NTNU

Norwegian University of
Science and Technology

Cognitive Radio Network System Demonstrator

Nemanja Trećakov

Master of Science in Electronics
Submission date: September 2011
Supervisor: Torbjørn Ekman, IET
Co-supervisor: Kimmo Kansanen, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem description

"Resource and interference control within cognitive networks is based on sensing in the network. The student will write a proposal for a cognitive network demonstrator using the radio platform USRP2 and implement the proposed demonstrator. The proposal is a full description of the demonstrator. The demonstrator is used to evaluate the performance of a cognitive radio network.

The demonstrator will serve as the basis for further development of CR on the USRP2 at NTNU. This work is directly linked to the activity in EU FP7 project SENDORA (Sensor network aided cognitive radio)."

Assignment given: 18. April 2011
Supervisor: Associate Professor Torbjörn Ekman
Co-supervisor: Associate Professor Kimmo Kansanen

Abstract

The frequency spectrum is presently poorly utilized and lies idle for the most of the time. In order to utilize the spectrum more efficiently, Dynamic Spectrum Access (DSA) should be used. A common approach to this thought is hierarchical DSA, where the license-paying, or the primary users, have the priority in using and accessing the network. The rest of the potential users, or the secondary ones, are accessing and using the resource in an opportunistic manner, "borrowing" it as long as the resource is sensed as idle. This kind of resource access implies the use of cognitive techniques for spectrum sensing in order to discover the available resources and avoid collisions. However, exactly these two are the main issues regarding the cognitive networks: sensing method and interference control.

Many theoretical studies indicate the increase in efficiency with use of DSA. However, the few successfully performed experiments which evaluated the real performance have been carried out with expensive and complex-to-use radio platforms, or those employing low bandwidth resulting in low throughput.

In this document, a proposal for a cognitive radio network system demonstrator is described in detail. The demonstrator is based on IEEE802.11g standard employing OFDM, and is supposed to be carried out with use of 9 highly flexible USRP2 software defined radio (SDR) platforms and its complementary GNU Radio software.

The goal with the demonstrator is to show that the introduction of the secondary user in the network will introduce no or negligible degradation to the primary network.

A common problem regarding SDR platforms is the latency, introduced due to the fact that the vast of signal processing is performed in the host processor. To address this issue, the split-function architecture is chosen to be used, implementing the most time-critical functions in the FPGA on the platform.

The projects resulted in the full description of the demonstrator for the cognitive radio, but unfortunately without the implementation of the same due to the complexity of the work and limited amount of time. However, the project resulted in a set of advices and manuals providing the building stones on the way to the goal of implementing the cognitive radio on the USRP2.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science (MSc) in Engineering at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU).

The thesis' scope corresponds to 30 ECTS points, and 20 weeks have been assigned for its completion.

The work is directly linked to the EU FP7 project SENDORA (**SE**nsor **N**etwork for **D**ynamic and c**O**gnitive **R**adio **A**ccess), which NTNU is taking part in.

Acknowledgments

Firstly, I would like to thank Associate Professor Torbjörn Ekman and Associate Professor Kimmo Kansanen, my supervisor and co-supervisor respectively, for support, instructive discussions and advices during the project. I would also like to thank Jonny Nordheim, my mentor in Telenor, for advice in choosing the theme for my master thesis.

Moreover, I would like to thank Bruhtesfa Ebrahim Godana and Ruben Undheim for their start-up help with the GNU Radio software; Bjarne Drotninghaug for introducing me to the FPGA design and Xilinx Design Suite; and Morten Flå for his advices related to GNU/Linux OS. Furthermore, I would like to thank Javier Gozálvez from Uwicore Research Laboratory, University Miguel Hernández, Spain for kindly providing their implementation code of 802.11 MAC in GNU Radio, and George Nychis for discussion and advices regarding the practical implementation of my project. I can not end this section without thanking to all of those who have responded my questions on the GNU Radio mailing list, especially to Marcus D. Leech and Josh Blum for their tireless effort to explain and help with the emerged problems.

Finally, I would like to thank my family for their continuous support in my endeavors.

Trondheim, September 2011

Nemanja Trećakov

Contents

List of figures

List of tables

Abbreviations

DSA	Dynamic Spectrum Access
SDR	Software Defined Radio
CR	Cognitive Radio
SUU	Secondary Unlicensed User
PLT	Primary Licensed Technology
FCC	Federal Communications Commission
USRP	Universal Software Radio Peripheral
WSN	Wireless Sensor Network
WiFi	Wireless Fidelity
ADC	Analogue to Digital Converter
DAC	Digital to Analogue Converter
DSP	Digital Signal Processor
FPGA	Field-Programmable Gate Array
GPC	General Purpose Computer
PHY	Physical layer
MAC	Medium Access Control
OSI	Open System Interconnection basic reference model
HW	Hardware
SW	Software
TX	Transmitter/Transmission
RX	Receiver/Reception
WARP	Wireless Open - Access Research Platform
BEE2	Berkley Emulation Engine
KUAR	Kansas University Agile Radio
MARS	Maynooth Adaptable Radio System
WinC2R	The WINLAB Network Centric Cognitive Radio Hardware Platform
COBRA	COgnitive Baseband Radio
OSSIE	Open Source SCA Implementation – Embedded
IRiS	Implementing Radio in Software
ORBIT	Open Access Research TestBed
WNT	Wireless Network Testbed

CORNET	COgnitive Radio NEtwork Testbed
AFH	Adaptive Frequency Hopping
ISM	Industrial, Scientific and Medical band
GENI	The Global Environment for Network Innovations
DDC	Digital Down Converter
CIC	Cascaded Integrate-Comb filter
DUC	Digital Up-Conversion
PLL	Phase Locked Loop
VCO	Voltage Controlled Oscillator
SWIG	Simplified Wrapper Interface Generator
RRM	Radio Resource Management
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
TDD	Time Division Duplex
FDD	Frequency Division Duplex
DSSS	Direct Sequence Spread Spectrum
OFDM	Orthogonal Frequency Division Multiplexing
CCK	Complementary Code Keying
MSDU	MAC Service Data Unit
MPDU	MAC Protocol Data Unit
PSDU	Physical layer Service Data Unit
PPDU	Physical layer Protocol Data Unit
AP	Access Point
STA	Station
IFS	Inter Frame Spaces
PCF	Point Coordination Function
DCF	Distributed Coordination Function
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
RTS	Request-to-Send
CTS	Clear-to-Send
NAV	Network Allocation Vector

1. Introduction

1.1 Background

The frequency spectrum has become overcrowded in the modern time. The reason for this is the explosion in number of wireless systems and the historical legacy of allocating frequency bands with exclusive licenses to each new system, continuously making the spectrum tighter and denser. As the spectrum is a limited resource, the present allocation policy together with the high inquiry constantly contributes to higher prices. However, the measurements of the actual spectrum usage, e.g. [1-5] are indicating very low efficiency, with the time-average spectrum occupancy generally much less than 20%. This implies that the biggest part of our precious spectrum lies idle at any time, i.e. we have spectrum holes. The spectrum holes of longer nature can be observed when frequencies allocated for broadcasting are not used locally in some areas; these particular holes are known as white spaces.

To utilize the spectrum more efficiently, the frequencies should be allocated according to users' needs and spectrum availability, i.e. dynamically, as opposite to the current static spectrum management. The Dynamic Spectrum Access (DSA) and its different models are thoroughly described in [6]. However, this approach is not a novelty. In 1998, Mitola introduced the Cognitive Radio (CR) conception [7] as a flexible radio that is fully aware of and sensible to the changes in its surroundings, and behaves (transmits) according to these, i.e. by using the unused resources in the signal-space on an appropriate way. Moreover, this flexible radio should also be able to learn from previous events and use that knowledge with the current inputs for optimizing its own, and with no or insignificant degradation of others' performance.

However, the required flexibility sets new demands to the radio equipment which are extremely cumbersome to meet with the traditional hardware radios. In order to be capable of exploiting all the possibilities in the signal-space, the radio must be able to almost instantly adjust to a variety of frequency bands, modulation types, and the last, but not the least, to adjust its sending power. This is provided with the Software Defined Radios (SDR), where the vast of the signal processing is performed and defined in the software. The SDR paradigm was introduced in 1992 also by Mitola [8], and in fact, the research in this field resulted in the CR concept.

Indeed, the introduction of the digital signal processing made a turnover in the radio implementation, providing both extremely flexibility and ease in the implementation of new services, and consequently lowering costs. Hence, SDRs represents the basis for development of cognitive radios.

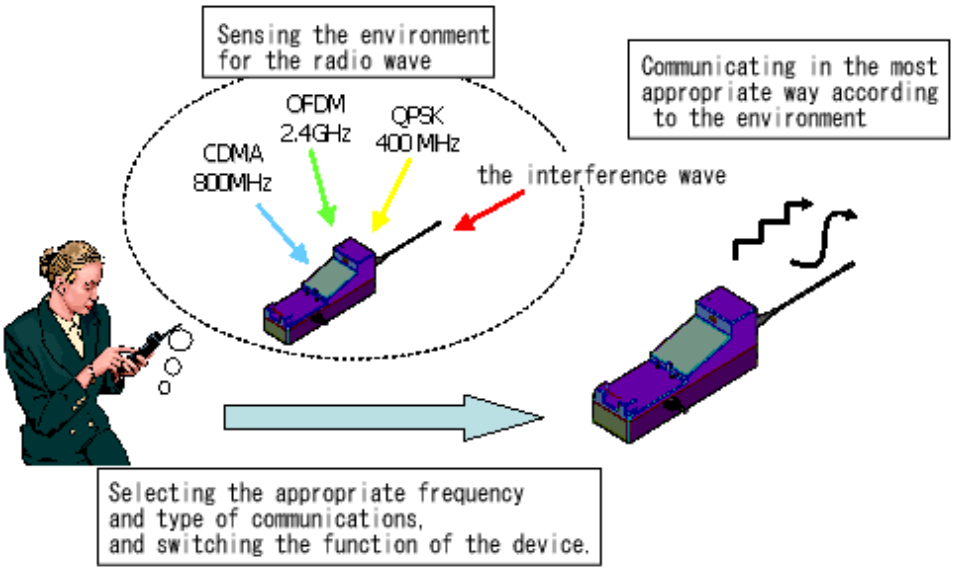


Figure 1.The principle of the Cognitive Radio function. Figure based on [9].

The possibility of constructing a CR radio consequently led to the thought of deploying a CR network, a network which could exist in addition to the original, licensed network, and would effort to exploit the possibilities in the signal-space of the primary network in an opportunistic manner. The concept of the CR network function can be described as follows [10]: The Secondary Unlicensed User (SUU) senses the spectrum it wishes to use, and characterizes the presence of the Primary Licensed Technology (PLT) users, if any present. Based on that information and eventual regulatory policies, the SUU identifies communication opportunities ("holes") in signal-space, i.e. in frequency, time or even code - and transmits using those opportunities in a manner that limits the interference perceived by the PLT. In summary, the CR's objective is the hierarchical DSA with no/negligible interference introduced for its surroundings, an access method called Opportunistic Spectrum Access.

However, the design of a fully cognitive radio or a network, i.e. which would take into account every possible parameter observable by a wireless node or a

network, would be extremely demanding. The most common approach for a CR operation is to detect and exploit the frequency bands which have been idle for some period of time, and it is the only approach considered in this thesis.

In order to interact with the primary network according to the above described operation, each user of the secondary network must have the ability to:

- Determine available points of the spectrum
- Select the best available channel
- Coordinate access with the other secondary users
- Vacate the channel as soon as the licensed user is detected

Indeed, these four listed required features are the main challenges in CR networks, respectively spectrum sensing, decision, sharing and mobility [11]. However, from the standpoint of the primary user, the first and the last of the issues mentioned are the most important ones; namely reliable sensing and quick adjustment, both in order to introduce as least interference as possible to the primary network. Furthermore, in order to obtain sufficient information about the channel, the secondary user has to monitor the channel, which can be considered as a security issue. Therefore, the allowed interference limit and the security considerations should be clarified and regulated by policies.

Fortunately, the CR conception is not only addressed in the theoretical studies, e.g. [6, 10]. The biggest effort to encourage the practical development of CR has been made by the U.S.'s Federal Communications Commission (FCC), which in 2008 opened a part of the Digital Dividend, the unused broadcast television spectrum due to the transition to the digital television, allowing the unlicensed radio transmitters to operate as cognitive radios in accordance with certain policies [12]. Back home in Europe, the best effort to speed up the CR implementation is the SENDORA project [13], included in European Union Seventh Framework Programme (EU FP7), with the objective to develop innovative techniques for DSA based on use of the sensor networks for spectrum sensing.

In summary, the cognitive radio concept extremely increases the spectrum utilization and the number of efforts aiming to exploit this property grows continuously.

1.2. Current problem

Theoretical studies and simulations regarding the inclusion of cognition in the wireless networks, as DSA and CR, indicate spectrum efficiency increase from 40 to 100% per user [14]. However, the few successfully performed experiments which evaluated the real performance of CR, e.g. [15, 16, 17], have been carried out with expensive and complex-to-use radio platforms, or those employing low bandwidth resulting in low throughput.

The objective of this project is to propose a full description of a practical cognitive radio network demonstrator using the radio platform USRP2 and implement the proposed demonstrator. The project is based on the earlier directed study, and the vast of the material in this thesis are the revisited ideas.

The goal with the demonstrator is to evaluate the real performance of the cognitive radio network by employing relatively cheap and easy-to-use USRP2 radio platform with considerable throughput, showing that the introduction of the secondary user in the network introduces no or negligible degradation to the primary network. The eventual degradation of the primary network will be measured by means of key performance indicators as Bit Error Rate (BER) and Packet Loss (PL).

This project will serve as the basis for further development of CR on the USRP2 at NTNU. The work is directly linked to the EU FP7 project SENDORA (**SE**nsor **N**etwork for **D**ynamic and c**O**gnitive **R**adio **A**ccess), which is briefly presented in section 1.5.

1.3. Limitations

The number of available radio platforms USRP2 at the university to be used in our demonstrator is 9. Controlling software which will be used with these units is the GNU Radio, licence-free software complementary with USRP family. Moreover, RF-front-end available for each unit is XCVR2450, transceiver card with operating frequencies in ISM band, both at 2.4 and 5 GHz. The research area is limited to the indoor environment of the Norwegian University of Science and Technology (NTNU). Lastly, the resources assigned to the project in means of time and people are 20 weeks and one person. These limitations are summarized in the table 1.

Table 1.Limitations

Parameter	Limitation
SDR	USRP2
# of units	9
Daughter board	XCVR2450
Software	GNU Radio
Research environment	Indoor, NTNU
Time/No. of Persons	20 weeks/1 person

1.4. Structure of the report

The scope of the project includes a full description of a demonstrator for a cognitive radio network and the description of its implementation. It consists of 7 chapters, beginning with an overview of the content of the study. The following chapter 2 describes the basics behind the software defined radio, and give a brief overview of the SDR platforms and cognitive test beds built around the world. Further on, the chapter 3 introduces a detailed overview of USRP2 and GNU Radio, the two which will be used in the demonstrator development. The chapter 4 describes the theory behind the Radio Resource Management for cognitive networks. The following chapter 5 explains the choice of the primary technology, and gives a comprehensive demonstrator description. The implementation details including discussion are given in the chapter 6. Finally, I summarize the conclusions and give my suggestions for the future work in the chapter 7.

1.5 Related work

The USRP hardware, which will be used in to build the demonstrator, has been used for many different applications, both non- and commercial ones, including GPS receiver, GSM receiver, amateur radio transceiver, RFID reader, digital TV, WLAN node etc. Information about the past and the current projects can be found in [18] and [19]. However, only a few of these are directly relevant for this particular project, and a short description of these is included in the implementation discussion, chapter 6.

1.5.1 SENDORA

As the current project is directly related to Sendora project, it is suitable to give a brief overview of this project. The SENDORA project (**SE**nsor **N**etwork for **D**ynamic and c**O**gnitive **R**adio **A**ccess) is an EU seventh framework programme project that "develops innovative techniques based on sensor networks, that will support the coexistence of licensed and unlicensed wireless users in a same area" [13]. It is a small to medium-scale project led by European universities and commercial actors listed: Thales, Eurecom, NTNU, Telenor, KTH, TKK, Universities of Rome, Valencia and Linköping.

The innovative and original concept introduced in Sendora, is the use of sensor network to address the problem of reliable sensing of the spectrum holes. This problem is explained in detail in the chapter 4.4. Furthermore, project's objectives are to identify and analyze the business scenarios of the Wireless Sensor Network (WSN) aided CR technology, define and simulate strategies for the WSN aided opportunistic access and dynamic resource allocation, and at last but not the least to design a flexible architecture and demonstrate the concept in practice. This project started in January 2008 with planed duration of 3 years, and the work is structured in 8 work-packages. All the information about the project, published project deliverables and reported research papers and can be found in [13].

The work-package 7 (WP7) in Sendora project addresses system integration and demonstration. The more important of two real-time demonstration scenarios which are to be practically tested is summarized as follows: all of the sensors in the sensor network will report their perception of the channel use to a Fusion centre, which, according to available information about the channel, informs the secondary network about the available spectrum for opportunistic use. If transmissions by the primaries are sensed, the Fusion centre informs the secondary nodes to adapt its transmission to avoid introducing harmful interference to the primary network [20]. Moreover, cables are used to connect the sensor network nodes and secondary nodes with Fusion centre in this demonstrator. WiFi standard is chosen for the primary network, with the Orthogonal Frequency Division Multiplexing (OFDM) scheme in the PHY layer.

Software and hardware testbed used in this demonstrator is Eurecom OpenAirInterface platform. This family of platforms targets innovation in air-interface technologies through experimentation. To meet the requirements of the project, Eurecom introduced in 2009 new RF board "AgileRF" and baseband processing engine "Express MIMO" into its family, where PHY and Data layer are fully reconfigurable, also in real-time. Including powerful FPGAs, on-board RAM memory and MIMO, this platform offers up to 20MHz of bandwidth and is capable of compliance with LTE and WiMax standards in the frequency band from 200 MHz to 7.5 GHz. Moreover, the platform uses open source software tools, and is connected to a PC via PCI-express. Specifications of the platform can be found in [20]. For more information about the OpenAirInterface platform, see [21].

2. SDR & CR

This chapter gives a brief introduction to the Software Defined Radio (SDR), design trade-offs and different SDR platforms present on the market, as well as experimental Cognitive Networks (CR). An interested reader should refer to references in the text.

2.1 Basics of SDR

Software Defined Radio (SDR) is a radio communication system where a relatively large part of the signal processing is performed and defined in software. The main philosophy behind SDR is to use digital signal processing as close as possible to the antenna, both providing flexibility and lowering costs drastically. Block diagram in figure 2 illustrates the principle.

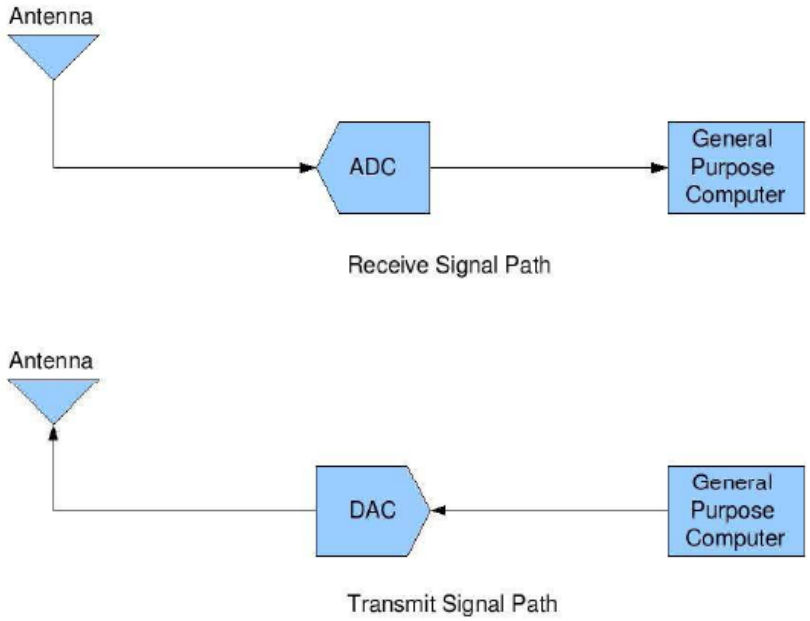


Figure 2.Block diagram of an ideal software-defined radio communication system [22, fig.1.1]

SDR aims to receive the signal, digitalize it and process it further in the software. Similar approach is used when transmitting. However, due to the limitations in sampling rate of ADCs and processing speed of currently available processors, such systems are still not realizable for high frequencies. In order to convert the signal to a

frequency that can be handled by ADCs, an RF section in the front-end is used. The intermediate-frequency signal is further forwarded to a digital signal processing unit which could be realized as a digital signal processor (DSP), field-programmable gate array (FPGA) or a general purpose computer (GPC). This unit mainly performs decimation and filtering, and sends the signal to further processing to the host processor. The similar approach applies for transmitting. This is shown in figure 3.

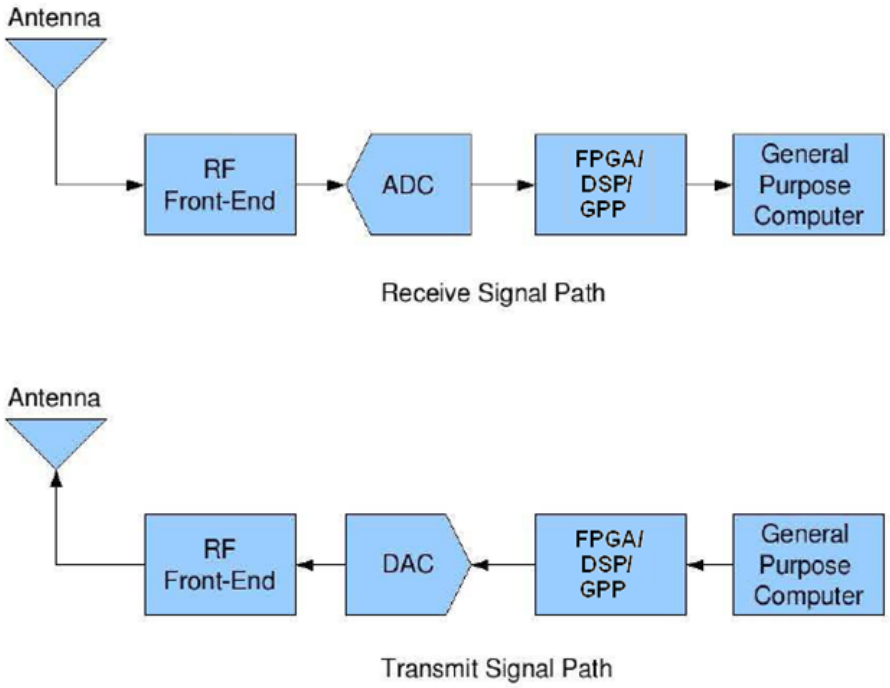


Figure 3.Block diagram of a currently realizable software-defined radio communication Figure based on [22, fig.1.2]

Depending on their architecture, SDRs can offer access to the lowest layers of the OSI model, providing the full control and flexibility to the user. This is shown in the combined figure 4, where the question marks depicted in the figure present the variation of the frontier between hardware and software in different manifests of SDR concept. Due to these attractive properties, SDR platforms represent the basis for development of cognitive radio platforms and are the building stones of cognitive networks.

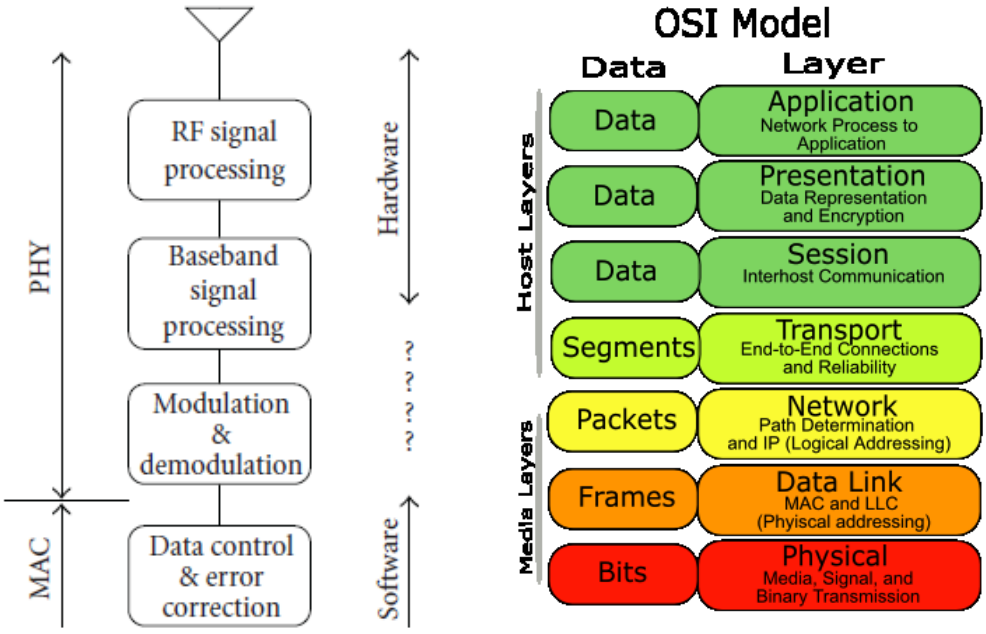


Figure 4.Partitioning between the software and hardware in an SDR, and comparison to the complete OSI model.[23, fig 1 and 24, respectively]

2.2 Design of SDR platforms

The demand for more flexible radio systems, especially from the military sector, was the major driving force for development of SDRs, even before the concept was named by Mitola in 1992 [8]. It resulted in many research programs not only by governments and military, but also by universities and commercial actors. An excellent and comprehensive overview of the historical development and research programs, together with challenges and opportunities involving SDR can be found in [25].

The important considerations within the design of an SDR platform are [23]:

- **Partition of Resources** – the choice of the division line between HW & SW
- **Frequency Flexibility** – the choice of frequency bands to support
- **Interference Management** – the choice of how to handle the "dynamic" filters and receivers
- **Transceiver necessity** – the choice of employing only Tx or Rx when sufficient

The choices made within the design directly influence the performance and the flexibility, but also the complexity and the cost of the platform. However, these properties usually do not go together and a trade-off has to be made according to the goal with the design. Through the years, a large number of different SDR platforms have been developed to support the individual research projects, and all these differ in the performance, flexibility, price and the ease of usage. The most known SDR platforms are USRP, WARP and BEE2, but the USRP family is definitely the most used one. Since USRP2 is used in this project, a detailed description will be given in chapter 3. In addition to these, other platforms as MARS, KUAR, OpenAirInterface, WinC2R and COBRA have also been designed.

Moreover, since the SDR platforms depend as much on software as on hardware, a development of dedicated software was also required. The most known software frameworks are GNU Radio, OSSIE, IRiS and WARPLab.

All of the above mentioned platforms are described and compared in detail in [23] and [26], both in their hardware and software. The full names of the SDRs and their software are omitted in order to provide easier reading, but can be found in the abbreviations table.

2.3 Cognitive Networks

The principle of cognition is being more and more included in the existing wireless standards. Besides the already mentioned conception of users with different priorities, cognition is also used with sensor networks, which have to operate in a same spectrum with high power devices. Sensor networks are covered more in chapter 4. Furthermore, cognition is applied in cellular networks, offering different users different types of service. Finally, the cognitive features are also employed in the shared, unlicensed spectra as ISM band to make the coexistence without interfering possible. One of these is the 802.11k standard [27], which is an update for a radio resource management, and includes channel load information, station statistic report and noise histogram report, all three used to increase the efficiency of the traffic distribution. Another standard using cognition is the Bluetooth standard, which employs Adaptive Frequency Hopping (AFH) to avoid interference in the crowded unlicensed 2.4 GHz band.

In order to introduce the possibility of practical on-air tests of new algorithms and techniques, a number of experimental wireless networks of different dimensions have been built around the world. The most known test wireless networks are ORBIT, WNT, Emulab and CORNET. Some of them can even be remotely accessed and controlled via internet. However, very few nodes in these networks are SDRs, limiting the research with CR concept only within higher layers. More information about these four test networks can be found in [28]. The full names of the test networks are omitted to provide easier reading, but are included in the abbreviations table.

Furthermore, a few global research networks have also been established, like GENI [29] and the PlanetLab [30], including over 1000 at more than 500 sites all over the world. These networks are consisted of academic, governmental and commercial actors interested in research of new network services, which "borrow" their wireless nodes to each other for performing large-scale tests. Even though these not necessarily include SDRs at the moment, it is a great opportunity for remote global tests in the future.

Although all these mentioned networks represent the opportunity for practical test in some degree, the real-time experimental validation of the theoretical studies of CR is still lagging behind.

3. USRP2 & GNU Radio

This chapter gives an introduction to the USRP family, and a thorough review of the USRP2 radio platform, both for hardware and the software. For more information, refer to [18] and [31].

3.1 Introduction to the USRP family

Certainly the most popular SDR family on the market is the Universal Software Radio Peripheral (USRP) family from Ettus co, which offers fair performance and relatively easy usage for fair price. The first member of the family introduced in 2004 that made the family famous, the USRP1, uses a small FPGA and USB to connect to a computer, which limits its' operating bandwidth. The direct descendent, the USRP2 introduced in 2008, uses a gigabit Ethernet for computer connection, and supports a bandwidth up to 25 MHz. It also includes more powerful FPGA. Since USRP2 is going to be used in this project, it is thoroughly described in the following subchapters. Recently, new members of the family have been introduced, which offer operating bandwidth up to 50MHz and additional processor to be able to operate as a stand-alone system.

The platforms are completely open-source, both in hardware and software, and the possibility of creations with low budget and minimum of effort is the true value of this family. This product family is used by military, commercial and academic actors, as well as amateurs around the world.

Figure 5.The front face of the USRP2 [32]

3.2 USRP2 Hardware

USRP2's architecture follows the pattern of the figure 3. The platform is comprised of a motherboard and additional RF-front-ends daughter boards which could be easily interchanged in accordance to the desired operational frequency band. The opened platform can be observed in figure 6.

Figure 6.Opened USRP2 platform.

3.2.1 Motherboard

The major components comprising the motherboard of the USRP2 platform are the FPGA and the ADC/DACs. In the most common operation, i.e. by using the common FPGA image, the high sampling rate signal processing which includes down & up conversion, decimation, interpolation and filtering are done in the FPGA. The rest of the signal processing as modulation/ demodulation, estimation and further processing are easily manageable in the processor of the general purpose computer. However, the powerful FPGA offers a possibility for a standalone operation, and in that case all signal processing is performed on the platform itself. Figure 7 shows the simplified block diagram of the USRP2 motherboard. From this figure the independent transmit and receive signal path can be observed. Even though the multiplexer and demultiplexer are included by default FPGA configuration, they are

currently not used and are meant for possible extensions. Ergo, current operation involves only one digital down converter (DDC) and one Cascaded Integrate-Comb (CIC) filter, both with programmable decimation and interpolation rates, respectively. Moreover, it can be observed that the digital up-conversion (DUC) and filtering with the half band filter (HBF) in the transmit signal path are in fact performed outside the FPGA. For more detailed information about the features of each block, see [18, 34].

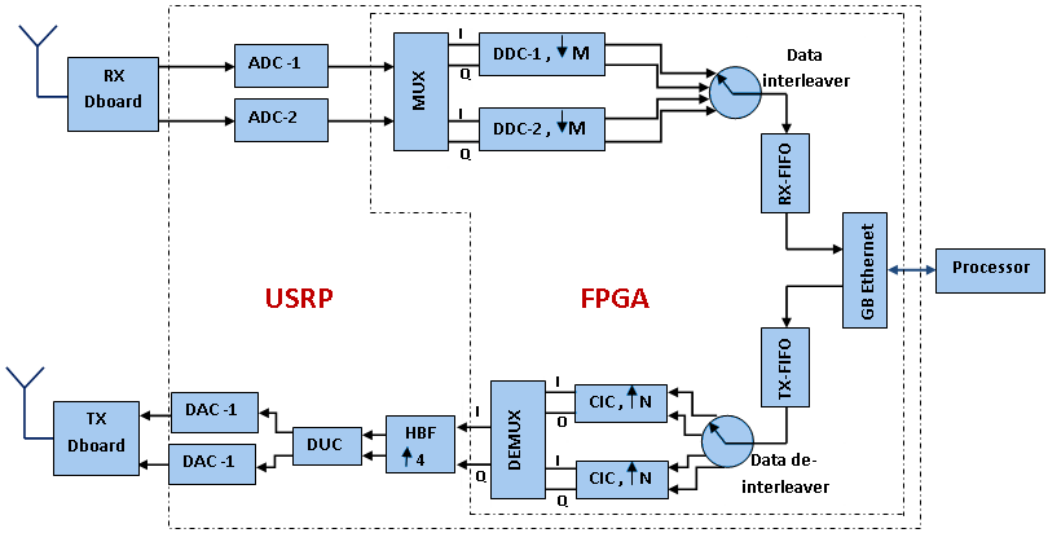


Figure 7.Simplified block diagram of the USRP2 motherboard.[33, fig.3]

The USRP2's motherboard with indicated components is shown in the figure 8, while motherboard's front interface is shown in the figure 9. The USRP2s' mother board is connected with the host PC via Gigabit Ethernet cable which allows processing of high bandwidth signals. Together with the ADCs sampling at 100 MS/s, this interface allows applications to simultaneously send 50 MHz of RF bandwidth in and out of the USRP2. However, a default FPGA configuration requires a decimation rate of at least 4, which results in maximum sampled signal bandwidth of 25 MHz. Furthermore, the USRP2 represents a complex sample of the signal using 32 bits, 16-bit I and 16-bit Q channel. However, in order to fully exploit the bandwidth of 25 MHz, a complex sample is represented by using 16 bits. No analogue filters what so ever are implemented on the motherboard, for maximum flexibility in frequency planning with daughter boards. In addition, the motherboard also includes a Programmable Gain Amplifier (PGA), both in the receive and in the transmit path, which can amplify the signal up to 20 dB before the ADC or after the DAC, respectively.

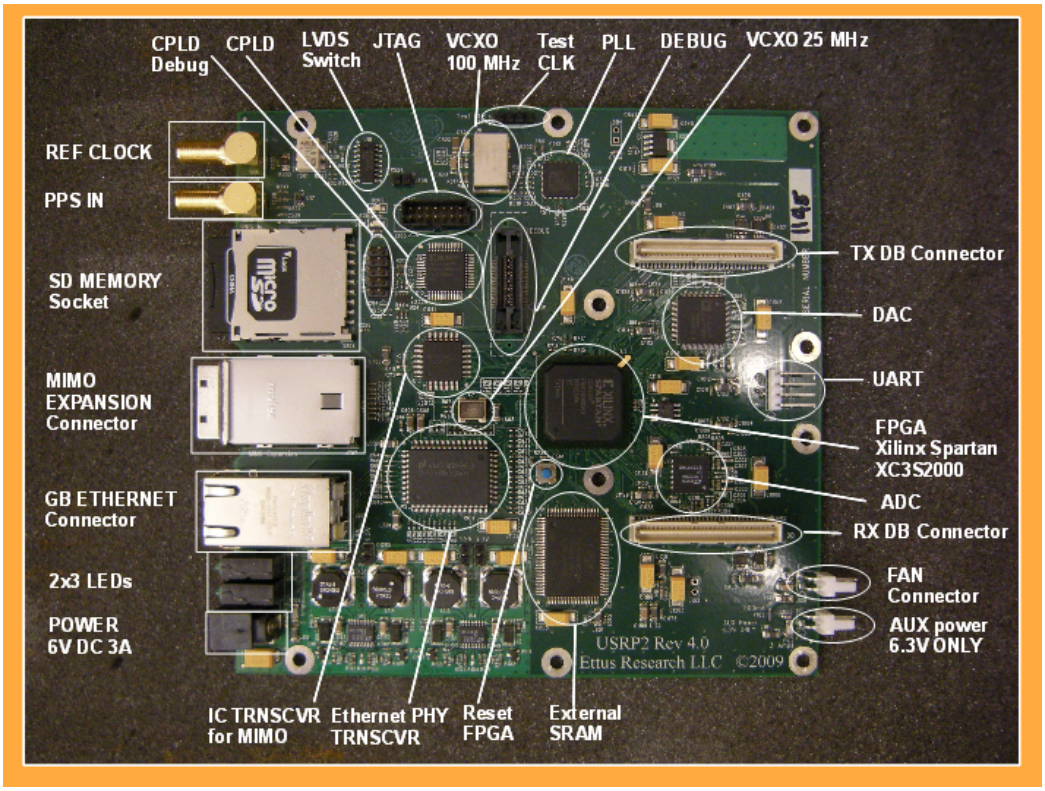


Figure 8.The USRP2's motherboard.

Figure 9.The USRP2's motherboards front interfaces.

Since RX and TX paths are independent, the platform can simultaneously receive and transmit on two antennas in real time. Moreover, all the sampling clocks and local oscillators are fully coherent, thus creating the opportunity for MIMO systems. Multiple USRP2 systems are connected together via MIMO cables and fully coherent multiple systems up to 8 antennas are possible. In order to synchronize the multiple systems, a common frequency reference should be used. For this purpose, the platform offers two external time reference inputs, 1 PPS (pulse per second) and 10 MHz reference clock, which are both standard outputs of a GPS device. Further on, this platform supports one transmit and one receive daughter board, or one transceiver. For easy programming without special hardware, the platform uses a Secure Digital (SD) card for storing FPGAs configurations and microprocessors firmware. As it can moreover be observed from the figure 8, 1MB of external SRAM is also available on the motherboard, as well as many debug connectors. There are also two oscillators, where the 25 MHz one is used to clock the CPLD, while the 100 MHz one is used by the FPGA.

The USRP2s motherboard parameters are summarized in the table 2.

Table 2.USRP2 motherboard parameters. Based on [34]

Parameter	USRP2
FPGA	Xilinx Spartan 3-2000, 32-bits RISC
SRAM	1 MB, high-speed
PC Interface	Gigabit Ethernet
RF bandwidth	25 MHz @ 16 bits
ADC	2 x 14-bit, 100 MS/s
DAC	2 x 16-bit, 400 MS/s
PGA	20 dB both on TX and RX path
Daughter board capacity	(1 TX, 1 RX) or 1 TXR
MIMO possibility	up to 8 antennas, fully coherent
MIMO interface	2 Gbps MIMO cable
Time reference inputs	1 PPS and 10 MHz
Power	6V, 3A

3.2.2 RF front-end daughter cards

Variety of RF-front-ends is available for a broad range of applications, and all the existing RF front-ends are compatible with the whole USRP family. Daughter boards are listed in the table 3. Datasheet for RF-boards can be found in [35]. Moreover, all the boards have a nominal noise figure of 5-7 dB.

Table 3.USRP family daughterboards [35]

Daughter board	Function	Operating Frequency (MHz)	Bandwidth (MHz)	TX Power (dBm)
BasicRX	RX	1 – 250; For use with external RF hardware	-	-
BasicTX	TX	1 – 250; For use with external RF hardware	-	-
LFRX	RX	DC – 30; For use with external RF hardware	30	-
LFTX	TX	DC – 30; For use with external RF hardware	30	-
TVRX	RX	50 – 860	6	-
TVRX2	RX	50 – 860	10	-
DBSRX	RX	800 – 2400	1 – 60 (adjustable)	-
DBSRX2	RX	800 – 2400	1 – 60 (adjustable)	-
WBX	TXR	50 – 2200	30	20
SBX	TXR	400 - 4400	30	14.8 - 20
RFX400	TXR	400 – 500	30	20
RFX900	TXR	750 – 1050	30	23
RFX1200	TXR	1150 – 1450	30	23
RFX1800	TXR	1500 – 2100	30	20
RFX2400	TXR	2300 – 2900	30	17
XCVR2450	TXR	2400 – 2500 & 4900 – 5900	30	20

As stated earlier, the daughter board available for this project is XCVR2450. As the most of these RF-front-end boards, this one is also a direct down-conversion daughter board, meaning that it translates the RF signal directly to a base band signal. Motherboards oscillator on 100MHz, 20 ppm is used as a reference clock for a Phase Locked Loop (PLL) based frequency multiplier IC, which controls a voltage controlled oscillator (VCO) in order to obtain the desired frequency. The IC doing this operation is indicated as the Clock Distribution IC on the figure 10.

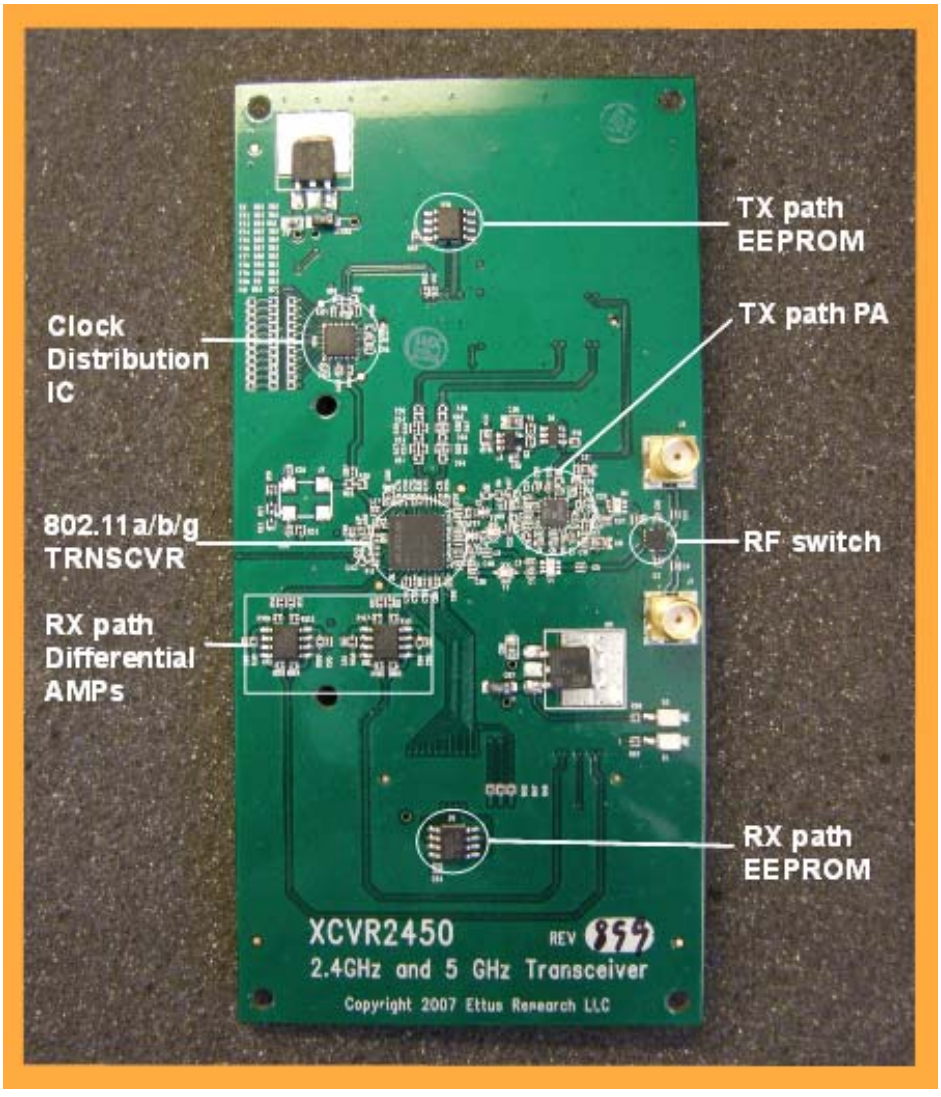


Figure 10. The XCVR2450 daughterboard.

3.3 GNU Radio Software

The primarily used software framework with the USRP family is the GNU Radio [18]. It is a highly flexible licence-free, open-source development toolkit for software defined radios with the focus is to provide a sandbox for easy radio development. It abstracts the signal flow by using signal graphs puzzled from different blocks, allowing easy and relatively quick programming. The GNU radio applications are mainly written in the Python [36] programming language, which is used to glue together the performance-critical signal processing blocks implemented in C++ [37]. To interface these blocks from C++ to Python, a wrapper SWIG (Simplified Wrapper Interface Generator) [38] is used. The structure of the GNU radio software and the main C++ blocks available are given in the figure 11. More information about the blocks can be found in [18]. Moreover, these block-libraries can also be interfaced with even higher languages, as MATLAB/Simulink [39] or LabVIEW[40].

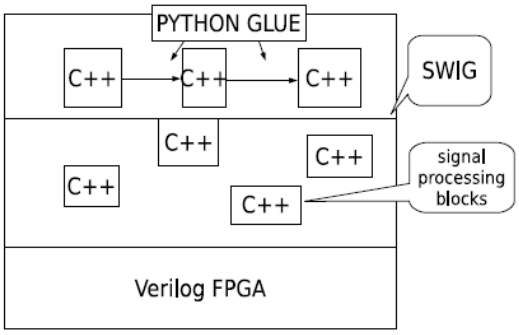


Fig. 1 GNU Radio software structure.

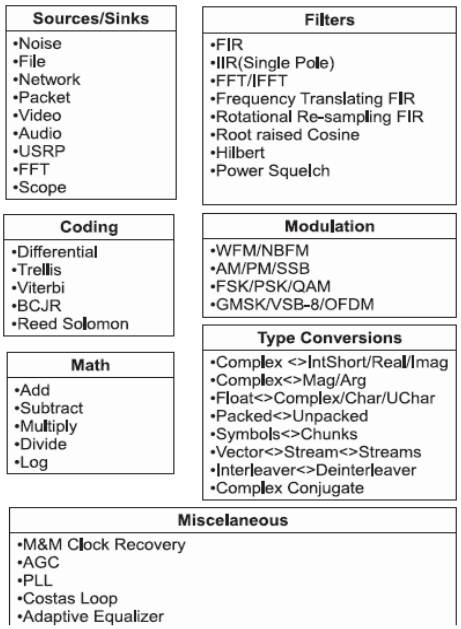


Fig. 2 GNU Radio modules.

Figure 11.GNU Radio software structure and modules available. Based on [41], fig1&fig2. respectively.

This layered structure allows the user to implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment. To ease the

implementation even more, there is also a graphical user interface available, called GNU Radio Companion (GRC), which allows putting together the GNU Radio blocks graphically, constructing signal flow graphs. GRC generates the corresponding code in Python according to these graphs. An example of how these graphs look like is illustrated in figure 12. More information about GRC can be found in [42].

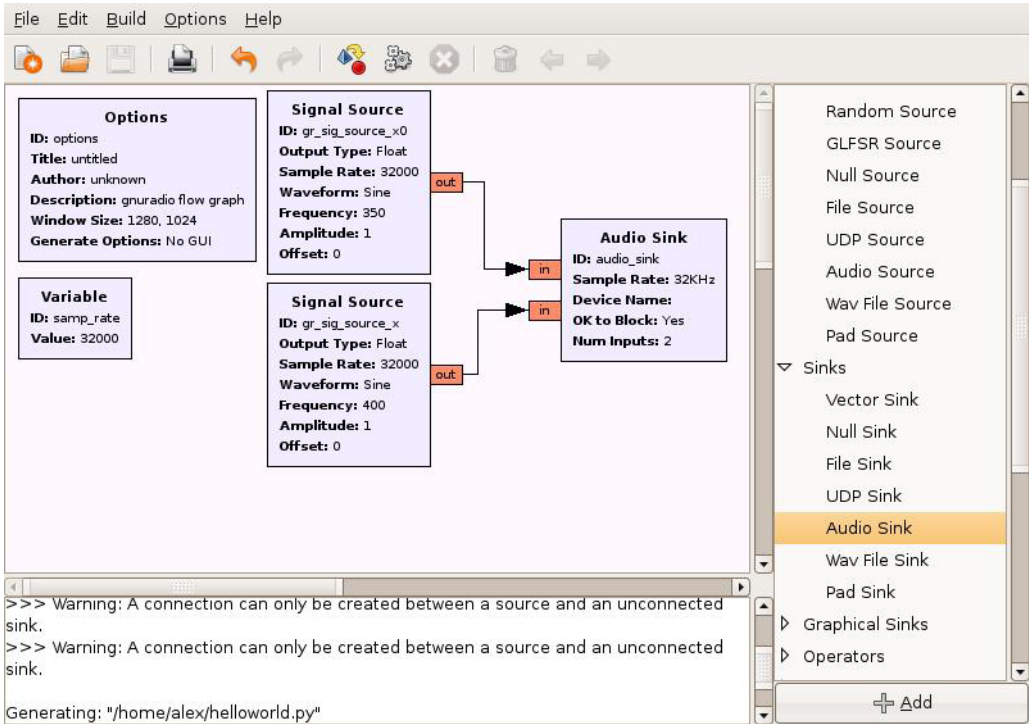


Figure 12.GNU Radio Companion (GRC) screenshot.

To summarize, the GNU Radio has the following features:

- It is open source, under GNU GPL 3 license [43]
- Works with the most of operative systems as Windows, Mac OS, FreeBSD, NetBSD, but Linux is preferable.
- In addition to base building blocks, it contains a set of utility libraries and classes written for different radio applications
- The data flow between blocks is not accessible, contributing to data flow abstraction
- RF-parameters can be modified in run-time

To conclude, the USRP2 and the GNU Radio constitute a fully open source SDR platform. Both of these are flexible enough to be used with some other software or hardware platforms, respectively. Indeed, much more detailed documentation is available for USRP1 platform [44]. However, due to the fact that USRP2 has a lot in common with its predecessor, these documents are usable and can offer much help in resolving problems with the USRP2. For more information about the USRP2 and GNU Radio, see [18] and [31]. Project sites and forums can be found in [19]. For installation and a short introduction manual on the GNU Radio, see appendix A.

4. Radio Resource Management (RRM)

When constructing a radio network, there are some important considerations (and challenges) that have to be taken into account. An umbrella term for these is Radio Resource Management (RRM). Even though RRM includes much more, the most important aspects of it are frequency band(s) of operation, duplex method, multiple access method, and power control. Moreover, RRM involves strategies and algorithms for different controlling parameters with the objective to use the limited radio resource on the most efficient way. This chapter addresses RRM, and is mainly based on [45].

4.1 Frequency band

Frequency spectrum is divided into bands by legislative organs for telecommunications, both on national and international plan. The most of these bands are licensed; however, license-free bands open for more freely use are also provided and can be used for personal wireless communications and research. The choice of the frequency band is important, as it defines requirements of the RF-front end. Increasing the number of frequency bands one wants the radio system to operate in, the complexity and price of the hardware is increasing together with the potential license costs. For research purpose, the unlicensed bands are the most used ones. However, in 2008 FCC opened the digital dividend for cognitive operation, so research can be performed under certain limitations [12].

4.2 Multiple Access method (MA)

There are different methods allowing multiple users to use the same signal - space resource, i.e. to share it. The division can be made in the frequency, time or code domain. In the Frequency Division MA, each of the multiple users gets a narrow band in the spectrum for its disposal. To ensure no interference between the users, guard bands are necessary in this scheme. On the other hand, each of the multiple users gets a time slot for its full disposal in the Time Division MA. Clearly,

synchronization is crucial in this case. Lastly, the Code Division MA is a method where each of the multiple users gets the whole signal-space resource for disposal and gets assigned its own code. The transmitted signal is spread in the bandwidth, reducing the power accordingly. Only the transmitter with the right code can collect and decode the signal, while for the other users in the network the signal looks like low-power noise. These three methods are illustrated in the figure 13.

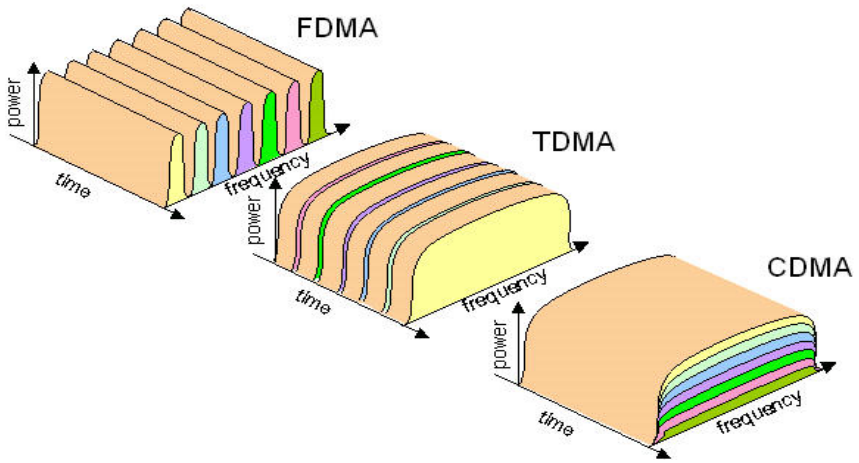


Figure 13. The Multiple Access Methods [46]

These traditional multiple access methods discussed above are used in non-cognitive networks with static base stations and mobile users, where the base stations are delegating the signal-space disposal between the users. On the other hand, in the networks where cognition is used, the signal-space resource is divided in the opportunistic manner, i.e. use if available. The most common way of operation is to sense the channel and decide if a particular narrow frequency band is in use, and if not, to use this opportunity to transmit its own data. Therefore, a cognitive network can be considered as using a version of TDMA with flexible slots.

4.3 Duplex method

The two units can communicate between each other in both directions on two ways: one at the time or simultaneously. If these are communicating one at the time, they are utilizing a half-duplex communication. On the other hand, if the communication is simultaneous in both directions, these are utilizing a full duplex. Furthermore, there are two methods to achieve the full duplex. The Time Division

Duplex (TDD) is full duplex method where the communication in each direction has an assigned time slot. This method achieves full duplex over a half-duplex communication link, and it has a big advantage in the links where asymmetry on up- and down-links is present, since number of the time slots can be adjusted to a needed data rate. Consequently, it is the most used method. On the other hand, the Frequency Division Duplex (FDD) method is a full duplex method where the communication in each direction takes place on different carrier frequencies.

When considering duplex method in cognitive networks, the most common method is the time duplex. However, frequency duplex could also be used, but it requires sensing on two different frequency bands which should be relatively distanced from each other in the spectrum in order to avoid interference. In addition, if the radio possesses only a single antenna, a duplex band filter is needed.

4.4 Power Control

Power control is an important issue in wireless networks, and is used in general to achieve good performance within the system. It is used for different purposes, for example to dynamically adapt the transmit power to the wireless channel for increased effectiveness, i.e. not sending when channel is in a deep fade, and sending with maximum power when channel offers high SNR. However, in the case of cognitive networks, the power control is thought as a method used to reduce interference between networks operating on the same or near frequencies.

4.4.1 Hidden node problem

The easiest way to avoid the need for any power control is to sufficiently increase the distance between the networks. However, the practical cases where this is possible are very seldom. Furthermore, an often problem regarding power control arising in the wireless networks is the hidden node problem, illustrated on the left hand side on figure 14. In the particular situation, node A wants to transmit to the node B. In the same time, the node C, which is not in the range of the node A, but is in the range of the node B, wants also to transmit to the node B. Since not in range of each other, the nodes A and C don't "see" each other when sensing the surroundings,

and start transmission procedure, resulting in collisions and degradation of the network capacity.

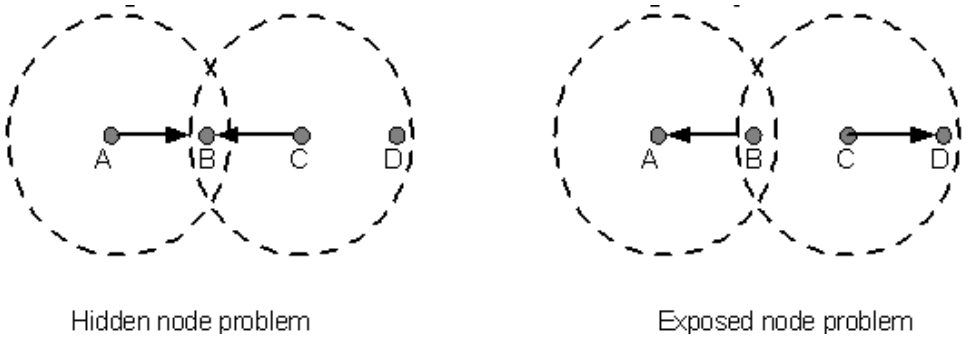


Figure 14. Hidden and exposed node problem [47]

In order to solve this problem, the nodes involved in the potential future transmission have to perform a so called "handshake" prior to a transmission, meaning that the sending node informs the receiving node about its sending intention and waits for an acknowledge. The receiving node sends a positive answer if no transmission is sensed in its range. Moreover, all the nodes in a network are by standard listening to these "handshake" messages, so all the nodes in the range of the involved nodes are informed about the transmission and defer their attempt of accessing the medium for amount of time determined in these messages. In WLAN, the "handshake" messages used are the Request to Send (RTS) and Clear to send (CTS). However, this solution introduces a new problem called the exposed node problem, illustrated on the right hand side on the figure 14. Even though the transmission from node C to D would not introduce collision, node C defers its transmission to node D because it noticed a RTS from node B, which attempts to transmit to node A. This is solved on the following way: If a node hears RTS, but not the corresponding CTS, it deduces that itself is an exposed node and can safely transmit to its neighbours [48].

The hidden node is especially a problem in the cognitive networks employing different priority networks, where the interference is of even greater importance. Even though the primary and the secondary node which are out of each others range would not necessarily wish to transmit to the same node due to different networks, the medium can experience a fading dip which would result in wrongly depicting the channel as idle. The following transmission of the secondary user would introduce harmful interference. This is illustrated on the figure 15.

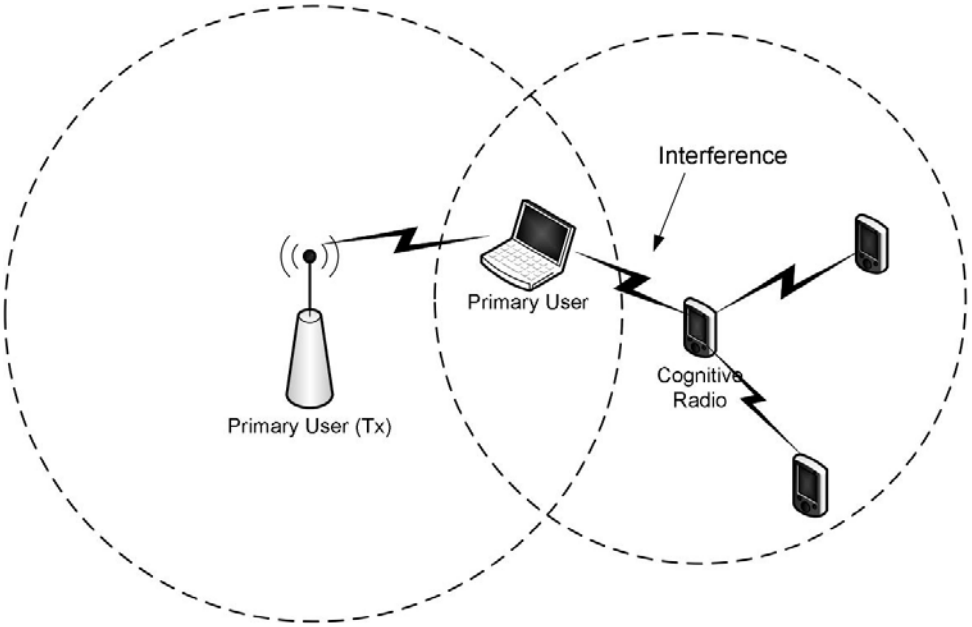


Figure 15.Illustration of the hidden primary user in cognitive radio system [49, fig 2].

4.4.2 Spectrum Sensing Techniques

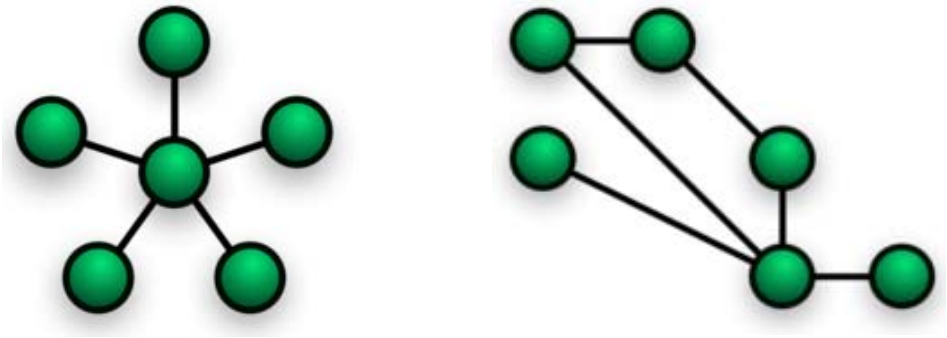
In order to avoid introducing interference to the primary network, it is important to sense the channel on a proper and reliable way, minimizing the possibility of wrong decision about the channel state. This leads to discussion of spectrum sensing techniques, which can be generally classified into three groups [11]:

- **Primary transmitter detection** - based on the detection of the signal transmitted by the primary node, and is the most used method. The usual approaches are matched filter detection, energy detection, and signal feature detection, with cyclostationarity and eigen value as the most exploited features.
- **Primary receiver detection** - based on detecting the primary nodes which are receiving the data in the range of CR user, where the usual approach is to detect the leakage power from the receiver's local oscillator.
- **Interference temperature management** - based on the interference arising between the sensed signal and the signal generated by detector.

4.4.3 Cooperative sensing

All the sensing methods have their advantages and disadvantages, and their extensive overview can be found in [50]. However, none of these methods resolves the problem of the hidden primary user stated above. In order to avoid wrong decisions about the channel, a diversity gain should be introduced in the sensing process, achieved by using the cooperative sensing. This implies the use of multiple sensors, which communicate and cooperate with each other for the best sum result of the channel sensing, and the information obtained can further be used to delegate the communication between secondary nodes.

A sensor network can be structured in different ways, while the two most general ways are the star and mesh networks. In the star structure, all the sensors nodes communicate with one main sensor, a Fusion centre, which they report their results to. The fusion centre communicates with all the secondary network nodes, and can suggest management of the available spectrum resources according to the obtained information. On the other hand, in the mesh structure, all the sensor nodes communicate with at least one other sensor in the network. The information travels via other sensors in range until it reaches its destination, whether the network has a "main" sensor or not. Consequently, this structure is the base of ad-hoc networks. The two network structures are illustrated in figure 16 a) and b).



a) **b)**

Figure 16. Star and partially connected mesh topology [51,52 respectively]

The sensors can communicate with Fusion centre or between themselves by deploying cables, wireless or a mix of these. In the case of wireless communication, the thought is to deploy communication on a very narrow band.

Widely known wireless sensor nodes are the so-called “motes”. These are small autonomous units capable of sensing and communication, mostly used by industry. They have been present on the market for some years, and there are many prototypes and commercial motes/sensor nodes available like BTnodes [53] or Mica2 [54]. A common mote consist of a microcontroller, transceiver, external memory, power source and one or more sensors. Moreover, these nodes also employ special operative systems for low-power wireless devices, e.g. TinyOS [55].

In the case of static primary network, meaning that the nodes are not mobile, sensor network nodes can be placed on strategic places which are optimal to reliably sense the activity of the primary network, i.e. early enough with high certainty. This information can be handed over further to the fusion centre, which would delegate the communication between the secondary units. This just described scheme corresponds to a star topology. However, if the primary network is dynamic and mobile, static sensors would gain variable result depending of on the movement of the primary. A possible alternative in this case is to include a sensor in all the secondary nodes in the network, and employ wireless communication between these sensors. As there is no main node to suggest the spectrum delegation, secondaries have to cooperate in solving this issue. This scheme corresponds more to the mesh type of structure, and is usually called the Mobile Ad hoc NETwork (MANET).

Figure 17.BTnodeRFID reader, and the same reader compactly packed [56]

To summarize, by combining the use of the sensor network and the handshake mechanism results in minimization of the hidden node problem in cognitive networks. Indeed, the concept of the cooperative sensing is used in the mentioned Sendora project [13], where a Wireless Sensor Network (WSN) is deployed in order to obtain reliable channel information. The concept is illustrated on the figure 18.

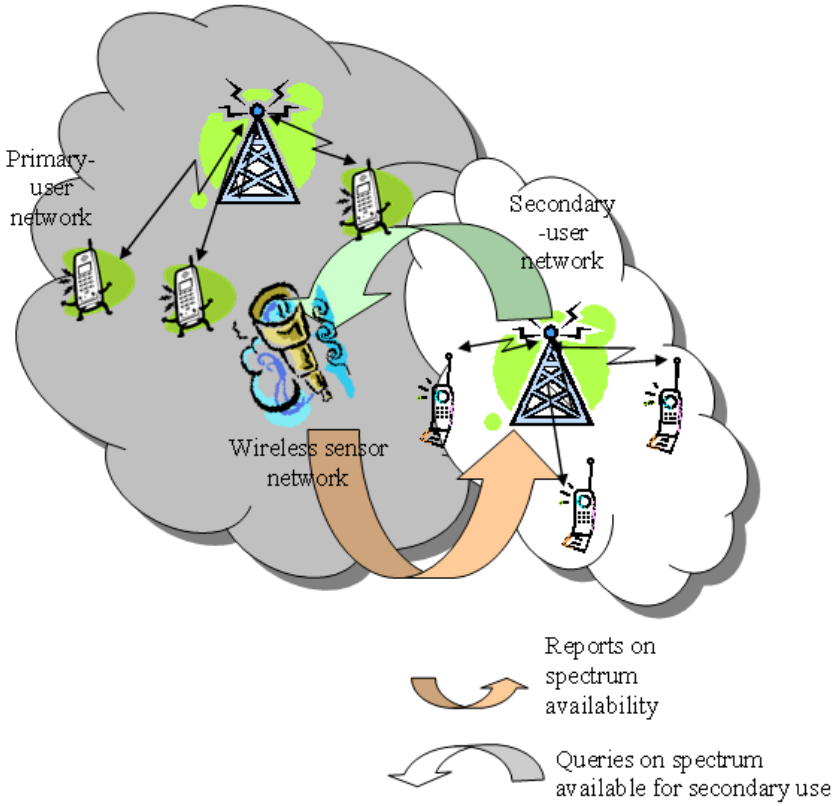


Figure 18.SENDORA WSN aided Cognitive Radio concept [13]

5. Demonstrator description

The goal with the demonstrator construction is to evaluate the real performance of the cognitive radio network, showing that the introduction of the secondary user in the network introduces no or negligible degradation to the primary network. Moreover, the demonstrator will employ USRP2s to perform the over-the-air tests, and an eventual degradation to the primary network will be measured by means of key performance indicators as BER and PL. This chapter describes the thoughts behind the demonstrator in detail.

5.1 Choice of technology

In order to ease the demonstrator construction, it was decided to use an existing technology for the primary user rather than finding out a new standard, while the secondary user's operation is to be adapted in a way which introduces the least interference to the primary user. At the same time, a standardised primary user makes a faster integration of the secondary network more probable, since these primary networks already exist. Fortunately, the secondary user introduces the least interference if it operates similarly as the primary, with, of course, some small adaptations.

The first considered standard was the industrial standard WirelessHART[57], a wireless mesh network communications protocol for process automation applications where each node of the network can serve as a router, providing simple, reliable and secure control of the processes. However, the radio interface of this technology employs IEEE 802.15.4 standard utilizing direct-sequence spread spectrum (DSSS) modulation technique with frequency hopping (FH) on 2.4 GHz ISM band. Since the process of sensing such transmission includes the time granularity which is hard to sense properly and which was intended to be avoided, the choice fell on IEEE 802.11g OFDM PHY standard.

5.1.1 IEEE 802.11g

IEEE 802.11g [58, chapter 19] is a well-known Wireless Local Area Network (WLAN) standard, used to provide wireless connection to the internet. It offers up to 54 Mbit/s raw data rate on 2.4 GHz ISM band. This subchapter is based on [58, 59] and [60], and for more information please referred to these.

Since this standard was thought as a back-compatible improvement of the 802.11b standard, the physical (PHY) layer, besides the Orthogonal Frequency Division Multiplexing (OFDM) also includes Direct Sequence Spread Spectrum (DSSS) and Complementary Code Keying (CCK) modulation schemes, or mixes of these three [58]. However, only the OFDM modulation scheme is discussed here since it is the only one to be used in our demonstrator. Moreover, this standard employs 13 channels (in Europe) in the 2.4 GHz ISM band, each 20 MHz wide and with 5MHz centre frequency separation, resulting in only 3 non-overlapping channels. Each channel employs 64 subcarriers, with 52 active ones. The most important parameters of the PHY layer for the standard are summarized in table 4, while the channel deployment is illustrated on the figure 19.

Table 4. 802.11g PHY parameters. Based on [59, table 24.2]

Parameter	Value
Maximum allowed transmit power	100 mW (Europe)
Information data rate	6, 9, 12, 18, 24, 36, 48, 54 Mbit/s
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
FEC	K=7 convolutional code
Coding rate	1/2, 2/3, 3/4
Total number of subcarriers	64
Number of data subcarriers	48
Number of pilot subcarriers	4
Subcarrier frequency spacing	0.3125 MHz
Occupied Bandwidth	16.6 MHz
OFDM Symbol duration	4.0 μs
Guard interval duration	0.8 μs

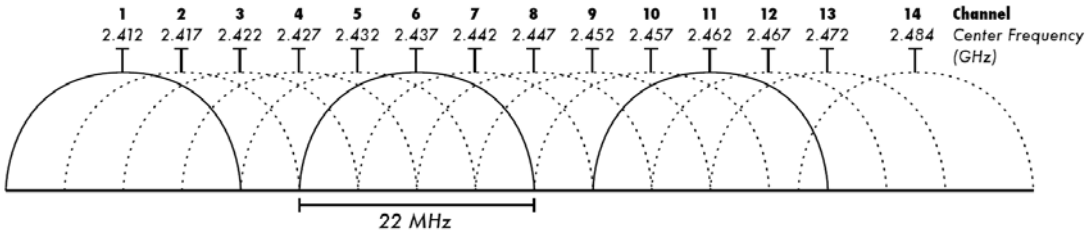


Figure 19. Channel deployment of the 802.11g standard in ISM band in Europe. Due to the side lobes, channels are in practice 22 MHz wide, in contrast to the mentioned 20 MHz of which only 16.6 MHz is used, because of the 12 null-subcarriers of in total 64 subcarriers employed. [61]

Before continuing to the Medium Access Control (MAC) layer, the relationship between the layers and notation used are briefed. The standard defines the PHY and MAC layers, and these receive data payload from higher layers. Before sending this data on the air, the PHY and the MAC layers attach different headers and trailers to it. The relationship between the layers and the data units sent between these is shown in the figure 20. Moreover, note that the MPDU and the PSDU are identical in practice; the MPDU changes its name to the PSDU when handed over to the PHY layer.

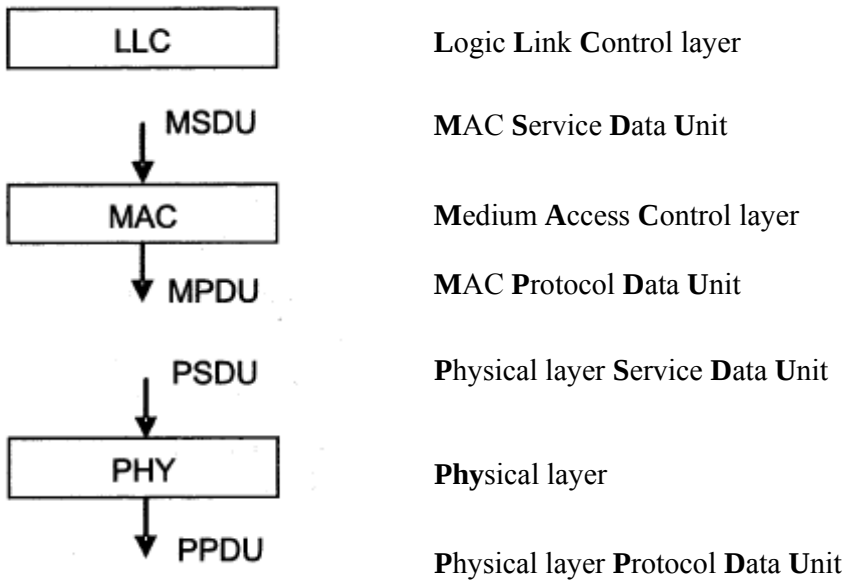


Figure 20. Relation among data units between layers. Based on [59, fig.24.1]

The size of the MSDU can vary from 0 to 2304 octets. Furthermore, the MSDU is taken over by the MAC layer in order to make a MPDU frame. The MPDU frame format is depicted in the figure 21. The Frame Body field is of variable size, with the maximum size determined by the maximum MSDU size plus any overhead from security encapsulation.

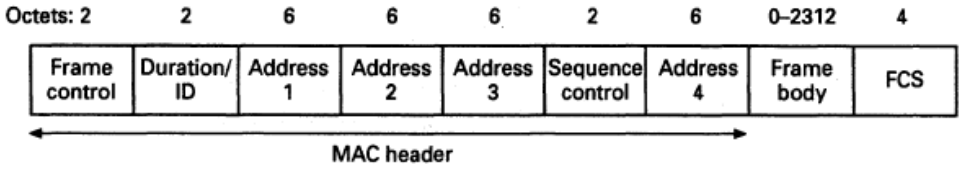


Figure 21. MPDU frame format [59, fig.24.11]

Indeed, the MAC layer is identical for all the standards of the 802.11 group, and hence the MPDU frame formats are identical. However, in contrast to the MAC layer, the PHY layer differs for the different standards, and hence the PPDU frame formats are different. The PPDU frame format for 802.11a/g is depicted in the figure 22. The PSDU is of variable size, and can vary from 1 to 4095 octets.

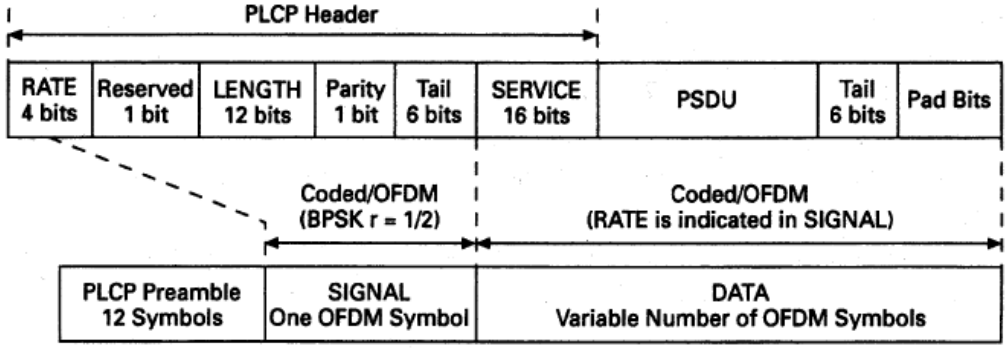


Figure 22. PPDU frame format [59, fig.24.5]

The difference in the maximum size between the MPDU and the PSDU, even though it was stated earlier that these data units are identical, is due to the convolutional coding performed on the PSDU, and therefore more correct would be to write coded PSDU (C-PSDU).

Before continuing with the description of the MAC layer, I would like to mention all the tools of MAC management used in WLAN: authentication, association, address filtering, privacy, power management, and synchronization. Indeed, the MAC layer performs the RRM function in this standard. However, in this review only the most important functions for our purpose are described. For more information about the reviewed and omitted MAC functions, please refer to [60].

The MAC layer association tool, which is responsible for the access, employs a temporal "superframe" structure with the Contention Period (CP) and the Contention Free Period (CFP). As it is illustrated on the figure 23, the superframes are separated with the "beacon frames". These "beacons" are sent by the Access Point (AP) and carry the clock information, which is used for the synchronization of the AP with wireless stations (STAs) in the network. In order to control the medium access, MAC uses also different inter frame gaps called Inter Frame Spaces (IFS) to regulate priority of the STAs. Their actual values in time units depend on the PHY parameters of the standard. The Inter Frame Spaces are listed in the table 5 in order from the shortest to the longest, including the real values for 802.11g OFDM PHY. These IFSs are illustrated on the figure 24.

Table 5. Inter frame spaces in 802.11g with OFDM PHY

Parameter	Abbreviation	Length [1] for 802.11g
Short Inter Frame Space	SIFS	10 μs
Priority Inter Frame Space	PIFS	30 (19*) us
Distributed Inter Frame Space	DIFS	50 (28 *) μs
Extended Inter Frame Space	EIFS	-

*Optional, used only when all STAs in a network employ 802.11g OFDM PHY

In the contention-free period (CFP), the access is controlled by using the Point Coordination Function (PCF), a method based on polling where the AP coordinates the traffic, and all the "hooked up" STAs on the AP are obeying inherently. Since the

[1] Refering to [58, page 270], the formulas for IFSs:

SIFS = aSIFSTime
PIFS = aSIFSTime + aSlotTime
DIFS = aSIFSTime + 2 · aSlotTime

Refering to [58, Table 19.7] for values of 802.11g OFDM PHY

aSIFSTime = 10us
aSlotTime: Long = 20us; Short = 9us, short is optional, used only when all STAs in network employ the same standard version, 802.11g OFDM PHY

usage of the medium is strictly controlled, PIFS is used as the sufficient time between transmissions. This method can be useful when much load unevenly divided between STAs is introduced to the network, but it is rarely used in practise.

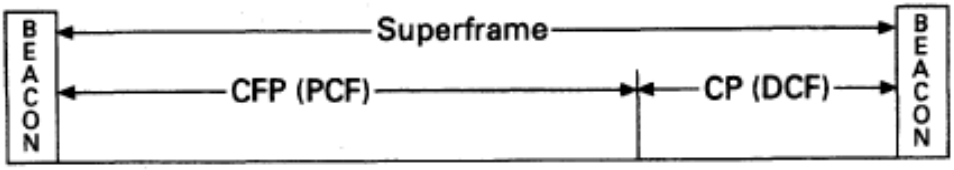


Figure 23. Superframe structure of 802.11 [59, fig.24.10]

On the other hand, the contention period (CP) based method employing the Distributed Coordination Function (DCF) to regulate the medium access is the more used one. When in this mode, a STA "listens before talking", and uses in addition a random backoff mechanism to avoid collision when the medium is first sensed idle. This access mechanism is called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), corresponding to the explained functions respectively. Besides this partly "physical" sensing mechanism, MAC employs also a pure virtual carrier sensing mechanism, called Network Allocation Vector (NAV), which's usage is explained later in this chapter. In his book "Wireless Communications", Molisch explains the DCF function [59, page 555]:

"In DCF mode, each STA checks whether the channel is idle before attempting to transmit. If the channel has been sensed idle for a DIFS period, transmission can begin immediately. If the channel is determined to be busy, the STA will defer until the end of the current transmission. After the end of the current transmission, the STA will select a random number called a "backoff timer", in the range between 0 and a *Contention Window* (CW). This is the time the WM has to be free before the STA might try to transmit again. The size of the CW increases (up to a limit) every time a transmission has to be deferred. If the transmission is not successful, the STA thinks that a collision has occurred. Also in this case, the CW is doubled, and a new backoff procedure starts again. The process will continue until transmission is successful (or discarded). "

Indeed, the backoff procedure algorithm gives a random number which is uniformly distributed in the range between 0 and the current CW value, and this number is the

number of the defined time units the STA has to wait in order to transmit after the channel is sensed idle. Consequently, this semi-randomness decreases heavily the possibility of collisions. When the transmission is successful, the CW takes again its minimum value (CWmin). The parameters of CWmin and CWmax are fixed for particular PHY, but these may differ from one PHY to another. For illustration of the basic access method and backoff procedure, see the figures 24 and 25, respectively.

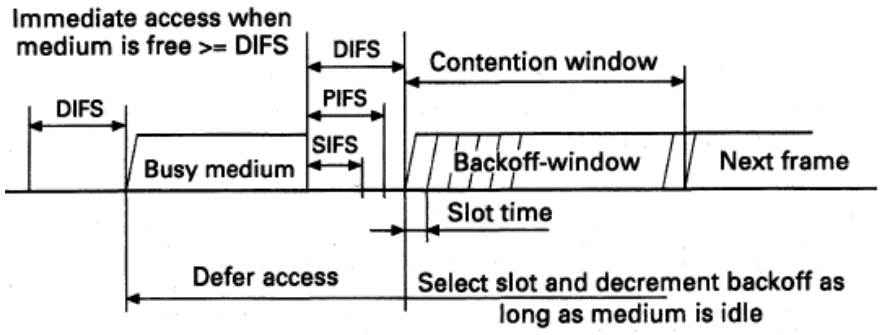


Figure 24. Basic medium access method [59, fig.24.12]

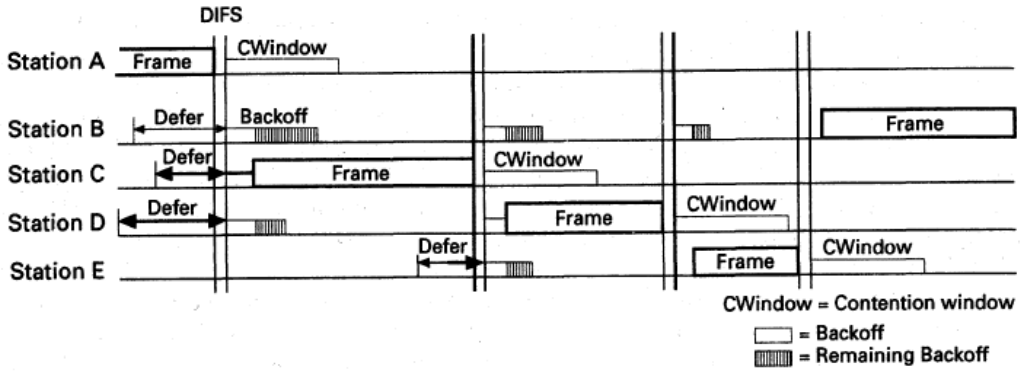


Figure 25. Timing backoff procedure [59, fig.24.13]

However, it can happen that the sensing STA is not in the range of the currently transmitting STA, and the STA is not sensing any transmission on the medium even if the medium is used. To avoid collisions in these cases, the earlier mentioned NAV gets into the picture. NAV is a value that indicates to a station the amount of time that remains before the medium will become available, and is kept current by transmitting the duration values in absolutely all transmitted frames. Indeed, it is NAV which is used in the double handshake mechanism RTS/CTS to

solve both the hidden and the exposed node problem. Its mechanism is depicted in the figure 26. As it can be confirmed from the figure, all the STAs in the network are using NAV, and since it is transmitted in absolutely all frames, it is also updated with each RTS and CTS sensed.

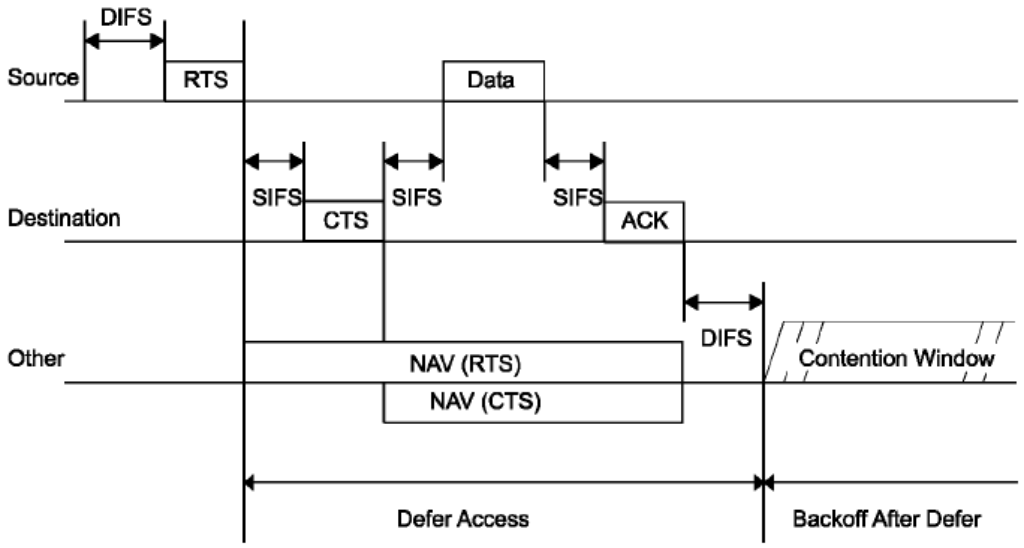


Figure 26. RTS/CTS mechanism [58, fig.9-7]

The RTS frame is illustrated in the figure 27, and it is sent to the desired destination after the channel has been sensed idle for a DIFS period. Note the RA and TA fields, which are the addresses of the receiver and the transmitter STAs, respectively. In addition, the duration field is the broadcasted new NAV value.

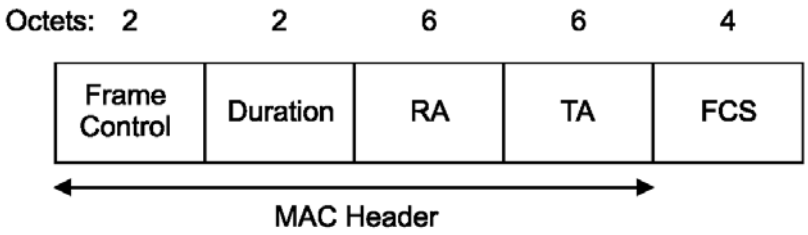


Figure 27. RTS-frame [58, fig.7-6]

After the destination STA receives the RTS frame, and if no transmission has been sensed for a SIFS period of time, the CTS frame is sent to the source STA. The RA field of the CTS frame, which is depicted on the figure 28, takes value of the TA field in the prior RTS.

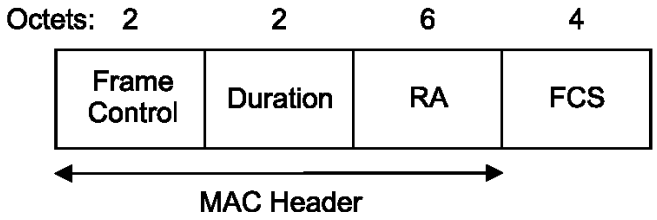


Figure 28. CTS-frame [58, fig.7-8]

After the transmission of the data, and if the transmission succeeded, acknowledge (ACK) frame is sent to inform the sender about it. The ACK frame has identical structure as CTS frame, which is depicted in the figure 28.

Finally, in order to conclude the overview of the 802.11g standard deploying the OFDM PHY, a table of the time parameters from the standard is provided to give a better understanding over the strict time requirements demanded by the standard. For more details about the 802.11g standard, see [58].

Table 6. Time parameters for the 802.11g OFDM [58, table 19-7, page 719.]

Characteristic	Value
aSlotTime	Long = 20 µs, short = 9 µs
aSIFSTime	10 µs
aCCATime	<15 µs for long slot time or <4 µs for Short Slot Time
aPHY-RX-START-Delay	24 µs for ERP-OFDM, 192 µs for ERP-DSSS/CCK with long preamble, and 96 µs for ERP-DSSS/CCK with short preamble
aRxTxTurnaroundTime	<5 µs
aTxRxTurnaroundTime	<10 µs
aTxPLCPDelay	Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met.
aRxPLCPDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	<<1 µs
aTxRampOnTime	Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met.
aTxRampOffTime	Implementation dependent as long as the requirements of aSIFSTime are met.
ATxRFDelay	Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met.
ARxRFDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	<<1 µs
aMACProcessingDelay	<2 µs
aPreambleLength	20 µs
aPLCPHeaderLength	4 µs
aMPDUMaxLength	4095
aCWmin(0)	31

5.2 Consequences of the limitations and choices

In accordance to the limitations and the choice of the technology, and the consequences these carry with them, a demonstrator description is made. These are listed and discussed below.

- Due to daughter cards available, for this project the ISM frequency band is chosen. Primarily the frequency of 2.4 GHz will be used, but 5 GHz is also available in case we want to spare WLAN traffic already present at the university.

- The 802.11g standard using the OFDM modulation scheme, chosen for the primary network, is going to be used in the distributed coordination mode (DCF). It employs time division both for multiple access and duplex method. However, in our arrangement no AP will be present.

- Limited people resources on the project and limited number of USRP2s led to decision to omit the sensor network for sensing the channel. If the demonstrator test is performed with all the involved units present in the same room, an assumption that all the sensors would sense the same channel is quite correct. In order to acquire information about the activity on the channel in the case with omitted sensor network, each secondary node will sense the medium using its own CSMA/CA protocol implemented in the MAC layer. This protocol detects the Received Signal Strength Indication (RSSI), which corresponds to the energy detection of the primary transmitter. Furthermore, the choice of omitting the sensor network also eases the demonstrator complexity.

The below explained limitations, choices and the following consequences are summarized in the table 7.

Table 7. The limitations, choices and following consequences.

Parameter	Limitation	Consequence
SDR	USRP2	-
# of units	9	No sensor network
Daughter board	XCVR2450	ISM band, both 2.4 and 5 GHz
Software	GNU Radio	-
Research environment	Indoor, NTNU	-
Primary standard	802.11g	TDMA & TDD
Sensing technique	Primary transmitter detection, RSSI	Energy detection

5.3 Demonstrator setup

The available 9 USRP2s are divided in the 4 primary users and the 5 secondary users, and the each USRP2 unit requires a dedicated GPC. Moreover, as explained earlier, the 802.11g standard employs 3 non-overlapping channels (1, 6, 11), and exclusively these 3 channels are used in the demonstrator. Moreover, the 4 primaries are divided in two pairs where each pair uses one channel to communicate with each other, leaving always one channel unused. The rest of 5 secondary units should exploit the one left unused channel. Alternatively, the available bandwidth of the unused channel could be divided in two subchannels, each used by one pair of the secondaries, always leaving one secondary unit to wait for an available subchannel. In this thesis, it is chosen to implement the first case, using the whole bandwidth.

The demonstration setup is summarized in the table 8, while the schematic illustration of the demonstrator's function is shown in the figure 29. The GPC required for the each node is omitted on this illustration.

Table 8. The demonstrator setup.

Cognitive radio network item	Item coverage
Primary Network	WLAN 802.11g with 4 fixed operating nodes
Primary Node	GPC + USRP2
Secondary Network	5 fixed operating nodes
Secondary Node	GPC + USRP2
Secondary network topology	Mesh
Secondary Network Standard	Adjusted* WLAN 802.11g

* Adjustments of the standard are given in the section 5.4

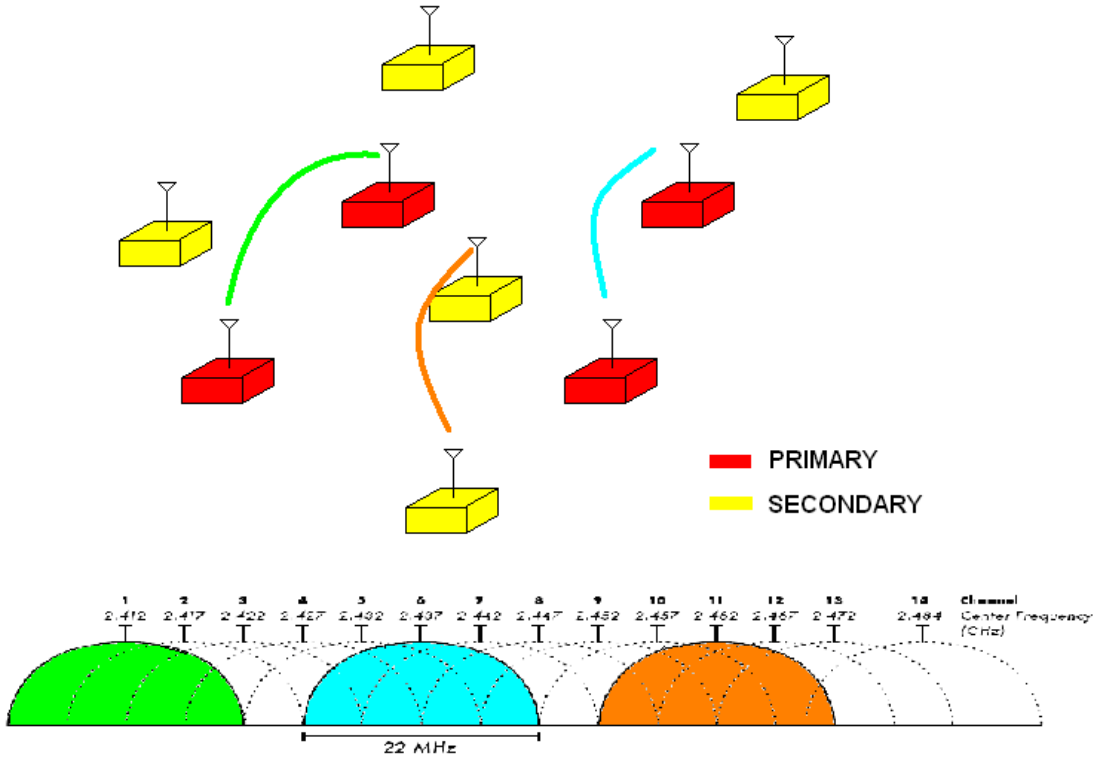


Figure 29. The illustration of the demonstrator function regarding channels and USRP2 units. Based on [61].

5.4 Adjustments of the standard

In order to ensure the priority of the primary network, i.e. that the primaries always get the possibility to access the channel first in any case, and the least possible capacity degradation of the primary network, the chosen WLAN 802.11g standard has to be slightly adjusted for the secondary units. The three proposed adjustments are listed and explained below. The other parameters of the standard are staying intact.

1. Longer mandatory channel-idle time and CW for the secondary units

All of the units, both the primary and the secondary ones, are operating in the DCF mode and sensing the desired channel prior to the transmission by using CSMA/CA, as described earlier in this chapter. The Distributed Inter Frame Space (DIFS), which is the shortest time the channel has to be sensed as idle prior to the transmission in DCF mode, will be extended for the secondary units compared to DIFS for the primaries in order to allow the primary units to always access the channel first if the use of the channel is desired.

Moreover, in order to always allow the primaries to access the channel first in a case of collision, the minimum contention window (CW) value for the secondaries has at least to be equal to the maximum CW of the primaries. In this way we increase the probability that the random backoff of the primary will be shorter than the secondary one's.

To summarize the two statements above, these are two conditions which have to be satisfied: $CWmin\_sec \geq CWmax\_pr$ and $DIFS\_sec \geq DIFS\_pr$. The optimal prolongation will be decided by practical try-and-fail tests, since longer waiting represents a potential resource wasting if the spectrum lies idle, so the balance between has to be found. These adjustments are both the MAC layer adjustments.

2. Shorter frames for the secondary units

If the secondary unit wants to exploit a channel not currently in use, after the channel is sensed as idle for sufficient amount of time, the secondary unit starts the transmission. While the sent frame is on-the-air, it is possible that the primary user desires to use precise this channel, but the activity is sensed on the channel and the primary user waits for the channel to be idle again. In order to introduce the shortest possible delay to the primary network, which directly introduces the capacity

degradation to the primary system, the secondary users should have shorter frames than primaries, i.e. MSDU_sec ≤ MSDU_pr. One possibility is to set the secondary unit MSDUs to be half of the size of the MSDUs used by primaries. However, the primary MSDU size varies between 0 and 2304 octets. In addition, it is difficult to calculate the length of frames in time units, since it depends both on modulation, coding rate and the data rate of the network. Therefore the optimal size has to be determined by practical try-and-fail tests. However, the optimation will be started with limitation of the maximum size of the secondary packet to half of the size of the primary one. This adjustment is a higher layer adjustment, since the MSDU is originating from LLC layer.

3. The primary and the secondary frames must be distinguishable

In addition to the privilege of always getting the possibility to access the channel first, the priority of the primaries also includes the privilege of keeping the channel for itself when one accessed at the first place. This implies that the secondary units using one particular channel, when activity of the primary network on the same channel is noticed, should immediately stop in their intention to transmit and leave the channel. To transmit again, the secondary units have to sense the wireless medium and exploit the channel which is not in use by the primaries. In order to be capable of noticing the activity of the primary network, the secondary units have to be able to distinguish between the primary units' and its own frames. To solve this, it is chosen to include one bit in the PLCP header of the PPDU frame which distinguishes these two networks. It is proposed that this bit takes place of the reserved bit in PLCP header, which is reserved for future use, according to the standard [58, page 603]. The reserved bit properties are defined in the table 9. This adjustment is the PHY layer adjustment.

Table 9. Reserved bit options.

Network type	Header bit
Primary	0*
Secondary	1

*0 is the default value in the standard

In addition to these three mentioned adjustments, the secondary units have to be “tuned” on the same channel in order to communicate, i.e. to hear each others RTS and CTS. Since there is no information exchange about the spectrum holes between the secondaries, a way to get all the secondaries on the same channel has to be found. Approach proposed here is the following: At the very start of the operation of the secondary network, when no channel has yet been chosen, all the secondary units are sensing the wireless medium in order to find the one unused channel. Since sensing can be performed only on one channel at a time, all the three channels are sensed for a small amount of time in a cyclic order, and this cycle is performed for a number of times. After a determined, sufficient number of the cycles have been performed, the secondary units calculate the possibility of collision in each channel, by calculating the relation between the number of times the channel has been sensed and the number of times activity on the channel has been noticed. Consequently, the channel is chosen according to the least possibility of collision. In this particular case, one channel will have zero-possibility of collision, and this one will be chosen by the secondary units. However, this approach is successful only under assumption that the primaries are relatively active. This “startup” procedure is also performed each time the secondaries have to find a new available channel, i.e. if a primary activity has been sensed in the used channel.

Alternatively, a secondary node, which did not receive the following CTS frame when RTS is sent, could assume that the receiver node is sensing some other channel. The solution for this is to send RTS on more channels, and listen for CTS on all channels in a cyclic order. However, this approach is limited by our goal of introducing negligible interference to the primary.

5.5 Example of operation

This chapter presents a possible example of demonstrator operation. For example, let the primaries to be labelled as P1, P2, P3 & P4, and the secondaries as S1, S2, S3, S4 and S5. Moreover, only channels 1, 6 and 11 are the possible channel alternatives. Assuming the restart of the networks, no channels are yet in use. In order to get the proper operation of the demonstrator, the following assumptions are made:

- The primaries are relatively "chatty", meaning that they are not silent for more than a small, apriori known amount of time
- The spectrum occupancy is changing slowly "enough", meaning that the primaries change between the channels in a rate which allows the secondaries to sense and use the available channel

Steps in the event sequence of the demonstrator operation	**Comments**
1) The first pair of the primaries (P1, P2) is set up to use the channel 1	
2) The second pair of the primaries (P3, P4) is set up to use the channel 6	
3) All the secondary users (S1, S1, S3, S4, S5) are sensing the channels 1, 6 and 11, in a cyclic order	
4) After 10 rounds of sensing the three channels, all the secondary calculate the collision probability	10 is a random chosen value
5) All the secondaries "tune" themselves to the unused channel 11	
6) Suppose S1 desires to transmit to S4, and acts in the following manner:	
a. S1 senses for activity on the channel 11	
i. S1 does not notice any activity on this channel for a DIFS long time	
ii. S1 sends RTS to S4	RTS contains the length of the frame

	iii. S4 does not notice any activity on the channel for a SIFS long time, and sends CTS to S1	Ref. fig. 26
	iv. S1 waits a SIFS and transmits a frame to S4	Ref. fig. 26
	v. Suppose successful transmission; S4 waits a SIFS and transmits an ACK	Ref. fig. 26
	vi. S1 desires to continue to transmit to S4, and waits a random backoff	Ref. fig. 25
	vii. Suppose S2 desires to transmit to S5, and senses the channel for activity	Just after S4 started backoff
	viii. Suppose S2s' DIFS is shorter than S1s' random backoff, and S2 does not notice any activity on the channel for a DIFS	S2 DIFS < S1 DIFS + backoff
	ix. S2 sends RTS to S5	All the STAs hear RTS
	x. S5 does not notice any activity on the channel for a SIFS long time, and sends CTS to S2	CTS also contains the length of the frame
	xi. S2 waits a SIFS and transmits a frame to S5	
	xii. S1 analyzes the packet header and finds the bit distinguishing the packets	
	xiii. It determines that this packet was from the secondary network, and continues to wait to transmit	Reserved bit = 1
	xiv. Suppose successful transmission; S5 waits a SIFS and transmits an ACK	
	xv. S1 waits now a DIFS + random backoff in order to send to S4	
	xvi. Suppose no activity sensed on the channel for a DIFS + backoff, and S1 sends RTS to S4	

	xvii. S4 does not notice any activity on the channel for a SIFS long time, and sends CTS to S1	
	xviii. S1 waits a SIFS and transmits a frame to S4	
	xix. P3 desires to transmit to P4, and senses for activity on the channel 11	P3 changes channel from 6 to 11
	xx. P3 notices the activity on the channel, and waits for the transmission to be done	Latency introduced for the primary
	xxi. Suppose successful transmission; S5 waits a SIFS and transmits an ACK	
	xxii. P3 waits a DIFS + random backoff in order to access the channel	
	xxiii. S1 desires to continue to transmit to S4, and waits a random backoff	
	xxiv. Since P3s' DIFS is shorter than S1s' backoff, P3 accesses the channel first and sends RTS to P4	P3 DIFS + backoff < S1 backoff
	xxv. P4 does not notice any activity on the channel for a SIFS long time, and sends CTS to P3	
	xxvi. P3 waits a SIFS and transmits a frame to P4	
	xxvii. S1 analyzes the packet header and finds the bit distinguishing the packets	
	xxviii. It determines that this packet was from the primary network, and immediately leaves the channel	Reserved bit = 0
	xxix. All the secondaries present in the channel analyze the header bit in the sent frame	
	xxx. They determine this packet was from the primary & immediately leave the channel	Reserved bit = 0

Further on, all the secondaries start again by sensing all the three channels in a cyclic order to find an unused channel.	

This example shows a possible scheme for practical operation of the demonstrator. When all the nodes, both primary and secondary, are in contention about the same channel, the operation is completely the same, but the random backoff mechanism gets included more actively.

As stated earlier, both RTS and CTS contain the duration field with information about the length of the following packet, so all the nodes in the neighborhood operating at the same frequency get informed of approximately how long they have to defer the new try to transmit.

Furthermore, as described in the example, if a primary unit decided to change the channel, it will get the priority in accessing the new channel, while the secondary unit will leave the channel as soon as the operation of the primary has been noticed. In addition, since there is no difference in RTS/CTS or ACK frames between the primary and the secondary, the units comprising the secondary network analyze all the frames sent in the network, in order to be able to “evacuate” from the channel as soon as the primary activity is present. As the reader may have concluded already, the security of the primary network towards the secondary is disregarded in this case.

6. Implementation & evaluation

In this chapter the related work and implementations are given, as well as the description of the implementation. Moreover, the results and discussion is given. Finally, the conclusion and future work are concluding the chapter.

6.1 Related work

As stated earlier, the goal with the demonstrator is to show that the introduction of the secondary user in the network will introduce no or negligible degradation to the primary network. To be able to show this, the frame loss on the PHY layer in the primary network, due to the interference with secondary network, has to be measured. The easiest way of measuring the frame loss is to measure the data loss reported to the higher layers. However, it is unclear how high up one need to go; is it possible to measure it in the Data layer, or is it required to go higher up to the Network layer? If using higher layer communication, IP packets can be quite big, and consequently have to be fragmented in more MSDU frames. If one sent frame gets lost, it can result in discarding the whole IP packet and reporting the loss of the whole IP packet. On this way, the information on the actual frame loss, exactly the quantity we want to measure gets lost. In addition, as higher one goes in the layer system, the bigger latency is introduced. However, at least the PHY and MAC have to be implemented.

Fortunately, the PHY layer of 802.11g has already been implemented in the GNU Radio with the USRP2. Indeed, The Telecommunications Research Center Vienna (FTW) implemented an OFDM encoder, which is fully standard-compliant with the IEEE 802.11a/g/p in means of the frame structures. Moreover, in the same paper [62], an industrial chipset and the USRP2 are compared in generated power spectrum, frame-error-ratio when decoded by the industrial chipset, and the received signals generated by these two. The results indicated that USRP2 is quite on par with the industry-chipset, with some small imperfections. However, only transmitter is fully implemented at the moment, and the receiver is under development.

Continuing there where the FTW stopped, Ubiquitous Wireless Communications Research Laboratory (Uwicore) developed a fully programmable and modifiable SDR implementation of the IEEE 802.11 MAC layer [63], which can

be used to develop advanced cross-layer communications. The MAC layer has been fully implemented in Python language, and is constructed to communicate with other OSI layers through sockets, so it can be used independently of the PHY layer.

Another approach is used in the Hydra multi-hop wireless testbed [64], which is used to investigate MIMO ad-hoc networking, developed by the University of Texas at Austin and Drexel University. In this testbed, the PHY compliant to IEEE 802.11n is designed using GNU Radio. However, the MAC layer is implemented using "Click modular router" [65], which is a software development tool created by MIT Parallel and Distributed Operating Systems group. Click enables a flexible and modular MAC and routing layer design, which is easily interfaced to the IP network stack via Linux TUN [66], allowing MAC to interoperate with higher layers, i.e. applications, enabling end-to-end tests. A similar approach is used in [67], where Linux TUN/TAP [66] acts as an emulator and provides interface whole the way from PHY to higher layers.

However, neither of these two approaches solves the latency problem when employing the software defined radios. The flexibility and easiness introduced by using the high-level languages and performing the vast of the signal processing in the GPC result in a software processing delay, and there is a well-known trade-off between these two. Even though the USRP2's sampling rate is fast enough to comply the demanding 802.11a/g/p standard time requirements, the software processing of the MAC and PHY layers results in important processing delays, hindering the standard compliance. Hence, in the developed code for 802.11a/g/p PHY [62] and 802.11 MAC [63], all the standard time parameters are multiplied with a constant which is sufficiently high in order to get a functional code which bears up with the latency. Furthermore, the Uwicore laboratory [63] performed tests to measure the processing delays incurred by the transmission of a data frame. The results are provided in the table 10. For details about the measurement, see [63].

Table 10. Process Delay Characterization [63, table 1]

USRP2 signal forming	16.665 µs/bit
PHY processing delay	0.942 µs/bit
Socket MAC – PHY delay	1.827 ms
MAC-Data processing delay	103.232 ms
MAC-CS processing delay	20.692 ms

Another latency test has also been performed by using a Wireshark packet sniffer [68] which recognizes the frame format. The RTS, CTS, Data and ACK frames are sent both by an industrial chipset and USRP2, as illustrated on the figure 30.

No. .	Time	Source	Destination	Protocol	Type
1	0.000000	LiteonTe_3a:1f:51 (TA)	Ubiquiti_84:3b:87 (RA)	IEEE 802.11	Request-to-send
2	0.000348		LiteonTe_3a:1f:51 (RA)	IEEE 802.11	Clear-to-send
3	0.000354	192.168.1.46	192.168.1.49	ICMP	Data
4	0.000357		LiteonTe_3a:1f:51 (RA)	IEEE 802.11	Acknowledgement

a) Commercial wireless card

No. .	Time	Source	Destination	Protocol	Type
1	0.000000	EttusResea_03:0c (TA)	EttusResea_03:10 (RA)	IEEE 802.11	Request-to-send
2	0.086678		EttusResea_03:0c (RA)	IEEE 802.11	Clear-to-send
3	0.182696	EttusResea_03:0c	EttusResea_03:10	LLC	Data
4	0.270464		EttusResea_03:0c (RA)	IEEE 802.11	Acknowledgement

b) Implemented SDR 802.11 MAC

Figure 30. Comparison of the latency in a message exchange sequence for a commercial wireless card and implemented SDR 802.11 MAC. For detail about the measurement, see [63].

According both to the table 10 and figure 30, the latency introduced by SDRs is obvious, and the values are in general tree orders of magnitude greater, i.e. from microseconds to milliseconds, than the one required by the 802.11 standard and achieved by the industrial chipsets. This fact is also confirmed in [67], where the round - trip time (RTT) when pinging the USRP1 is about 30ms, compared to the Ethernet's and WiFi network card's RTT of less than 1ms.

The researchers on the Carnegie Mellon University addressed the SDR's high latency issue in their extensive work to enable the flexible MAC protocol implementations on the SDRs [69]. In order to decrease the latency, they proposed a split – function architecture, i.e. to split up the core MAC functionality between the host processing unit and the processing unit on the SDR platform, implementing the control of the MAC on the host to preserve the flexibility, and the most time-critical MAC functions on the platform to achieve performance. The functions which were pointed out as "critical" are: precise scheduling in time, carrier sense, backoff, dependent packets, packet recognition, fine-grained radio control, and access to the PHY layer information. In order to succeed with this division of MAC functions, they introduced two architectural features, per-block meta data and a control channel. The

first one introduces a header to the each packet including a timestamp, a channel flag (data/control), a payload length and different single bit flags for some specific functions. On the other hand, the control blocks also carry the same meta-data header but with the channel flag set to control, and these bear the less frequent control information and are interleaved with data blocks on the same bus. The illustration of the architecture is given on the figure 31.

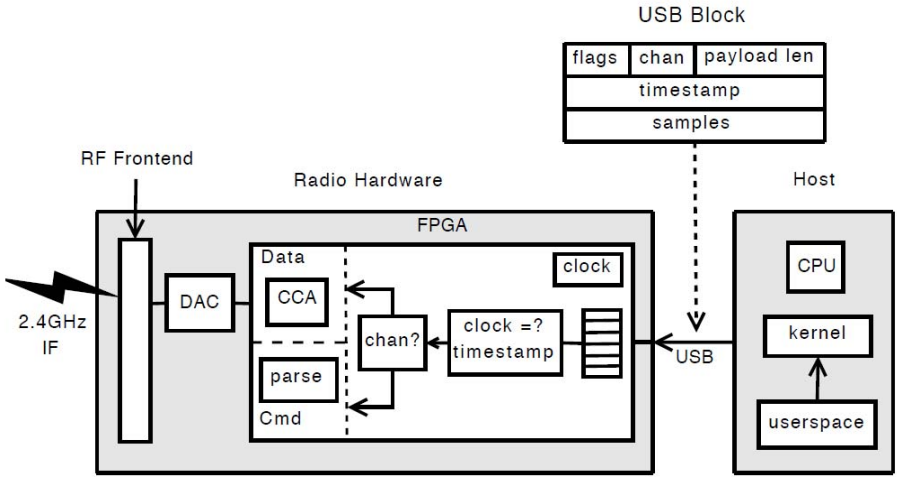


Figure 31.Split SDR architecture [69, fig.2]

As it can be seen from the picture, in this architecture the packets sent from the host are buffered and preceded further when the timestamp of the clock corresponds to the timestamp of the packet, providing precise time scheduling. Depending on the nature of the data, the packet is either transmitted or analyzed by the platform. Since the proof of the concept for the architecture has been implemented on the USRP1 and GNU Radio platforms, the USB is used as the interface between the host and the platform. The performed time measurements for the carrier sense and the precise time scheduling show a latency of some microseconds, an order of magnitude that is compliant with today's wireless standards.

Moreover, a system for fast packet recognition and dependent packets has also been designed. The design is based on having the access to the PHY layer information, i.e. by comparing the sensed RSSI value with a previous determined threshold. Furthermore, a matched filter is used in order to accurately detect the packets without demodulating the signal. The stored coefficients of the filter would

be the known signal, which is the time-reversed complex conjugate of the modulated framing bits. The dependent packages as ACK and RTS/CTS are produced in advance in the host, and as pre-modulated are buffered on the platform. The illustration of this design is shown on figure 32.

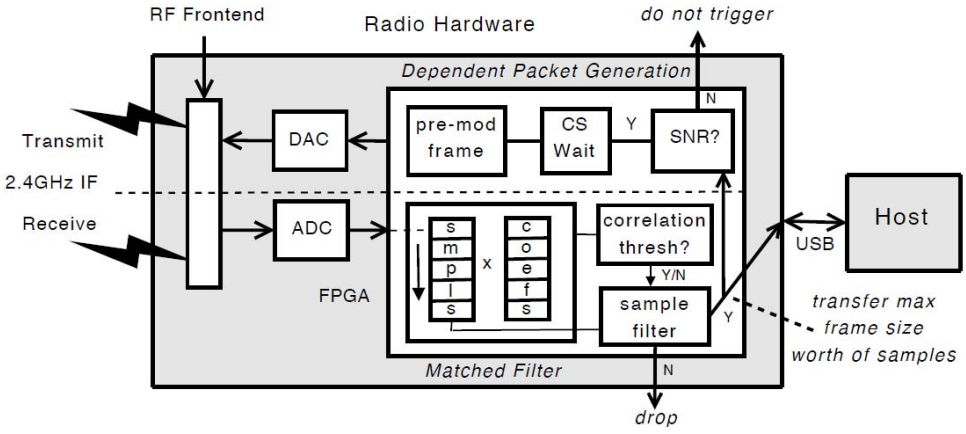


Figure 32. Matched filter & dependent packet design [69, fig.7].

Indeed, the approach of the split-function design is a way of solving the latency problem while achieving high performance, but can be cumbersome since it includes a mixture of both hardware and software design.

6.2 Implementation

In order to evaluate the real performance of the cognitive network by using the 802.11g OFDM PHY standard on the USRP2, the employment of the standard itself on the platform has to be done firstly. In the light of the description of the previous chapter about the related work, the only implementation meeting the strict standard time requirements of the decided standard is the split-function architecture where the time-critical functions are placed on the radio platform.

The used resources on the USRP2's FPGA when using the standard, default FPGA configuration is 37 out of 40 block RAMs, 16 or 18 of the 40 multipliers, and about 35 to 40% of the logic area [70]. In addition, the 1 MB SRAM provided on the motherboard, which earlier was not used at all in the default configuration, is in the

newer FPGA images configured as a TX buffer which offers the opportunity to say "send this data at exactly time X", and accordingly can be used to store the pre-modulated time-stamped packets.

Taking the last two paragraphs in consideration, it was decided to implement the 802.11g ODFM PHY standard by using the split-function architecture despite the fact that FPGA design has to be included. The implementation plan followed the split-function implementation [69] shown on the figures 31 and 32, as well as the MAC finite state machine on the figure 33.

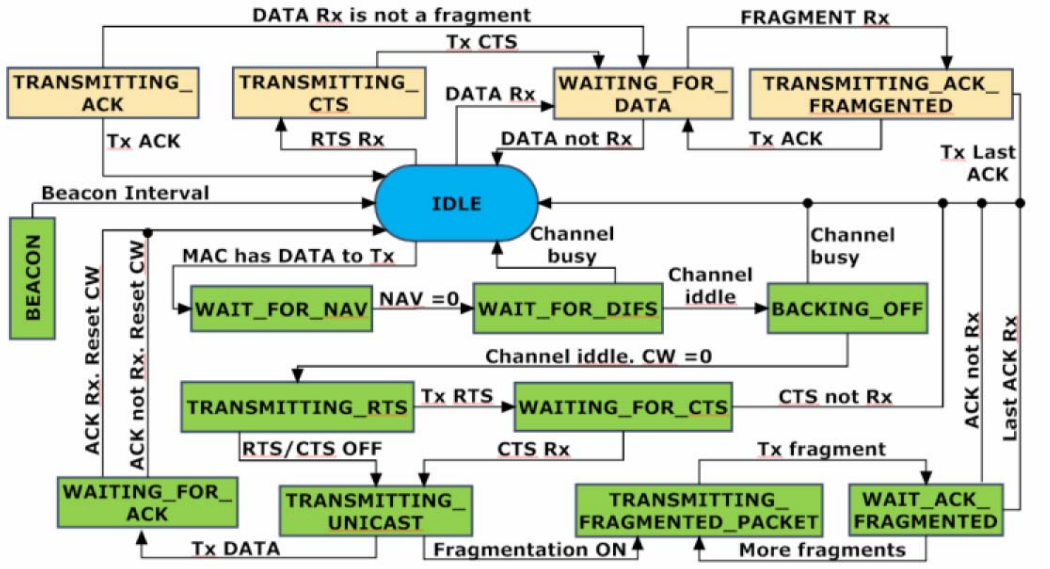


Figure 33. The Finite state machine of 802.11 [63, fig.1]

As it already was explained in the previous subchapter, the USRP2's FPGA will use its 1MB SRAM buffer to temporary store the time- stamped pre-modulated packages which are ready to be sent. In the case of our demonstrator, the data part of the frames is irrelevant, so the predefined dummy data frames will be stored. Moreover, the CSMA/CA will be employed by using the RSSI value from the XCVR2450 daughterboard. The RSSI threshold is defined by the try-and-fail method. However, it is important to be aware of the fact that the RSSI provided by the board is sensed on a wider bandwidth than the signal has when filtered, and may be incorrect. However, no other RSSI values are available in the hardware, but only in software after the signal filtering.

Moreover, the packet loss (PL), a parameter for the primary network which we want to measure, is shown to be highly correlated to the preamble detection loss. In order to simplify the demonstrator, an option is to concentrate only on the detection of the preamble. However, the proposed "reserved bit" difference between the primary and the secondary must be omitted. Although the matched filter is flexible to different modulation schemes (e.g. GMSK, QAM, PSK), for an OFDM signal as in our case it is required to implement an FFT. However, since there are only 9 users in our network, a Look Up Table (LUT) can be implemented with 9 predefined preamble symbols sets, in order to spare the resources on the FPGA. However, this does not give a general solution, but specific for our case. It is also important to be careful when making the choice of the symbol sets and avoid periodicity, in case of which the spectrum will be influenced.

To simplify the demonstrator even further, an option is to have a fast ACK frame rather than occupying the SRAM buffer for the pre-modulated dependent packages ACK and RTS/CTS.

In addition to this, the code of the IEEE 802.11 MAC implemented in GNU Radio was provided kindly by Uwicore [63], and can be used as a reference for our project.

6.3 Results

The product of this project is a full description of a cognitive radio demonstrator. In addition, the installation and introduction manual of the GNU Radio, and a short usage manual on making the FPGA images for the USRP2 are also available as the results of this project, making a building stone on the way towards the full implementation of the cognitive radio on the USRP2.

6.4 Discussion

This report provides a full description of a demonstrator. However, I failed in the effort to employ the real implementation of the demonstrator on the USRP2. There are a few reasons for not completing the implementation of the demonstrator on the USRP2.

Firstly, the amount of work needed to firstly implement the IEEE 802.11g OFDM PHY, and then utilize these to make a cognitive network and make the measurements in the network is huge and is extremely hard to carry out for one person. I have been in contact with one of the contributors of making the split-function architecture [69], and I have been informed that there were two persons working for two years almost exclusively on this project in order to get it done.

Secondly, I did not posses any practical knowledge about any of the programs & OS to be used in the project: the GNU Radio, Python, Xilinx or GNU/Linux when starting with this project. In addition, the sources of learning and explaining the GNU Radio are very sparse with many outdated webpages, and it is very time – demanding to accomplish something if one has to post a question on the discuss-gnuradio@gnu.org, to which I am very thankful, and to wait for an answer. In addition to this, there are very few in the GNU Radio community who change the FPGA image of their SDR, so the resources are very limited. In addition to this, I used a lot of time to learn and explore different programs, but I did not have time to use this new knowledge to get practical results.

Thirdly, the practical things as a simple as installing a program as Xilinx Design Suite took as long as one week because of different practical issues, the lost time one usually does not account when planning the project.

On the whole, the work with this demonstrator should be carried out with more persons involved that have some experience with at least one program listed above. Moreover, the school should provide a dedicated engineer who could work with GNU Radio since it is developing very fast and it is hard to, and help the students in their endeavours.

In order to provide a fast start to my successors on the project, I summarize my experiences and advices in the appendices.

7. Conclusions & Future work

The purpose of this project was to propose a full description of a cognitive radio system demonstrator by using the USRP2 software defined radio. In addition, the proposed demonstrator was to be implemented by using the USRP2 units.

In this report, an overview over DSA and CR is provided. Moreover, the principles of SDR are briefed since these are the building stones of the cognitive radio networks. The report gives an extensive overview of the USRP2 platform, both in hardware and software. In addition to this, the theory behind deploying and managing a radio network is described. Moreover, a full description for a cognitive radio network demonstrator based on the IEEE802.11g with OFDM PHY is given in detail.

The goal with the demonstrator is to show that the introduction of the secondary user in a network will introduce no or negligible degradation to the original, primary network. However, a challenge is to meet the timing demands requested by the chosen standard.

However, due to the dimensions and complexity of the task, and limited time I failed in my effort to complete the implementation of the demonstrator. The products of this project are the installation and introduction manual of the GNU Radio, and a short usage manual on making the FPGA images for the USRP2. In addition, the report includes many advices about the practical work with software, and represents a building stone on a way towards the full implementation of the cognitive radio employing USRP2 SDR platform and the 802.11 OFDM PHY standard.

Future work of course includes the further development of the described demonstrator and its full implementation. However, it demands more time and more people putting effort in it.

References

*All URLs checked on 19th September 2011.

[1] FCC Spectrum Policy Task Force, “Report of the spectrum efficiency working group,” Nov. 2002. URL:http://www.fcc.gov/sptf/reports.html

[2] Islam, M.H.; Koh, C.L.; Oh, S.W.; Xianming Qing; Lai, Y.Y.; Cavin Wang; Ying-Chang Liang; Toh, B.E.; Chin, F.; Tan, G.L.; Toh, W.; , "Spectrum Survey in Singapore: Occupancy Measurements and Analyses," Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on , vol., no., pp.1-7, 15-17 May 2008 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4562457&isnumber=4562434

[3] Shared Spectrum Company; “Spectrum Occupancy Measurements: Chicago, Illinois, November 16-18, 2005”, December 2005. URL:http://www.sharedspectrum.com/papers/spectrum-reports/

[4] Shared Spectrum Company; “General Survey of Radio Frequency Bands (30 MHz to 3 GHz): Vienna, Virginia, September 1-5, 2009”, September 2010. URL:http://www.sharedspectrum.com/papers/spectrum-reports/

[5] Cabric, D.; Mishra, S.M.; Brodersen, R.W.; , "Implementation issues in spectrum sensing for cognitive radios," Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on , vol.1, no., pp. 772- 776 Vol.1, 7-10 Nov. 2004 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1399240&isnumber=30419

[6] Qing Zhao; Sadler, B.M.; , "A Survey of Dynamic Spectrum Access," Signal Processing Magazine, IEEE , vol.24, no.3, pp.79-89, May 2007 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4205091&isnumber=4202144

[7] Mitola, J., III; , "Cognitive radio for flexible mobile multimedia communications," Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on , vol., no., pp.3-10, 1999 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=819467&isnumber=17761

[8] Mitola, J., III; , "Software radios: Survey, critical evaluation and future directions ," Aerospace and Electronic Systems Magazine, IEEE , vol.8, no.4, pp.25-36, Apr 1993 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=210638&isnumber=5467

[9] CR principle; URL:http://3.bp.blogspot.com/_hZxRbcx_kjY/TRMnFD79nOI/AAAAAAAAACk/4UXlkZXT5tg/s1600/cognitive%2Bradio.gif

[10] C. Santivanez, R. Ramanathan, C. Partridge, R. Krishnan, M. Condell, and S. Polit. 2006. “Opportunistic spectrum access: challenges, architecture, protocols”. In Proceedings of the 2nd annual international workshop on Wireless internet (WICON '06). ACM, New York, NY, USA, , Article 13. URL:http://portal.acm.org/citation.cfm?id=1234161.1234174#

[11] Akyildiz, I.F.; Won-Yeol Lee; Vuran, M.C.; Mohanty, S.; , "A survey on spectrum management in cognitive radio networks," Communications Magazine, IEEE , vol.46, no.4, pp.40-48, April 2008 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4481339&isnumber=4481327

[12] FCC, “SECOND REPORT AND ORDER AND MEMORANDUM OPINION AND ORDER”, Nov.2008. URL:http://www.wispa.org/wp-content/uploads/2008/11/FCC-08-260A1(2ndR0-111408).pdf

[13] SENDORA (**SE**nsor **N**etwork for **D**ynamic and c**O**gnitive **R**adio **A**ccess) project; EU FP7 January 2008-December 2010; URL:http://www.sendora.eu

[14] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran y S. Mohanty, “NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A survey”, Computer Networks, vol. 50 nº 13, pp. 2127-2159, Sep. 2006. URL:http://www.sciencedirect.com/science/article/pii/S1389128606001009

[15] D7.7 ; “Performance of SENDORA networks”, SENDORA FP7-ICT-2007-1.1; Jan. 2011 URL:http://www.sendora.eu/system/files/SENDORA(216076)_D77_0.pdf

[16] S. Gu, P. Xu, X. Wang, X. Gan, and H. Yu, "A Real Time Testbed for the Evaluation of Cognitive Radio MAC", in Proc. GLOBECOM, 2010, pp.1-5. URL:http://www.csee.usf.edu/~labrador/Share/Globecom/DATA/01-008-04.PDF

[17] McHenry, M.; Livsics, E.; Thao Nguyen; Majumdar, N.; , "XG Dynamic Spectrum Sharing Field Test Results," New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on , vol., no., pp.676-684, 17-20 April 2007 doi: 10.1109/DYSPAN.2007.90 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4221552&isnumber=4221462

[18] GNU Radio; URL:http://gnuradio.org/

[19] The Comprehensive GNU Radio Archive Network (CGRAN); URL:https://www.cgran.org/

[20] D7.2 ; “Demonstration Specifications”, SENDORA FP7-ICT-2007-1.1; Jan. 2011; URL:http://www.sendora.eu/node/116

[21] OpenAirInterface; URL:http://www.openairinterface.org/

[22] Lee K. Patton; MSc“A GNU Radio Based Software-Defined Radar”; Department of Electrical Engineering; Wright State University; 2007 URL:http://etd.ohiolink.edu/view.cgi?wright1176142845

[23] Ronan Farrell, Magdalena Sanchez, and Gerry Corley, “Software-Defined Radio Demonstrators: An Example and Future Trends,” International Journal of Digital Multimedia Broadcasting, vol. 2009, Article ID 547650, 12 pages, 2009. doi:10.1155/2009/547650 URL:http://www.hindawi.com/journals/ijdmb/2009/547650/

[24] OSI Basic Reference Model; URL:http://www.ns-linux.org/Uputstva/Teorija/slike/osi-model-7-layers.png/view

[25] Ulversoy, T.; , "Software Defined Radio: Challenges and Opportunities," Communications Surveys & Tutorials, IEEE , vol.12, no.4, pp.531-550, Fourth Quarter 2010 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5462981&isnumber=5638591

[26] Gustafsson, O.; Amiri, K.; Andersson, D.; Blad, A.; Bonnet, C.; Cavallaro, J.R.; Declerck, J.; Dejonghe, A.; Eliardsson, P.; Glasse, M.; Hayar, A.; Hollevoet, L.; Hunter, C.; Joshi, M.; Kaltenberger, F.; Knopp, R.; Le, K.; Miljanic, Z.; Murphy, P.; Naessens, F.; Nikaein, N.; Nussbaum, D.; Pacalet, R.; Raghavan, P.; Sabharwal, A.; Sarode, O.; Spasojevic, P.; Yang Sun; Tullberg, H.M.; Vander Aa, T.; Van der Perre, L.; Wetterwald, M.; Wu, M.; , "Architectures for cognitive radio testbeds and demonstrators — An overview," Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM), 2010 Proceedings of the Fifth International Conference on , vol., no., pp.1-6, 9-11 June 2010 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5577684&isnumber=5577658

[27] "IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless Lans," IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007) , vol., no., pp.c1-222, June 12 2008 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4544755&isnumber=4544754

[28] Newman, T.R.; Hasan, S.M.S.; Depoy, D.; Bose, T.; Reed, J.H.; , "Designing and deploying a building-wide cognitive radio network testbed," Communications Magazine, IEEE , vol.48, no.9, pp.106-112, Sept. 2010 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5560594&isnumber=5560574

[29] The Global Environment for Network Innovations (GENI); URL:http://www.geni.net/

[30] PlanetLab; URL:http://www.planet-lab.org/

[31] Ettus Research LLC; URL:http://www.ettus.com/

[32] USRP2, front face;
ULR:http://www.olifantasia.com/gnuradio/usrp/files/usrp2.jpg

[33] Bruhtesfa Ebrahim Godana ;"GNU Radio and USRP Manual"; August, 2010

[34] USRP2 motherboard datasheet;
URL:http://www.ettus.com/downloads/ettus_ds_usrp2_v5.pdf

[35] USRP2 daughterboards;
URL:http://www.ettus.com/downloads/ettus_daughterboards.pdf

[36] Pyhon; URL:http://www.python.org/

[37] C++; URL:http://www.cplusplus.com/

[38] Simplified Wrapper Interface Generator (SWIG); URL:http://www.swig.org/

[39] Simulink; URL:http://www.mathworks.com/products/simulink/

[40] LabVIEW; URL:http://www.ni.com/labview/

[41] Minseok Kim, ``Prototyping and Evaluation of Software Defined Radio using GNU Radio-USRP," IEICE Technical Report, SR2010-25, Kyoto, Jul. 2010
URL:http://www.ap.ide.titech.ac.jp/~mskim/

[42] GNU Radio Companion (GRC); URL:http://www.joshknows.com

[43] GNU General Public License version 3 (GNU GPLv3);
URL:http://www.gnu.org/licenses/quick-guide-gplv3.html

[44] Firas A.Hamza; "USRP under 1.5 magnifying lens", GNU Radio community, June 2008.
URL:gnuradio.org/redmine/attachments/129/USRP_Documentation.pdf

[45] Andrea Goldsmith; "WIRELESS COMMUNICATIONS", Stanford University; 2005

[46] Multiple Access Methods;
URL:http://dolcera.com/wiki/index.php?title=Image:Cdma12.jpg

[47] Hidden and exposed node problem;
URL:http://www.cse.iitk.ac.in/users/dheeraj/cs425/fig.lec05/image002.gif

[48] Explained solution for exposed node;
URL:http://pdos.csail.mit.edu/decouto/papers/bharghavan94.pdf

[49] Yucek, T.; Arslan, H.; , "A survey of spectrum sensing algorithms for cognitive radio applications," Communications Surveys & Tutorials, IEEE , vol.11, no.1, pp.116-130, First Quarter 2009 doi: 10.1109/SURV.2009.090109 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4796930&isnumber=4796921

[50] Won-Yeol Lee; Akyildiz, I.F.; , "Optimal spectrum sensing framework for cognitive radio networks," Wireless Communications, IEEE Transactions on , vol.7, no.10, pp.3845-3857, October 2008 URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4657330&isnumber=4657304

[51] Star-network topology; URL:http://en.wikipedia.org/wiki/File:NetworkTopology-Star.png

[52] Partially connected mesh topology; URL:http://en.wikipedia.org/wiki/File:NetworkTopology-Mesh.png

[53] BTnodes; URL:http://www.btnode.ethz.ch/

[54] Mica2; URL:http://webs.cs.berkeley.edu/tos/mica2.html

[55] TinyOS; URL:http://www.tinyos.net/

[56] BTnodeRFID reader; URL:http://btnoderfid.sourceforge.net/Hardware.html

[57] wirelessHART; URL:http://www.hartcomm.org/protocol/wihart/wireless_technology.html

[58] IEEE 802.11-2007. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. URL:http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[59] A.F. Molisch; “Wireless Communications”, John Wiley & Sons, Ltd; 2005

[60] Bob O’hara , Al Petrick ; “The IEEE 802.11 Handbook: A Designers’s Companion”, Standards Information Network, IEEE Press 1999

[61] Channel deployment of 802.11g standard in ISM band; URL:http://en.wikipedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg

[62] P. Fuxjäger, et al., "IEEE 802.11p Transmission Using GNURadio”, in Proceedings of the IEEE Karlsruhe Workshop on Software Radios (WSR10), pp. 1-4, 2010. URL:http://userver.ftw.at/~cpaolo/Publications/wsr10.pdf

[63] J.R. Gutierrez-Agullo, B. Coll-Perales and J. Gozalvez, "An IEEE 802.11 MAC Software Defined Radio Implementation for Experimental Wireless Communications and Networking Research", Proceedings of the 2010 IFIP/IEEE Wireless Days (WD'10), 20-22 October 2010, Venice (Italy). URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5657724&isnumber=5657694

[64] K. Mandke, et al., “Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed”, in Proceedings of the IEEE Vehicular Technology Conference (VTC07), pp. 1896-1900, 2007. URL:http://www.profheath.org/research/mimo-system-prototyping/hydra-mimo-ad-hoc-network-phase-2/

[65] Click; URL:http://read.cs.ucla.edu/click/click

[66] TUN/Tap;URL:http://www.kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/tuntap.txt

[67] Zhifeng Chen, “Real-world Transmission with GNU Radio and USRP”; URL:http://www.wu.ece.ufl.edu/projects/wirelessVideo/project/GNU_Radio_USRP/index.htm

[68] Wireshark packet sniffer; URL:http://www.wireshark.org

[69] George Nychis , Thibaud Hottelier , Zhuocheng Yang , Srinivasan Seshan , Peter Steenkiste, “Enabling MAC protocol implementations on software-defined radios”, Proceedings of the 6th USENIX symposium on Networked systems design and implementation, p.91-105, April 22-24, 2009, Boston, Massachusetts; URL:http://dl.acm.org/citation.cfm?id=1558984

[70] USRP2 FPGA’s free resources; URL:http://gnuradio.org/redmine/projects/gnuradio/wiki/USRP2GenFAQ

Appendices

Appendix A: GNU Radio Introduction

A1. Install the GNU Radio

1. There are different ways of installing the GNU Radio software, and all can be found on: http://gnuradio.org/redmine/projects/gnuradio/wiki/BuildGuide. However, absolutely the easiest way to install it is by running the script made by Marcus D. Leech, which is available on http://www.sbrac.org/files/build-gnuradio. This script installs all the needed prerequisites, the latest versions of the Universal Hardware Driver (UHD) and GNU Radio, downloads the latest FPGA image and regulates some small additional things. It also removes any old GNU Radio installation. It can be used on the Fedora and Ubuntu OS. Even if the script is not compatible with your OS, it is recommended to follow the steps if the script in order to get a good and clean installation of the programme. All the following explanations in this appendix is supposing that the installation has be done by the script.

2. It can happen that your installation just stops. However, if you are known to Linux and go through the script, you will find that the outputs and errors of almost all of the processes are directed towards >/dev/null. This means that even if you get any error or you get prompted by installer about anything, you do not see it. It is quite nice property when everything goes as expected. The solution is to find the place in the script where the installer has stopped, and install these components separately without >/dev/null 2>&1. So if the script stopped at (an example):

sudo apt-get purge ‘gnuradio-*’ >/dev/null 2>&1

You write in the terminal:
sudo apt-get purge ‘gnuradio-*’

Now you get all the process outputs printed on the screen, and you can understand what went wrong. Do this each time if the script stops. After it is done successfully, try again with the script from the beginning, and everything should function and install properly.

3. The GNU Radio can be installed alone, including the default Raw Ethernet driver or with the UHD. When installed alone, the interface between the host PC and the GNU Radio is raw Ethernet. In this case, the ping command for the USRP is **sudo find_usrps**. However, when the UHD is installed, as the used script does, the utilized interface between the host PC and the GNU Radio is UTP. In addition, the UHD provides easy implementations with Matlab and LabView. In this case, the ping command for the USRP is **uhd_find_devices.** When using the UHD driver, you should not forget to set the static IP on the host, and the details can be found in the reference below. However, since the UHD is quite a new driver, the vast of applications will not apply to it since it is made for the default driver. This includes also the code examples provided with the installation. However, all the new users are encouraged to start with the UHD. For more information about the UHD, see the URL: http://code.ettus.com/redmine/ettus/projects/uhd/wiki.

If you really insist of installing the version with using raw Ethernet, you would have consult http://gnuradio.org/redmine/projects/gnuradio/wiki/BuildGuide.

In general, all the needed information about the GNU Radio is present on the www.gnuradio.org, but many pages are outdated and the place of information is being changed, so you should really know where things are in order to find them. In addition, many things described are hard to understand, so the threshold of starting with GNU Radio is quite high.

A2. Explore the GNU Radio

1. If one really wants to understand the USRP2's architecture, configuration and operation, there is a small list of some important trunk USRP2 code which is recommended for reading. In the FPGA code, the tx_control.v and rx_control.v show inband signalling, dsp_core_rx and dsp_core_tx show the DSP, and u2_core.v is the top level. In the firmware, start with txrx.c. These can be found in usr/local/share/uhd/fpga and usr/local/share/uhd/firmware, respectively.

2. Moreover, the SD card image contains two distinct images that are loaded into the memory - the FPGA image, and the firmware image. The firmware image configures the FPGA pins and creates a "Soft CPU" (used to be an aeMB, but is now a ZPU), which runs the firmware that manages a lot of the low-bandwidth configuration things related to the hardware, while the FPGA image configures the rest of the FPGA which handles DSP operations, and actual high-speed data transmission from/to the DSP and Ethernet.

3. The script saves the latest pre-built images in the usr/local/share/uhd/images, each in their own sub-directory "firmware" and "fpga". The source code and the images have to match in order to have the proper function of the code. If you want to "flash" or "burn" the FPGA and firmware image to the SD card, the easiest way is to use the GUI file found in /usr/local/share/uhd/utils/usrp2_card_burner_gui.py.

Note: It is important that you are careful when doing this, since choosing the wrong disk can result in deleting the hard drive. Useful commands which can help you to pick the right disk are **dmsg** and **fdisk –l**. Moreover, the two images, the firmware and the FPGA are burned separately.

4. There are very few proper introduction manuals to start with the GNU Radio. However, the site http://www.csun.edu/~skatz/katzpage/sdr_project/sdr/ provides a nice introduction of the GNU Radio Companion, the "graphical version" of the GNU Radio.

Appendix B: Xilinx Introduction

B1. Install Xilinx Design Suite

If you want to build your own FPGA image for the USRP2, you need a licensed version of the Xilinx design software which was provided by a university's engineer. The GNU Radio provides "make"-files which are designed to easily build the FPGA images. However, be aware that these "make"- files are optimized for GNU/Linux, but the building could be functioning on other OS too. However, I myself experienced big problems to get it function on Windows, and went over to GNU/Linux.

To install Xillinx in GNU/Linux, follow the guidelines from the Xilinx (http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_4/irn.pdf):

1. RUN the xsetup file

2. Set up your environmental variables by typing to your terminal (32 or 64 bits):
source settings32.(c)sh OR source settings64.(c)sh

3. Include Xilix xtclsh in the path in order to be able to build FPGA images. To do this, add the following to the .bashrc:

```
XILINX_ROOT=/opt/Xilinx/12.4/ISE_DSexport
PATH=${XILINX_ROOT}/ISE/bin/lin:${PATH} export
PATH=${XILINX_ROOT}/ISE/bin/lin/unwrapped:${PATH}
```

4. Follow the instruction document for adjusting the licenses. If you are using a Floating License which is on a server, then you have to do point towards the license on a way that is NOT described in the Xilinx document. In my case, I had to use the VPN client to the server, and in addition to add one enviromental variable to the system. This can be done by adding the following line in, e.g. .bashrc:

```
LM_LICENSE_FILE=port_nr@....com
EXPORT LM_LICENSE_FILE
```

An example:

LM_LICENSE_FILE=1710@fysel-vault.fysel.ime.ntnu.no
EXPORT LM_LICENSE_FILE

Note 1: As per today, the Raw Ethernet FPGA images can only be built with Xilinx ISE 10.x, while the UDP ethernet FPGA images can only be built by using ISE 12.x.

Note2: To add the environmental variables in Windows, do the following: right click on my comp. → System Properties → Advanced → Enviromental Variables →Add. Add a new variable called LM_LICENSE_FILE=... and add the Xilinx xtclsh to the path.

B2. Build the FPGA image

In order to build an FPGA image with the make files the GNU Radio provides, go in the folder where your top code is, and write in the terminal:

make proj
make bin

Examples:
~/bin/uhd/fpga/usrp2/top/USRP2$ make proj
~/bin/uhd/fpga/usrp2/top/USRP2$ make bin

These two commands will provide you with your FPGA image.

To burn the image on the SD-card, see A2.