com as tendências do comportamento da tupla v, pode-se encontrar o ponto de inflexão entre as curvas de vazão e nível de confiança. Esse ponto indica o valor da janela de sensoriamento onde a vazão e o nível de confiança possuem a menor variação entre si. O sistema de gerenciamento proposto para o sensoriamento espectral objetiva configurar automaticamente a janela de sensoriamento, segundo as duas regras de configuração.

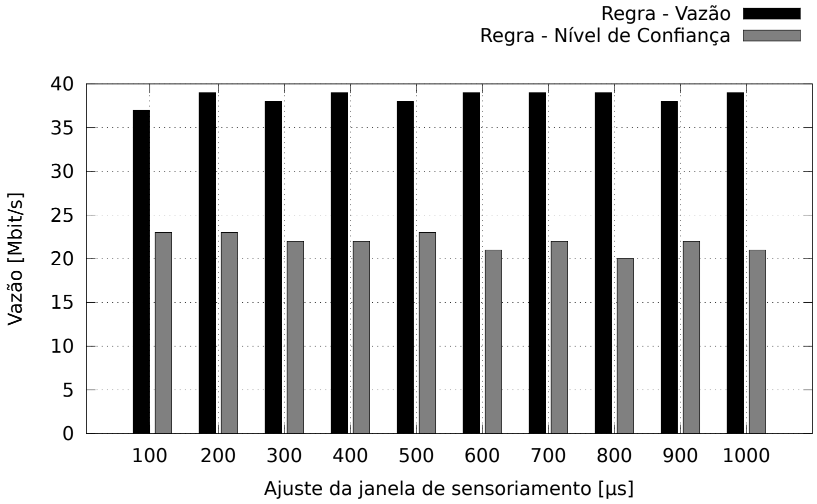


Figura 4. Vazão obtida para regras de confiança

A segunda avaliação de desempenho analisa as regras de configuração em relação ao ajuste da janela de sensoriamento. Essa analise é realizada em duas etapas, primeiramente considerando a vazão e posteriormente o nível de confiança. Na Figura 4 pode-se observar a vazão em detrimento ao ajuste da janela de sensoriamento. Nessa análise, pode-se confirmar que a utilização da regra da vazão aumenta a taxa de transmissão de dados em relação a regra do nível de confiança. No cenário analisado, esse aumento é em média 16 Mbit/s. Outra possível análise está relacionada ao ajuste da janela de sensoriamento. Diferentemente do esperado, esse ajuste não afeta significativamente a vazão do dispositivo gerenciado. Esse comportamento ocorre devido ao tamanho da janela de sensoriamento final ser semelhante para os diversos ajustes. A diferença está na quantidade de mensagens SNMP trocadas para configurar o tamanho a janela de sensoriamento.

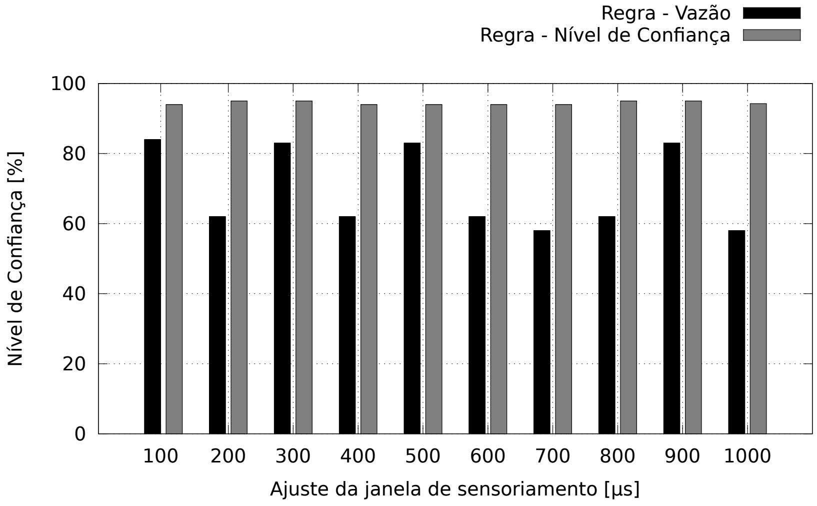


Figura 5. Nível de confiança obtido para regras de confiança

Na Figura 5, pode-se observar o nível de confiança em relação ao ajuste da janela de sensoriamento para as regras de configuração. Percebe-se que o percentual de confiança é de aproximadamente de 95%, quando utiliza-se a regra do nível de confiança. Além disso, ao utilizar essa regra, o percentual de confiança é similar para todos os ajustes da janela de sensoriamento, ou seja, existe uma convergência em torno de uma confiabilidade máxima. Por outro lado, pode-se observar que existe uma variabilidade no percentual do nível de confiança para a regra da vazão. Considerando-se essa regra percebe-se uma variação de aproximadamente 23% no nível de confiança, entre os diversos ajustes da janela de sensoriamento. Essa variabilidade é decorrente do crescimento logarítmico do nível de confiança, como apresentado na Figura 3, onde o nível de confiança para uma janela de sensoriamento inferior a $1.10^3 \mu$s possui um crescimento acentuado. Isto é, para valores de janela de sensoriamento próximos de $1.10^3 \mu$s, a variação do nível de confiança é maior.

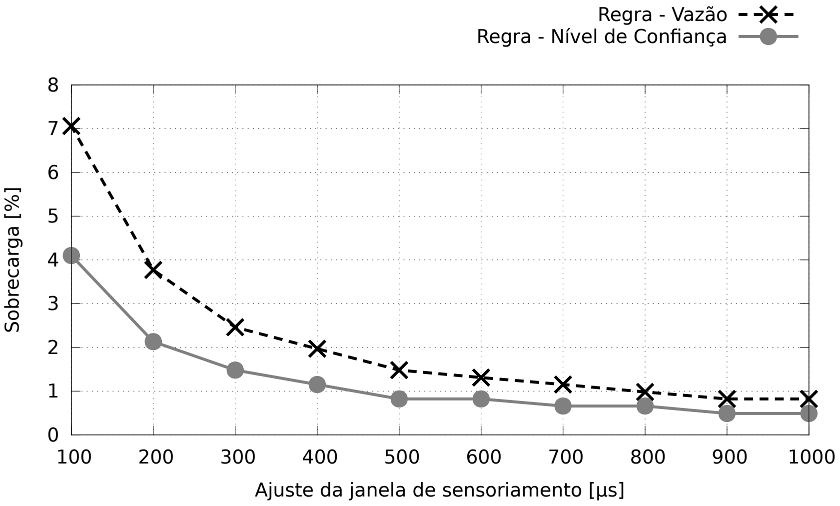


Figura 6. Sobrecarga do tráfego SNMP

A terceira análise deste artigo investiga a sobrecarga do tráfego SNMP gerada pelo sistema de gerenciamento e do dispositivo de rádio cognitivo. O cálculo da sobrecarga baseia-se na quantidade de pacotes SNMP em relação a capacidade máxima no canal de transmissão ($C(t)$), em Mbit/s, definido pela Equação 3. A Figura 6 apresenta o percentual da sobrecarga em função do ajuste da janela de sensoriamento para as regras de configuração. A investigação confirma que a sobrecarga de tráfego SNMP decresce à medida em que o valor do ajuste da janela de sensoriamento aumenta. Esta redução na sobrecarga ocorre pois, quanto maior o valor do ajuste na janela de sensoriamento, menos tráfego SNMP é necessário para alcançar o tamanho final da janela de sensoriamento. Outra análise que pode-se realizar está relacionada as diferenças de sobrecarga em relação as regras de configuração. A sobrecarga para a regra do nível de confiança apresenta-se menor do que a sobrecarga para a regra da vazão. Esse comportamento é justificado devido ao nível de confiança apresentar um crescimento logarítmico, como apresentado na Figura 3. Dessa forma, é necessário um menor ajuste total da janela de sensoriamento para alcançar o tamanho da janela de sensoriamento desejada.

A quarta e última análise realizada neste trabalho tem como objetivo encontrar a relação ideal entre a máxima vazão e o melhor nível de confiança. Nesta análise considera-se o período de *polling*, isto é o intervalo entre cada ajuste da janela de sensoriamento. Essa relação foi definida segundo a formulação descrita na Subseção 4.1,

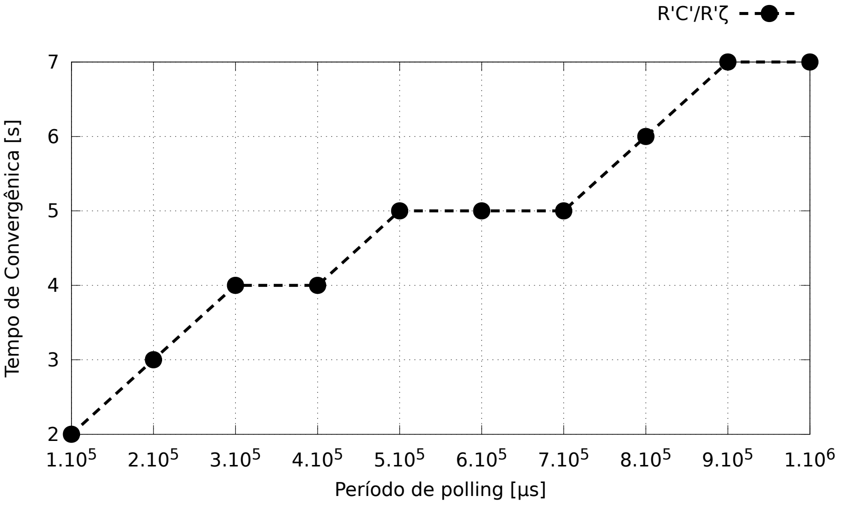


Figura 7. Tempo de convergência em relação as tuplas $R'C'$ e $R'\varsigma$

onde matematicamente pode-se chegar ao ajuste ideal para a janela de sensoriamento.

Considerando a relação entre as tuplas RC' e $R\varsigma$, o ajuste da janela de sensoriamento calculado para tupla RC' é de 900μs, já para a tupla $R\varsigma$ o ajuste é de 500μs. A Figura 7 apresenta os tempos de convergência considerando o período de *polling* para a melhor relação entre as tuplas RC' e $R\varsigma$. Além disso, pode-se observar o aumento no tempo de convergência quando o período de *polling* também aumenta. Este comportamento ocorre devido ao sistema de gerenciamento esperar o período de *polling* para realizar um novo ajuste na janela de sensoriamento.

5. Conclusões e Trabalhos Futuros

A proposta deste artigo apresenta um sistema de gerenciamento para radio cognitivo capaz de atuar com a dinamicidade das configurações de dispositivos cognitivos. Para tando, foram estudados os objetos da MIB IEEE 802.22 referentes ao processo de sensoriamento espectral. O sistema proposto mostrou-se capaz de melhorar o desempenho da transmissão de informações, ajustando os valores de configurações dos dispositivos de acordo com regras de configuração. Para prova de conceito, foi elaborado um protótipo, o qual ajusta o tamanho da janela de sensoriamento utilizando regras para maximização da vazão ou do nível de confiança. Acredita-se que o protótipo é capaz de operar de maneira satisfatória também em ambientes reais, uma vez que os resultados obtidos por simulação refletem em parte, as condições obtidas em campo.

Futuramente, pretende-se estender o estudo dos objetos que compõe a MIB IEEE 802.22, analisando como os outros parâmetros de configuração podem ser utilizados para melhorar o desempenho da operação da rede. Com isso, será possível desenvolver novas regras de configuração, baseando-se em um conjunto maior de parâmetros. Além disso, pretende-se aprimorar o sistema proposto para que o mesmo possa atuar no escopo Global de redes de rádios cognitivos. Para tanto, o sistema deverá considerar como as decisões tomadas dentro de uma rede afetam a operação de redes adjacentes. Pode-se ainda aumentar o nível de abstração na descrição das regras de configuração, através da descrição destas regras na forma de políticas.

Agradecimentos

Os autores gostariam de agradecer a Digitel S.A., Fundação de Apoio a Pesquisa do Estado do Rio Grande do Sul (FAPERGS), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Coordenação de Aperfeiçoamento de Nível Superior (CAPES) pelo apoio financeiro.

Referências

Akyildiz, I., Lee, W.-Y., Vuran, M., e Mohanty, S. (2008). A survey on spectrum management in cognitive radio networks. *IEEE Communications Magazine*, páginas 40–48.

Coutinho, P. S., da Silva, M. W. R., e de Rezende, J. F. (2012). Mecanismo de Handoff de Espectro para Rádios Cognitivos. *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 44–56.

IEEE 802.22 (2011). IEEE Standard for Information Technology - Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN) - Specific requirements Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands. *IEEE Std 802.22*, páginas 1–680.

IETF (1988). RFC 1067 - Simple Network Management Protocol. *Disponível em: http://www.ietf.org/rfc/rfc1067.txt*. Acesso em dezembro de 2012.

Khalid, L. e Anpalagan, A. (2010). Emerging cognitive radio technology: Principles, challenges and opportunities. *Computers & Electrical Engineering*, páginas 358–366.

Liang, Y.-C., Zeng, Y., Peh, E., e Hoang, A. T. (2008). Sensing-Throughput Tradeoff for Cognitive Radio Networks. *IEEE Transactions on Wireless Communications*, páginas 1326–1337.

Potier, P. e Qian, L. (2011). Network management of cognitive radio ad hoc networks. *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*, páginas 1–5.

Stavroulaki, V., Bantouna, A., Kritikou, Y., Tsagkaris, K., Demestichas, P., Blasco, P., Bader, F., Dohler, M., Denkovski, D., Atanasovski, V., Gavrilovska, L., e Moessner, K. (2012). Knowledge Management Toolbox: Machine Learning for Cognitive Radio Networks. *IEEE Vehicular Technology Magazine*, páginas 91–99.

Wang, C.-X., Chen, H.-H., Hong, X., e Guizani, M. (2008). Cognitive radio network management. *IEEE Vehicular Technology Magazine*, páginas 28–35.

Wang, J., Ghosh, M., e Challapali, K. (2011). Emerging cognitive radio applications: A survey. *IEEE Communications Magazine*, páginas 74–81.

Wang, S., Xie, L., Liu, H., Zhang, B., e Zhao, H. (2010). Acra: An autonomic and expandable architecture for cognitive radio nodes. *International Conference on Wireless Communications and Signal Processing (WCSP)*, páginas 1–5.

Yucek, T. e Arslan, H. (2009). A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys Tutorials*, páginas 116–130.

Zhao, Q. e Sadler, B. (2007). A Survey of Dynamic Spectrum Access. *IEEE Signal Processing Magazine*, páginas 79–89.