creative commons
COMMONS DEED

저작자표시-비영리-변경금지 **2.0** 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:48009-200000102932

A TESTBED FOR SPECTRUM SENSING IN COGNITIVE RADIO NETWORKS

DISSERTATION

for the Degree of

MASTER OF SCIENCE
(Electrical Engineering)

TENDENG RENE

MAY 2018

A Testbed for Spectrum Sensing in Cognitive Radio Networks

DISSERTATION

for the Degree of

Master of Science
(Electrical Engineering)

TENDENG RENE

May 2018

A Testbed for Spectrum Sensing in Cognitive Radio Networks

DISSERTATION

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

MASTER OF SCIENCE
(Electrical Engineering)

at the

UNIVERSITY OF ULSAN

by

Tendeng Rene
May 2018

Publication No.______________

©2018 - Tendeng Rene

All rights reserved.

A Testbed for Spectrum Sensing in Cognitive Radio Networks

Approved by Supervisory Committee:

Prof. Hyung-Yun Kong , Chair

Insoo Koo

Prof. In-Soo Koo, Supervisor

Prof. Vladimir V. Shakhov

Department of Electrical and Computer Engineering
University of Ulsan, South Korea
Date: May , 2018

ACKNOWLEDGMENTS

Praise be to God the Almighty for abundant blessings poured into my heart at different levels of my studies. I believe all the successes I obtained in my life are not by my own powers, but they are graces from God.

My deepest gratitude goes first and foremost to my supervisor, Professor IN SOO KOO, for introducing me to the very interesting field of wireless communication network. I am greatly indebted to him for his kindness, constant encouragement and guidance which offered me valuable suggestions in the academic studies. Without his patient instruction, insightful criticism and expert guidance, the completion of this thesis would not have been possible.

Special thanks also go to Dr. Lee and Dr. Hiep for their illuminating instructions, discussion and supports, I would like also to thank all the lab members of Multimedia Communication System Lab (MCSL) for their encouragements and my friends around the world, specially my best friend Marie Pierre Coly for her support, confidence and strong encouragement every day when I felt frustrated during the research process.

I also owe a debt of gratitude to all the professors, in electrical engineering, from whose devoted teaching and enlightening lectures I have benefited for my research. I am also grateful to the department of Electrical Engineering.

Next, I would also like to acknowledge NIIED program which has been financially supporting me throughout the course of my studies.

Finally, I would also like to thank my deceased parents and my family specially my lovely sister Tina,Toni and their children Annabel and Alex, they were always supporting me and encouraging me with their best wishes during all my study process .

Tendeng Rene

Ulsan, South Korea, May - 2018.

ABSTRACT

A Testbed for Spectrum Sensing in Cognitive Radio Networks

by

Tendeng Rene

Supervisor: Prof. Insoo Koo

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science (Electrical Engineering)

May 2018

Demand of wireless communication has been increasing rapidly in data transmission due to rising of the wireless services, devices and applications. Regarding such a context, the radio resource for supporting the demand is seriously lacked. In the coming decades, this problem will deteriorate due to an explosive growth of IoT (Internet of thing). Meanwhile, the spectrum allocation policy in which government agencies assign static spectrum to licensed users (or primary users (PUs)), leads to inefficient utilization of a large amount of licensed spectrum.

Due to under-utilization of radio spectrum resource, the concept of cognitive radio networks has been introduced by Mitola and considered as an alternative, that can be used to exploit the existence of unused licensed frequency bandwidth. In this concept the unlicensed users (or secondary user(SUs)) can sense the spectral environment to dynamically and opportunistically access the spectrum bands in absence of primary user at a particular time and specific geographic location.

In general, Pus have not been very receptive of the idea of opportunistic spectrum sharing. In particular, they are concerned that cognitive radio (CR) will harmfully interfere with their operation. There is considerable debate whether it is possible to build a CR that does not disturb PUs. However, this debate cannot be resolved on a theoretical basis. As it is impossible to test all possible cases, it is necessary to commonly agree on a set of representative test cases that a CR must pass to prove that the amount of interference is sufficiently low to justify allowing CR technology. There is a plethora of techniques such as cooperative sensing, Cyclostationary detectors, energy detection etc. that have been proposed to enhance detection. None of these techniques have been tested all in real world scenarios and their performance has yet to be characterized. Thus, there is a need for experimenting with different techniques in a real system, using a set of test cases and metrics to compare different CR implementations. In this research, we cover the details of such experiment testbed of spectrum sensing for CR based on energy detection algorithm and a software defined radio (SDR). This testbed has been built on GNU radio platform using the universal software radio peripheral (USRP) NI-2900 supported by National Instrument to evaluate the sensing performance of energy detection in a real environment. We also implement a cloud server-based spectrum sensing such that we can store, process, and share the sensing information more efficiently in the centralized way at the cloud server. Finally, we do implementation of a real LoRa network based on LoRaWAN protocol for Internet of thing (IoT), create a gateway and build an application, then process the data to the cloud server.

Contents

List of Figures

Nomenclature

Notation	**Description**
CR	Cognitive Radio
SDR	Software defined radio
SU	Secondary User
USRP	Universal software radio peripheral
IoT	Internet of things
PU	Primary User
OFDM	Orthogonal frequency division multiplexing
FCC	Federal Communications Commission
SNR	Signal-to-Noise Ratio
ISM	Industrial, Scientific and Medical
SS	Spectrum Sensing
ED	Energy Detection
FFT	Fast Fourier transform
UHD	USRP Hardware Drive
ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
AWGN	Additive white Gaussian noise
NI	National Instruments
I/Q	In-phase/quadrature
RF	Radio frequency
GRC	GNU Radio Companion
SMA	SubMiniature version A
DDC	Digital down converter
DUC	Digital up converter
FIR	USRP Hardware Drive
TTN	The Things Network
SF	Spread Factor
CR	Coding Rate
RSSI	Received Signal Strength Indicator
SNR	Signal to Noise Ratio

Chapter 1

Introduction

1.1 Motivation and Background

Wireless communication is one of the exciting areas in communication field today. Its usage has increased due to the need of mobility and convenience, in the last decade. Recently due to the exponential growth of wireless communication system because of their higher speed of data transmission, there's a need to use the spectrum resources as efficiently. Although there are great improvements in wireless communication's technical level, the shortage of available spectrum is considered as a vital problem. Wireless communication system transfers information through electromagnetic waveforms with different wavelength which have different propagation characteristics, and different wireless application use different frequency range (wavelength) for communication according to their requirements, [34]. The allocation of spectrum for each application is regulated by the Federal Communication (FCC) and International Telecommunication Union (ITU). The spectrum can be considered as a scarce resource due to the rapid growing number of users in various wireless communication applications, such as mobile phone and wireless data communications [35]. In order to fulfill the demand of increasing number of users, higher data transmission speed for new applications, as well as better mobility and security during the communication, there has been tremendous interest in the field of cognitive radio (CR) concept, which has been introduced by Mitola, [1]. CR is an enabling technology that allows cognitive radio users (SUs) to operate in the licensed (Pus) spectrum band. The most distinct characteristic of CR system is the ability to identify the presence or absence of unknown deterministic signals, and determine whether a PU active or not, based on the identification process. In this approach, the SUs proceeds for detecting the unoccupied spectrum bands or remains silence according to the decision of availability of spectrum and they should vacate the spectrum as soon as the primary licensed (PUs) begin its activities [2]. This is an effective way and solution to overcome the current scarce spectrum resource problem and achieve significant improvement over services offered by current wireless networks. To allow reliable operation of cognitive radio, we must be able to detect precisely the spectrum holes at link level, which gives spectrum sensing a critical role.

Spectrum sensing (SS) is a method used for observing the spectrum holes within the interesting frequency band. To detect the presence or absence of PUs several spectrum sensing techniques has been developed [3, 4]. among them, energy detection, matched filter Cyclostationary detection, wavelet detection etc. In real practical implementation energy detection is by far the simple and most widely used due to their low computational complexity. ED based analyses are commonly formulated as a Neyman-Pearson binary hypothesis test, each of the two hypothesizes represents a simplistic scenario: the interesting frequency band under sensing is either only noise or is occupied by a PU with noise, both having constant power spectral density (PSD). The main disadvantage of energy detection concept is its dependence on the knowledge of the noise variance [5]. The performance of ED decreases significantly due to the effect of the noise uncertainty. To overcome this limitation under noise uncertainty scenarios, several improvement have been developed. One of them uses the information of maximum and minimum value of the sub-band energies within the sensing band, and it exhibits more robust performance than the regular ED sensing algorithm. Also, multi-antenna sensing, and cooperative sensing perform well. However, they are impractical for several applications due to the drawback of increased hardware complexity and size.

Our thesis focused on the performance of ED based spectrum sensing algorithms and their implementation in real environment. The aim of this thesis is to implement a real-time CR system prototype with different ED based sensing algorithm such as FFT with frequency averaging window. The study focuses on Software Define Radio (SDR) platform which has been used in recent years by wireless communication researchers and developer to implement new protocols, methods or techniques. The SDR approach could help researcher and engineers to reduce the time effort and cost to implement a wireless system. One of the most popular SDR is the Universal Software Peripheral Radio (USRP). The advantage of SDR technique motivate us to implement a testbed for spectrum sensing. We are using the ISM band for transmitting signals to test the SS algorithms in a controllable manner. The physical experimental platform differentiates this project from many preceding researches which focus on simulated results.

In this thesis, our testbed system is divided into two parts, the transmitter and the receiver. The transmitter is implementing using a PC, GNU radio and NI-USRP. The receiver is implementing using laptop GNU radio and NI USRP-2900, to quantify the spectrum availability. Finally, we provided a comprehensive literature of LoRa concept and implementation of LoRa network.

1.2 Thesis Outline

The first chapter provides introduction and objectives of this research. In the second chapter, the concept of cognitive radio and its basic properties are introduced. This chapter also includes the details of SS algorithms used in this implementation. The sensing algorithms are explained with analytical models and the analytical performance of different sensing algorithms is briefly discussed. The third chapter introduces background information and related work to real-time SS implementation, which provides and explanation of the concept of software defined radio system and basic information about NI USRP hardware devices used in this research. The fourth chapter presents the implementation details of ED based SS. It provides the details information about how the transmitter and receiver are implemented, with graphs and instructions. In this Chapter, the real time-test results of different sensing algorithms with different channel models are discussed and shown in figures. The measurement results are compared with simulation results providing SNR versus detection probability, in order to show the detection performance with different SNR values. The effects of different channel models are also discussed in the same chapter.

Chapter 5 presents the literature and Implementation of LoRa network. Chapter 6 provides a brief conclusion based about possible future improvements of this implementation and future research topics.

Chapter 2

Cognitive radio Introduction and Spectrum Sensing Algorithms

2.1 Introduction

In the last decade the usage of wireless data communication and its applications has grown rapidly. Although there are a greats improvement in wireless data communication's technical level, so far, the problem of shortage of available spectrum is still a vital stifling for the future development of wireless data communication Technologies. The spectrum allocation currently used can reach for limited utilization of spectrum as for each application the uses spectrum allocation is fixed. In order to make use of the unoccupied spectrum, advanced methods are needed so that better spectrum utilization can be reached. Due to this demand the concept of cognitive radio (CR) has been introduced [6]. In cognitive radio Spectrum Sensing is one of the most important operations of CR devices, it informs the system about which part of the specified frequency band are not occupied by the licensed Primary Users (Pus). Hence, unlicensed secondary users (SUs) may initiate communication using some of the spectral holes. There are various spectrum sensing methods which use different mathematical models for different scenarios. The goal of this chapter is to introduce the general and basic idea of CR systems and SS methods used in this thesis.

2.2 Fundamental concept of Cognitive Radio Networks

In order to solve the problem of spectrum shortage and increase the data transmission rate, the concept of cognitive radio has been proposed by Joseph Mitola [6]. Cognitive Radio is defined by FCC as a radio or system that senses its operational electromagnetic environment and can dynamically and autonomously adjust its radio operating parameters to modify the system operation, such as maximize throughput, mitigate interference, facilitate interoperability, access secondary markets [6]. The concept of CR is now widely recognized and there is a considerably large amount of studies and research of CR systems also several applications. The most distinctive difference of CR system with the traditional wireless communication system is its ability of identifying the presence or absence of unknown deterministic signals and determine whether a PU signal is active or note based on the SS process. SUs can either proceed to use the unoccupied spectrum or remain silent according to the decision regarding spectrum occupancy. In this way, the CR system can determine the spectral holes within a frequency band and allowing the unlicensed users to access those spectral holes opportunistically. This is an effective way of spectrum utilization and a good solution for the current scarce spectrum resource problem. [6,8,17]. The aim of CR is to achieve dynamic spectrum access instead of fixed spectrum allocation and having better spectrum utilization.

In order to achieve the basic idea of spectral reuse, which gives the SUs permission to operate over spectrum that is allocated to licensed Pus when the specified spectrum is not

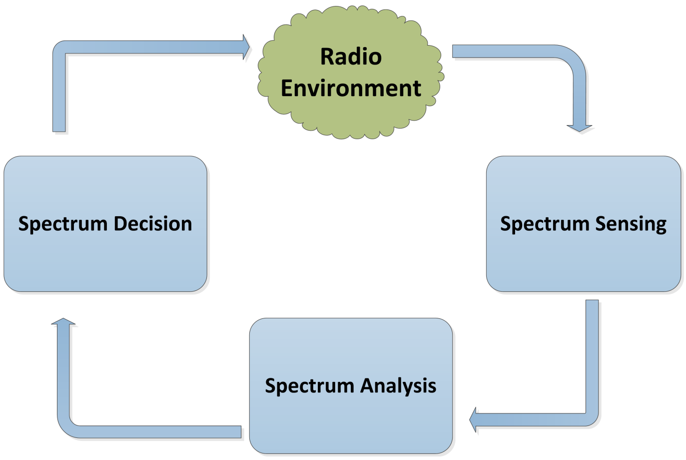


Figure 2.1: Cognitive radio operation.

fully occupied, there are several tasks a CR device should fulfill. These tasks and operation are usually defined as the basic cognitive cycle, which can be seen in **Figure. 2.1**. There are three basic cognitive tasks as seen in the figure. First the available spectrum is sensed and analyzed to find any available spectrum holes. On the basis of spectrum analysis, a decision is made to opportunistically assign the available frequency to the SU. Spectrum sensing is the most integral part of CR because all remaining operations rely on precise sensing of available spectrum. In this thesis, the focus is on the SS part.

2.3 Fundamental concept of Spectrum Sensing

The main duty of spectrum sensing is to obtain awareness of the spectrum usage of licensed PUs in the specified frequency band. SUs can decide whether there are spectral holes available or not based on the information provided by the SS operation. The frequency spectrum can be categorized into three different types according into the activity on sub-bands, namely white space, grey space and black space [3]. Unoccupied bands only contain noise and they are called white space. Partially occupied are called grey spaces and bands which are fully occupied are called black spaces. Spectrum sensing method can be used to determine white and grey spaces, hence the SUs are able to use the specified spectrum if not occupied. The most important task of SS operations is to prevent the PU from being interrupted by SUs. The primary users have absolute rights to use the assigned frequency band. SUs, using CR technology are able to sense and occupied the spectral holes when the frequency bands are not occupied by the PU. It is very crucial to keep sensing for the presence of PU and release the frequency spectrum occupied by cognitive users immediately after the presence of PU is sensed, hence the interruption can be avoided. Spectrum sensing technique is still under developed, there are several challenges and topics to be researched. Current SS techniques make use of time, frequency and space domains. In real-world scenario, the PU signals are expected to be detected under very low SNR case. The mobile channel effects such as multipath fading and frequency selectivity can also introduce additional challenges. An effective spectrum sensing method needs to have a good balance between sensing accuracy, sensing time, and the implementation complexity of the method. There are several different sensing methods with different emphasis on the balance of implementation complexity and accuracy. These sensing techniques can be classified into three categories according to the amount of information they require: blind, semi-blind and non-blind sensing [3]. Blind sensing methods such, as eigenvalue and covariance-based detection, and wavelet-based detection refers to detection schemes that require no information about the Primary user system and the noise distribution. [19–22,33]. Hence, wavelet-based detection is often preferred to be selected as the first stage of a multi-stage detection schemes. Semi-blind sensing detection scheme such as energy detector require only the information of power noise variance [23–26]. Energy detection simply measures the energy summation of received signal in either the time or frequency domain. However, its performance is limited by the SNR wall due to noise or/and system uncertainty.

Non-blind sensing schemes such as, Cyclostationary detection and matched-filter detection need prior information about the PU signal and can provide good timing and frequency synchronization. Matched-filter detection is optimal in the sense of complexly know data sequence detection. However, it is very sensitive to frequency offset.

Spectrum sensing using ED scheme for CR in real environment is studied and implemented using GNU radio and NI-USRP platform in this thesis.

2.4 Spectrum sensing techniques

The first task of spectrum sensing is detecting the primary user signal. Various techniques to sense the spectrum hole are available in the literature [3,4].The most common detection techniques used for spectrum sensing are:

2.4.1 Matched Filter Detection

The matched filter (MF) is a filter designed to maximize the output of signal to noise ratio for a given input signal. When secondary user has a priori knowledge of primary user (PU) signal, matched filter detection is applied. Matched operation is equivalent to correlation in which the unknown signal is convolved with the filter whose impulse response is mirror and time shifted version of a reference signal. Matched filter detection method is commonly used in Radio Detection and Ranging (RADAR) communication [7].

2.4.2 Energy Detection

Energy Detection (ED) is a non-coherent method that detect the primary signal based on the sensed energy. [8]. ED based SS algorithm is widely the most used sensing methods due to their low computational and implementation complexity compared to other sensing methods. Through Energy detection method, we measure the energy of available radio resource and compare it against a predefined threshold level. If the measured energy level is below the threshold level, then spectrum is marked as available otherwise it considered as occupied when the energy level is above the threshold. ED method does not need any requirement on a prior knowledge of PU signal. It only relies on the energy received in the band, if PU is absent then ED measure only the noise energy, otherwise it measures the signal plus noise energy in presence of PU. In the analytical model, spectrum sensing is usually formulated as a binary hypothesis testing problem [33] which is as follows:

$$y(i) = \begin{cases} n(i) & H_0 \\ h(i)s(i) + n(i) & H_1 \end{cases} \tag{2.1}$$

where $y(i)$, is the complex signal observed by the sensing receiver (SU),and $s(i)$ is the transmitted PU signal, $h(i)$, stands for the PU channel impulse response and $n(i)$ is the zero-mean complex, circularly symmetric, wide-sense stationary additive white Gaussian noise ($AWGN$). Under hypothesis H_0 the primary user is considered absent, and the received signal sample $y(i)$ contain only noise. On the contrary, the received signal sample $y(i)$ under hypothesis H_1 consists of the transmitted signal after channel $h(i)$, together with the noise. Based on this binary hypothesis test model, the test statistic for ED can be formulated as:

$$T_y = \frac{1}{N} \sum_{i=0}^{N-1} |y(i)|^2 \tag{2.2}$$

where N denotes the length of the observation sequence. The test static can be modeled by the Gaussian distribution [25]. $\xi\left(.,.\right)$. According to this the following formulation is deduced:

$$T_y \mid_{H_0} \sim \xi\left(\sigma_n^2, \frac{\sigma_n^4}{N}\right) \tag{2.3}$$

And

$$T_y\,|_{H_1} \sim \xi\left(\sigma_X^2 + \sigma_n^2, \frac{\left(\sigma_X^2 + \sigma_n^2\right)^2}{N}\right) \tag{2.4}$$

where σ_X^2 denotes the variance of the transmitted signal with possible fading channel effects while transmission, σ_n^2 the variance of AWGN.σ_X^2 and σ_n^2 are assumed to be statistically independent. SNR is denoted as $\gamma = \sigma_X^2/\sigma_n^2$. The corresponding probability false alarm P_{f_a} and probability of detection P_D can be expressed as:

$$P_{f_a} = \Pr\left(T_y > \lambda\,|_{H_0}\right) = Q\left(\frac{\lambda - \sigma_n^2}{\sigma_n^2/\sqrt{N}}\right) \tag{2.5}$$

$$P_{f_D} = \Pr\left(T_y > \lambda\,|_{H1}\right) = Q\left(\frac{\lambda - \sigma_n^2\left(1+\gamma\right)}{\sigma_n^2\left(1+\gamma_s\right)/\sqrt{N}}\right) \tag{2.6}$$

Respectively, where Q(.) denotes the Gaussian Q-function and λ is the predefined threshold. The value of λ is determined by the assumed noise variance σ_n^2, targeted false alarm probability P_{f_a} and observation sequence N length as follows:

$$\lambda = \sigma_n^2\left(1 + \frac{Q^{-1}(P_{fa})}{\sqrt{N}}\right) \tag{2.7}$$

where $Q^{-1}(.)$ denotes the inverse Gaussian Q-function. The energy detection method's working principal can be explained with the **Figure. 2.2**

After the receiver front-end signal, channel filter and analog to digital converter (ADC), FTT is employed to split the signal into comparatively narrow frequency bands as seen in **Figure. 2.2**. In ED based spectrum sensing algorithms, the absolute square of FFT processed data block are calculated to obtain the test statistics. The output from the average data block is then compared to a predefined threshold, which is used to discover the existence of absence of the primary user. The threshold can be set to be fixed or variable

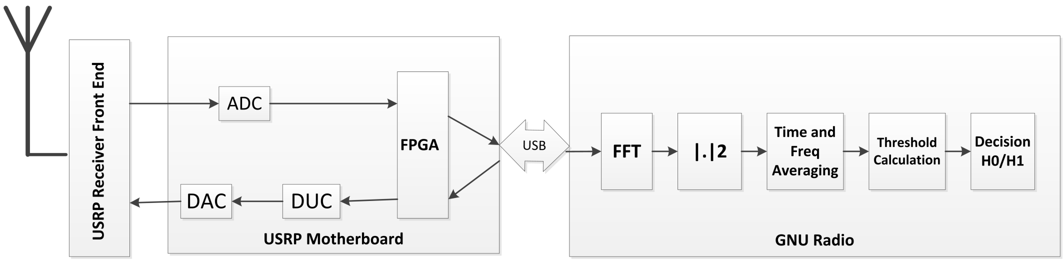


Figure 2.2: Block diagram of energy detector with GNU Radio based spectrum sensing with time and frequency domain averaging.

based on the channel condition. It is assumed that the sampling rate used in each sub-band is equal to the ADC sampling rate divided by the number of FFT bins. H_0 and H_1 illustrate the absence hypothesis and present hypothesis of a PU, respectively. When there is only noise present, it is modelled as a zero-mean Gaussian random variable with variance $\sigma_n^2/\mathrm{N}_{FFT}$, where N_{FFT} is the number of FFT bins or subchannels. The threshold value play an important role and is calculated based on the assumed noise variance and the target probability of false alarm. The detection probability P_d is the probability of the scenario that T_y is larger than the threshold λ, which means that the corresponding sub-band is considered to be occupied by the PU. The probability of false alarm is the probability that the receiver incorrectly detects the signal when the PU is not transmitting and T_y is greater than λ.

The main advantages of ED based SS algorithms, is its low computational and implementation complexity as seen in the above illustration. In contrast, the main limitation of ED is its high dependence of the knowledge of noise variance. Noise variance uncertainty can cause apparent decrease to the sensing performance [27].

2.4.3 Cyclostationary Detection

The Cyclostationary feature detection is more robust to noise uncertainties and performs better than energy detection in low SNR. It exploits the periodicity in the received primary signal to identify the presence of PU. The periodicity is commonly embedded in sinusoidal carriers, pulse trains spreading code, hoping sequences or cyclic prefixes of Pu signals. Although it requires a priori knowledge of the signal characteristics, Cyclostationary feature detection can distinguish the CR transmissions from various type of PU signals. This estimates the synchronization requirement of energy detection in cooperative sensing. Moreover, CR users mays not be required to keep silent during cooperative sensing and thus improving the overall CR throughput. The Cyclostationary method has its own shortcomings due to its high computational complexity and long sensing time. Due to these issues, this method is less common than energy detection in cooperative sensing [9].

The above techniques of spectrum sensing mentioned have advantages and disadvantages. The analysis of these spectrum sensing techniques are summarized and listed in Table 2.1.

Table 2.1: Pros and cons

	Spectrum Sensing Technique	Advantages	Disadvantages
	Matched filter detection	Low computation cost,accurate detection	Requires prior knowledge about PU
	Energy Detection	Does not require any prior knowledge about PU signal, efficient,less complex	Functionality limited in low SNR
	Cyclostationary	Perfect works in low SNR,Robust against interference	Requires prior knowledge about PU,less efficient in term of computation cost

Chapter 3

Testbed Implementation Environment

3.1 Introduction

The aim of this thesis is to study the real-time sensing performance of ED spectrum sensing algorithms in CR at the 2.4 GHz ISM band. The transmitter and receiver in this testbed are working with real time complex signals. In order to implement this testbed personal computer (PCs), for signal generator, NI-USRP 2900 hardware, GNU Radio and MATLAB software are used. NI-USRP is a flexible low-cost platform which is designed for the implementation of software defined radio (SDR) in a wide range, developed by Matt Ettus [12], It is a software programmable radio transceiver with the ability to transmit and receive radio frequency (RF) signals across a real-time bandwidth up to 70 MHz- 6 GHz. USRP paired with GNU radio platform can be used as a powerful PC-hosted wireless communication system. On GNU software platforms graphical programming environment and several build-in toolboxes for USRP, flowgraph block can be implemented easily and work efficiently. In this chapter the main purpose is to introduce the hardware and software we used for the implementation of SS system.

3.2 Software defined Radio

Software defined radio (SDR) is a radio communication system where components that have been typically implemented in hardware (e.g. filter, amplifiers, modulation/demodulation, detectors etc.) are implemented by means of software on embedded system. It is the concept which implements radio transmitter and receiver with programmable software (or firmware) to perform signal-processing tasks instead of implementing them completely using hardware. SDR has become more prevalent in wireless communication networks. The idea behind SDR is to turn radio hardware problems into software problem [10]. Most of the signal processing is done via software by using Field-Programmable Gate Arrays (FPGA), or General-Purpose Processors (GPP). The main characteristic of SDR is that software defines the transmitted waveforms and demodulates the received waveforms. At the transmitter side, the SDR software transforms the data into processed digital signals with the users preferences and provides the digital signal to digital to analogue converter (DAC). The DAC transform the digital signal into analog waveforms. Then the analog waveform is transmitted through the antenna to the radio channel, for precise testing purposes. At the receive side the antenna acquires the received analog waveform which is then processed by the analog front end and provided to the analog digital converter (ADC), which transforms the analog waveform into digital signal. Afterwards the digital signal is proceeds to the SDR software for the further signal processing and reconverting the data that have been transmitted. The universal architecture of a typical current SDR system with GNU radio software can been explained as shown in **Figure. 3.1**. In **Figure.**

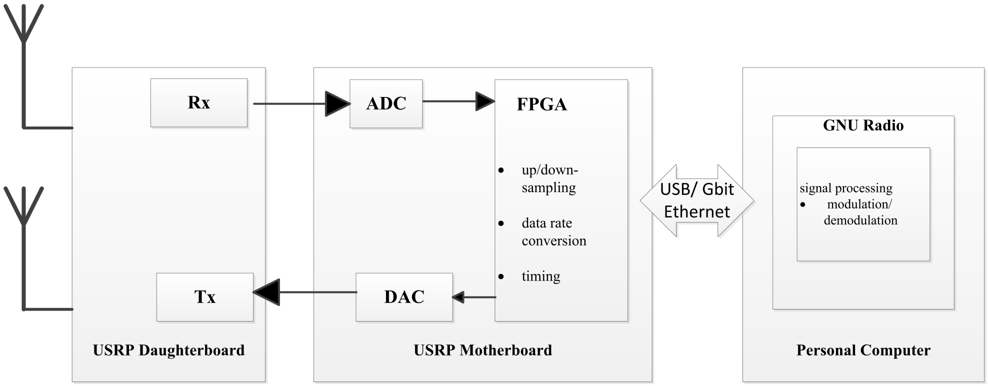


Figure 3.1: Typical Software defined radio structure.

3.1,the SDR structure is divided into three blocks. The left one builds the RF front end of the hardware which serves as interface to the analog RF domain. In the second block, the intelligence of the hardware part is implemented, forming the interface between the digital and the analog world. In the third block, the whole signal processing is done fully designed in software. An analog RF signal can be received or transmitted over antennas or can also be directly connected via SMA connectors to the SMA port of the RF front end called daughterboards.

3.3 Universal software radio peripheral

Universal software radio peripheral (USRP) is the most common hardware used with GNU Radio to build SDR system designed and sold by Ettus Research and its parent company, National Instrument. USRP is a flexible low-cost hardware device which facilitate of making SDR for a wide range developed by Matt Ettus [12],and commonly used by research labs and universities. The main task of USRP is to convert the digital base-band signal coming from the computer to analog signal in the RF band. Most USRPs connect to a host computer through a high-speed link, which the host-based software uses to control the USRP hardware and transmit/receive data. All USRPs product are controlled with the open source UHD driver. The USRPs product family includes a variety of models that use a similar architecture. Second generation USRPs are able to work with GNU radio, NI-LabVIEW and Simulink, [28–30]. A motherboard provides the following subsystems: clock generation and synchronization, FPGA, ADCs, DACs, host processor interface, and power regulation. These are the basic components that are required for baseband processing of signals. A modular front-end, called a daughterboard is used for analog operations such as up/down-conversion, filtering and others signals conditioning. Both motherboard and daughterboard can convey and/or receive in different frequency ranges from 70 MHz to 6 GHz.

A motherboard is composed of an onboard Field Programmable Gate Array (FPGA) and an onboard removable memory. Before using USRP device, a firmware and FPGA image must be downloaded into the memory. Then the USRP device is able to correspond with any computer installed USRP Hardware Driver (UHD), via Gigabit Ethernet or USB port in USRP. The FPGA does not perform any advance digital processing signal such as modulation, demodulation and coding. Such processing is always done in the computer. The primary objective of motherboard is to convert analogue IF signal to digital baseband signals and vice versa, the conversion of RF-IF is done on daughterboard side. The bandwidth of IF signal processed by the motherboard depends on the daughterboard specifications. The digitalized IF signal is then executed in the FPGA, then the digital decimating low pass filter preloaded in FPGA processes the signal. The data rate of this signal is generally high for general computer processing [13]. Therefore, the use of filtering is to reduce the sampling rate to suit the processing capabilities of the computer and the capability of USB cable.

The USRP devices should be able to capture and process weak signals in a wide dynamic range, which would demand for multi GHz speeds ADC. However, it is impossible to realize ADCs at that high ultra-speed because the USRP device has only 100 M samples. To solve this problem Ettus research had introduced the daughterboard that can be plugged into the motherboards via transmit/receive port. The frequency range is narrows down by the band pass filter on daughterboard, then the residual RF is converted down to IF such that the subsequent ADC can handle the speed of the signal. There are numerous daughterboards which each operating within a frequency range. Altogether, cover a wide frequency spectrum from 0 to 6 GHz. Table 3.1 shows the list of available daughterboards at Ettus research homepage [14]. The daughterboard XCVR2450 has two interface named TX/RX and RX2. It can provide up to two antennas, one on each interface. The TX/RX interface can be used either for transmission or reception at any given time, while RX2 interface is only used for reception. User can select the preferred interface for communication in the SDR script depending on the application.

3.3.1 NI USRP-2900

NI USRPs are SDR prototype platforms capable of numerous applications for education and research. Using NIs hardware paired with software offers flexibility and functionality for physical layer design, record and playback, signal intelligence, algorithm and more in affordable price. NI USRP is categorized into two series, namely NI USRP-2900/2901, both are connected to PCs using USB cable working in pair with GNU radio software and are able to perform multiple input, multiple output (MIMO) functionalities. In this thesis, the NI USRP-2900 is used for the implementation.

NI USRP-2900 is shown in **Figure. 3.2**, it contains two SMA signal ports RX1/TX1 which can perform both transmission and reception and RX2, a USB connection port, and an external reference clock input port, a pulse per second reference input port,

Table 3.1: Available Daughterboard

Daughter boards	Daughterboard Type	Frequency Band
$BasicRX$	Receiver	$0.1300MHz$
$BasicTX$	Transceiver	$0.1200MHz$
$LFRX$	Receiver	$DC - 30MHz$
$LFTX$	Transceiver	$DC - 30MHz$
$TVRX$	Receiver	$50 - 860MHz$
$DBSRX$	Receiver	$800 - 2400MHz$
WBX	Transceiver	$50 - 2200MHz$
$RFX400$	Transceiver	$400 - 500MHz$
$RFX900$	Transceiver	$750 - 1050MHz$
$RFX1200$	Transceiver	$1150 - 1450MHz$
$RFX1800$	Transceiver	$1500 - 2100MHz$
$RFX2400$	Transceiver	$2250 - 2900MHz$
$XCVR2450$	Dual- Band Transceiver	$2400 - 5000MHz$

a power adapter port in the front panel, and an external GPS antenna connection port at the back. NI USRP-2900 has a very wide frequency range from 70 MHz up to 6 GHz with an instantaneous Real-time bandwidth of 56 MHz (with 8-bit samples). When the USRP works as transmitter, the DAC rate is 61.44 MS/s for 8-bit samples with 2 channels, and the DAC spurious free dynamic range (SFDR) is 89.75 dB. When it works as receiver, the noise figure is around 5 to 7 dB and the ADC rate is 61.44 MS/s with SFDR of 76 dB. The maximum input power at RX is -15 dBm.

The NI USRP structure is shown in **Figure. 3.3**, which the RF switches are used

Figure 3.2: NI USRP-2900 Front end.

to select which port of TX/RX to use. The RX2 can be used only for receive and TX/RX for both. When it works as a receiver, the analog waveform is acquired from the selected RX port and fed to low noise amplifier (LNA) and drive amplifier, the signal amplified is then mixed with local oscillator (LO) in order to move the signal baseband or IF from carrier frequency. The real part and imaginary part of the signal are filtered by low pass filters (LPF) with bandwidth and then converted into digital signal by ADC and fed to programmable digital down converter (DDC) to perform down-conversion, if needed and finally the digital signal is transmitted via USB connection port for further processing. When it works as a transmitter, the processed digital signal is first transmitted from the PC to USRP hardware via the USB port ant then fed to digital up converter (DUC) for up-conversion, afterwards the digital signal is converted into analog signal by DAC. Afterwards the analog signals are filtered by the low pass filter and mixed by local oscillator in order to transfer the signal from baseband to a specific carrier frequency and amplified according to users preferences by the TX amplifier. Finally, the signal is transmitted through TX1 to antennas (in the setting of USRP the TX/RX should be correctly selected otherwise errors will occur).

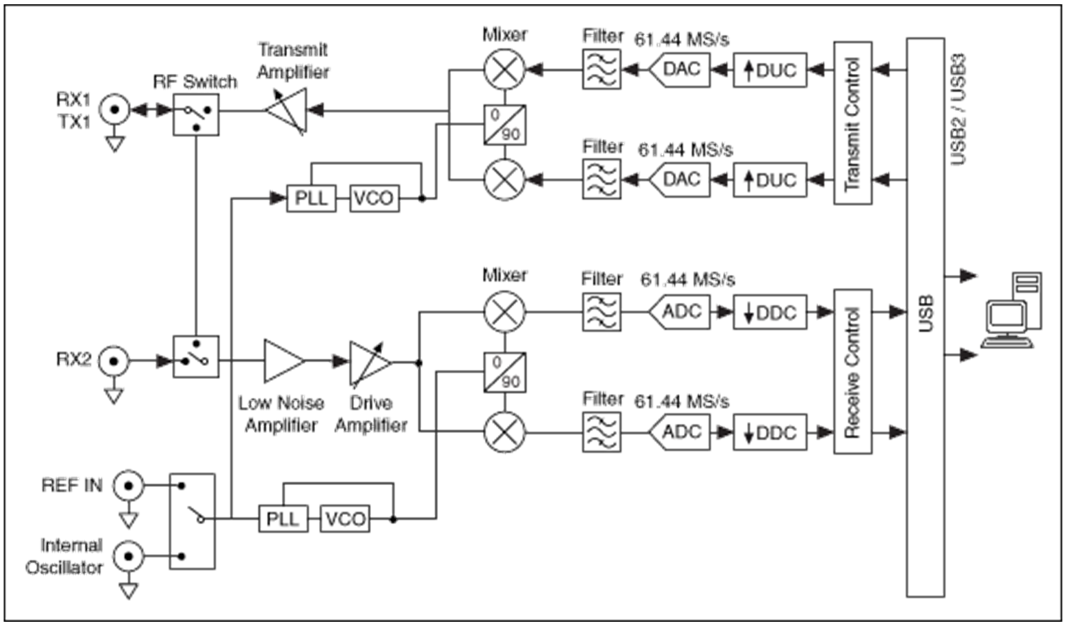


Figure 3.3: NI USRP-2900 Block diagram [31].

3.3.2 NI USRP-2900 limitations

While USRP is certainly a very flexible powerful and efficient model of SDR, limitations are inherent within any radio design. The maximum throughput from the USRP to the host computer where signal processing except up/down-converting and decimation are done, is well known limitation of the USRP. Some hove reported the USB 2.0 interface support data rates around 32 MS/s and an approximate bandwidth of about 6 MHz of I/Q data and 12 MHz of real data. Because of this limitation, standards 802.11b/g, which have 20 MHz bandwidth (sampling rate), are not feasible using USRP. Another limitation of USRP is that model of USRP working in pair with GNU radio software is very sensitive to LO leakage and DC offset, which introduced a very sharp peak at 0 MHz bandwidth as seen in **Figure. 3.4**.

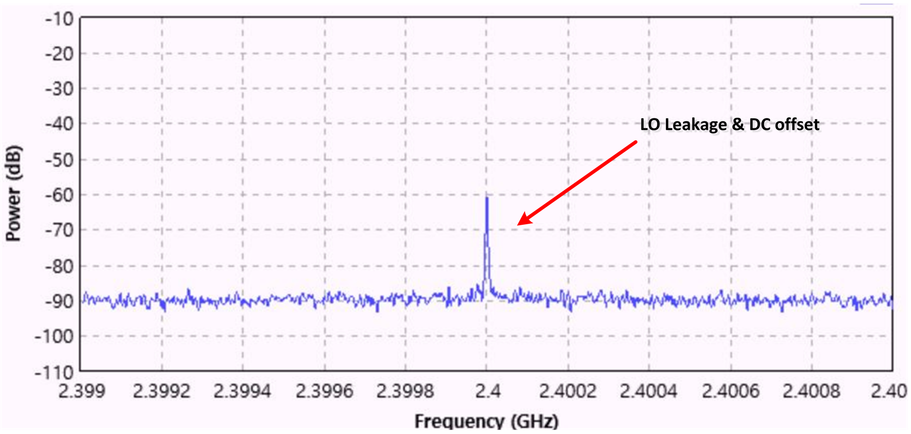


Figure 3.4: Frequency spectrum observed by receiver when no signal is transmitted by PU.

3.4 Gnu Radio

Gnu radio is a graphical programming platform which is suitable for a variety control and measurement systems. It offers a great integration with existing legacy software and hardware, which makes problem-solving and innovation faster and efficient. The GNU radio project was founded by Eric Blossom with the intention of creating an open source framework for software radios. GNU Radio is a free open-source software development toolkit platform for signal processing focusing on implementation of SDR [11]. It can be used with low cost external RF hardware to create SDR or without hardware in a simulation like environment. It is widely used by researchers in wireless communication and for a real-time implementation of SDR. The function and methods used in GNU Radio are represented by blocks library, and its applications are written and developed in C++ programming language while Python is used as a scripting language to tie the blocks together to form a flowgraph. Simplified Wrapper and Interface Grabber (SWIG) is used as the interface compiler which allows the integration between C++ and python language as shown in **Figure. 3.5**. Providing the application programming interface (API), GNU Radio represents the Centre of SDR development with USRP front end and offers various pre-building blocks for signal processing as well as a method to manipulate the data flow between the blocks.

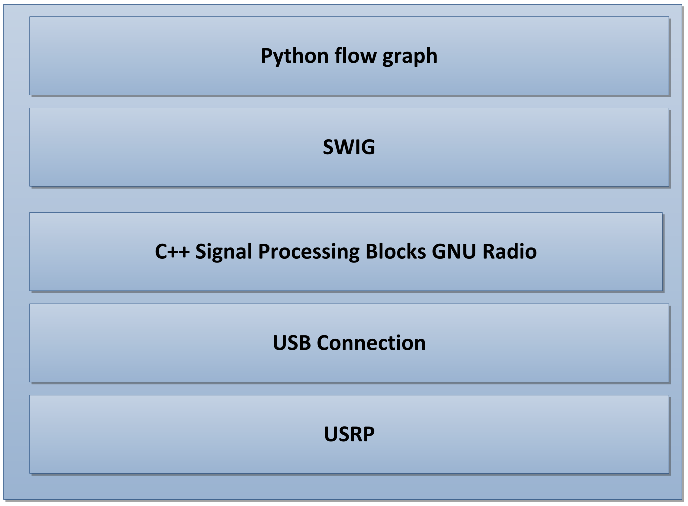


Figure 3.5: GNU Radio software architecture.

The architecture of GNU radio consists of two components. The first one is a set of various C++ blocks which implement digital processing operations such as filtering I/O operation, FFT/IFT modulation etc. The second one is the framework implemented as python script to control data among block. Blocks are building by using GNU Radio Companion (GRC), a graphical Application Programing Interface (API) that allows users to build GNU flowgraph and control the USRP device. A partial list of available functions offered by the GNU Radio platform is shown in **Figure. 3.6**, and an example of a data flow is shown in **Figure. 3.7**. GNU radio runs under several operating systems like Linux,

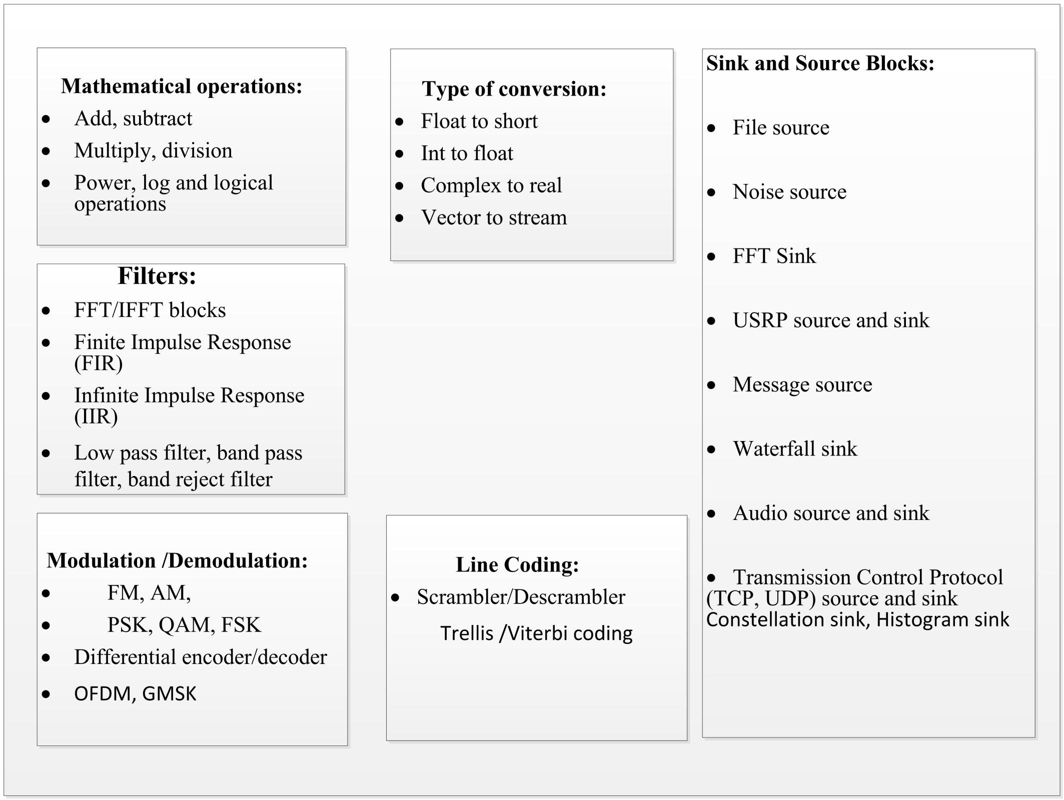


Figure 3.6: GNU Radio Functions.

Mac OS, and Cygwin porting for Windows exists. Each block defines a $general_work()$ function that performs on its input to procedure output data streams. The block provides also $forcast()$ function that returns the system runtime the number of input data streams it requires to perform, and the number of output data streams it produces from the given input items. On different input stream of block the data rate can be different, but the all output streams must have same data rate. Both input and output data streams of block have associated buffers, and each has a read/write buffer. Block read data from the read buffers for signal processing after that write data streams to the appropriate write buffer of its output data streams. The data buffers are utilized to implement the edges in the flowgraph; the write buffers for one block are the read buffers of upstream block in

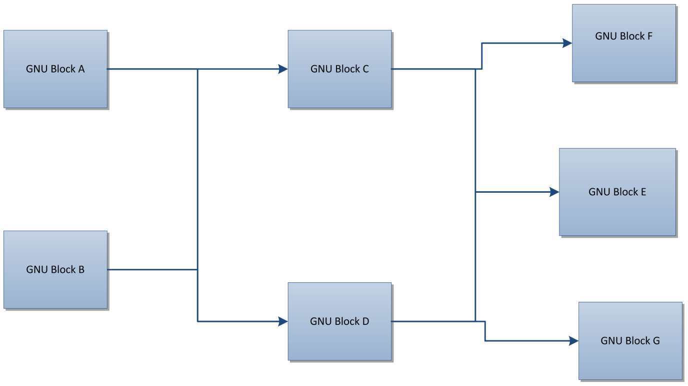


Figure 3.7: Example of GNU flowgraph.

the flowgraph. GNU radio buffers are single writer and multiple reader First in First Out (FIFO) buffers. It means that one output data stream can connect to one or more input data stream and one input on receive data from only one output. In python, blocks are connected by the virtual connect function which indicates how the output data streams of a block connect to the input data stream of one or more upstream blocks. Then the flowgraph mechanism automatically constructs the flowgraph and this is hidden from the user. The main function of the flowgraph mechanism is the allocation of data stream buffers to connect the blocks. This mechanism considers the sizes of input and output data streams used by the block and the associated data rate at which block consume and generate input and output streams. After buffers are allocated, they are linked to the input and output stream to appropriate blocks. Hierarchical block can be produced by combining several blocks. During the Gnu radio scheduler execution, the scheduler checks the input requirements of each block and the mentioned forecast () function is used to decide the received data rate of the block from its available inputs. If data is appropriate in the input buffers, block general work () function is executed. If it does not have appropriate data in input buffer, the scheduler will skip the block and move on the next block in the flowgraph. Skipped blocks will not be performed until more input data is sufficient.

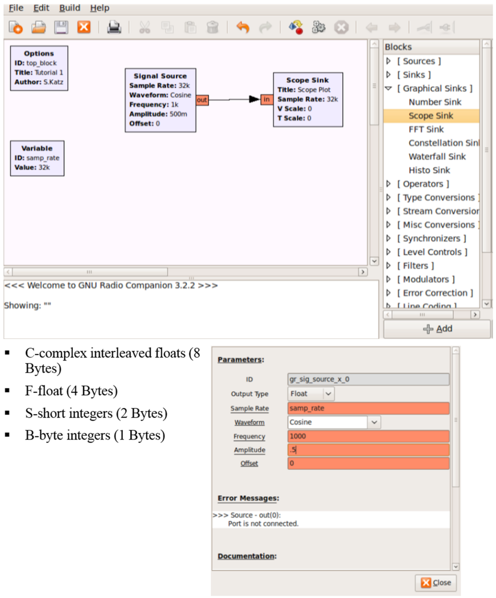
3.4.1 Benefits of programming in GNU Radio

One advantage that make the GNU radio attractive as an ideal programming language compared to other programing languages is its graphical, dataflow programming style. GNU radio allows a rapid implementation even a complex receiver and transceiver system. In GNU radio tool, Programming is performed by writing together signal processing flow chart on a block diagram instead of programming in text, and those blocks are executed according to the rules of data flow. The benefit of graphical programming on GNU radio is that the users can easily drag and dropped block and spend more time on actual system design and focus on problem-solving, rather than spend significant time learning the specific text-based syntax of programming languages. There is a great amount of pre-assembled signal processing blocks for waveform creation and analysis for different purpose provided in GNU radio tool.

GNU radio can be integrated with Scilab and Xcos for complex communication, it can start Scilab and send command to it for processing. It can also retrieve values from Scilab variables present in the Scilab memory for that instance. Features which are not available can be easily included by adding custom blocks. GNU radio has also the ability to plot the output in real-time.

3.4.2 NI USRP-2900 working with GNU Radio- Companion

According to the advantages mentioned above, GNU radio is used to control the NI USRP in our implementation. The graphical user interface for GNU Radio is known as GNU Radio Companion (GRC). GRC is an open source tool-based Python/C++ to build software radio systems. GRC allows connecting components and generating a signal flow diagram by using a graphical drag-and-drop interface. Component blocks are normally implemented in C++ and connected using Python programming language. With GRC, a simple data flow on GRC is shown in **Figure. 3.8**.



Options
ID: top_block
Title: Tutorial 1
Author: S.Katz

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 500m
Offset: 0

Scope Sink
Title: Scope Plot
Sample Rate: 32k
V Scale: 0
T Scale: 0

Variable
ID: samp_rate
Value: 32k

```
<<< Welcome to GNU Radio Companion 3.2.2 >>>

Showing: ""
```

- C-complex interleaved floats (8 Bytes)
- F-float (4 Bytes)
- S-short integers (2 Bytes)
- B-byte integers (1 Bytes)

Parameters:

ID: gr_sig_source_x_0
Output Type: Float
Sample Rate: samp_rate
Waveform: Cosine
Frequency: 1000
Amplitude: .5
Offset: 0

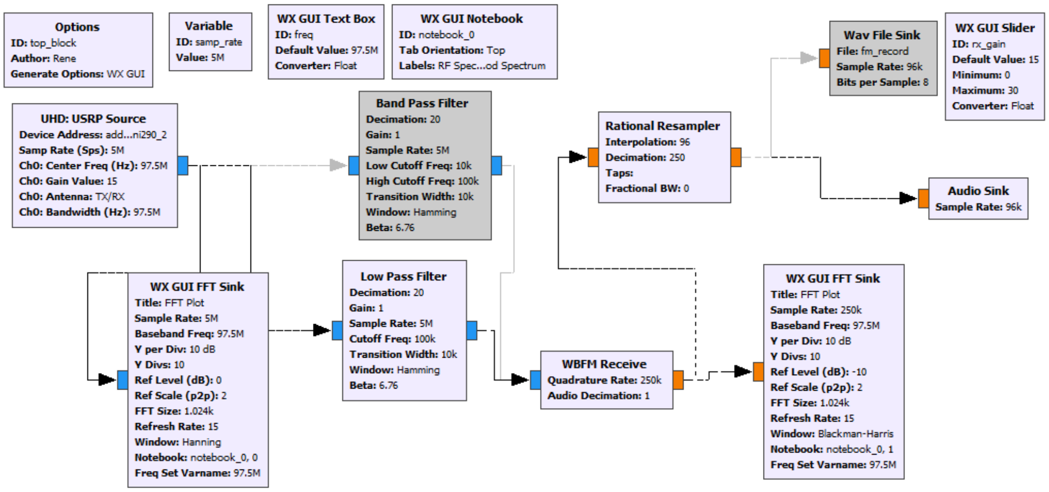
Error Messages:

```
>>> Source - out(0):
    Port is not connected.
```

Documentation:

Figure 3.8: Example of flowgraph in GRC.

Each block includes a set of input and/or output port and it receives data from input port and produce data for its output port. The blocks called source and sink have only input port or output port. For example, the block signal source and scope sink are blocks that read and write respectively from/to USRP receiver/transmitter port, socket and files. Blocks can have, as well as the data type on the corresponding port. With GRC, a simple FM broadcast receiver is easy to build in the SDR system using USRP as shown in **Figure. 3.9**.



Options
ID: top_block
Author: Rene
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 5M

WX GUI Text Box
ID: freq
Default Value: 97.5M
Converter: Float

WX GUI Notebook
ID: notebook_0
Tab Orientation: Top
Labels: RF Spec...od Spectrum

Wav File Sink
File: fm_record
Sample Rate: 96k
Bits per Sample: 8

WX GUI Slider
ID: rx_gain
Default Value: 15
Minimum: 0
Maximum: 30
Converter: Float

UHD: USRP Source
Device Address: add...ni290_2
Samp Rate (Sps): 5M
Ch0: Center Freq (Hz): 97.5M
Ch0: Gain Value: 15
Ch0: Antenna: TX/RX
Ch0: Bandwidth (Hz): 97.5M

Band Pass Filter
Decimation: 20
Gain: 1
Sample Rate: 5M
Low Cutoff Freq: 10k
High Cutoff Freq: 100k
Transition Width: 10k
Window: Hamming
Beta: 6.76

Rational Resampler
Interpolation: 96
Decimation: 250
Taps:
Fractional BW: 0

Audio Sink
Sample Rate: 96k

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 5M
Baseband Freq: 97.5M
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Window: Hanning
Notebook: notebook_0, 0
Freq Set Varname: 97.5M

Low Pass Filter
Decimation: 20
Gain: 1
Sample Rate: 5M
Cutoff Freq: 100k
Transition Width: 10k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 250k
Audio Decimation: 1

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 250k
Baseband Freq: 97.5M
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): -10
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Window: Blackman-Harris
Notebook: notebook_0, 1
Freq Set Varname: 97.5M

Figure 3.9: FM receiver build by GRC.

Chapter 4

Details of testbed Implementation

4.1 Introduction

This chapter covers the details of our implementation for testing the Spectrum Sensing algorithms use for this thesis. The system contains two parts as one transmitter and one receiver. The transmitter is implemented using one $USRP-2900$ with ID $ni2900_2$ controlled by the PC computer using GNU radio. The receiver is implemented using a second NI USRP-2900 with ID $ni2900_3$ and laptop using GNU radio toolkits for spectrum sensing purposes and acted as secondary user. The schematic diagram and the dedicated photographs of the experimentation setup are shown in **Figure. 4.1**, and in **Figure. 4.2**

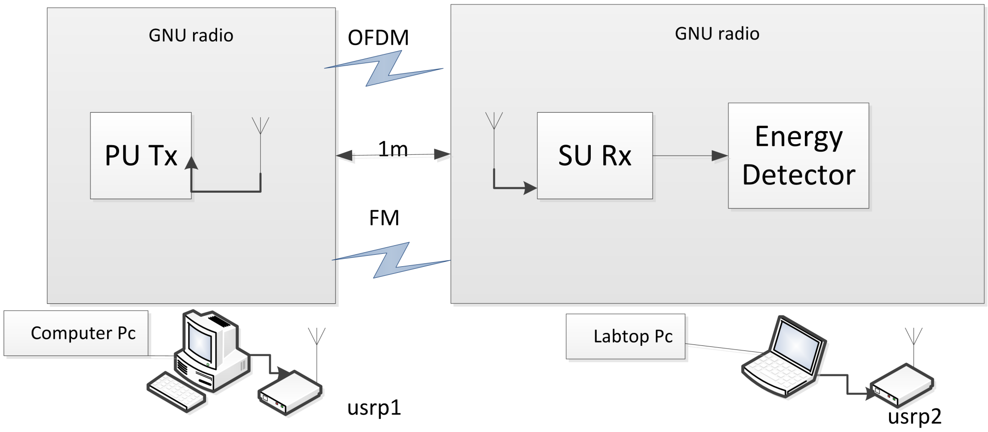


Figure 4.1: Schematic system diagram.

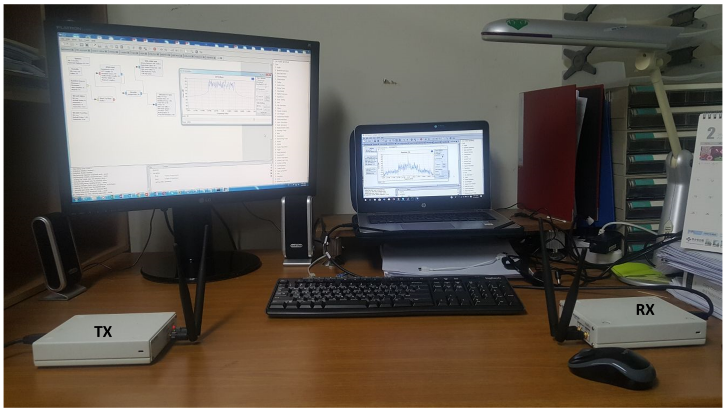
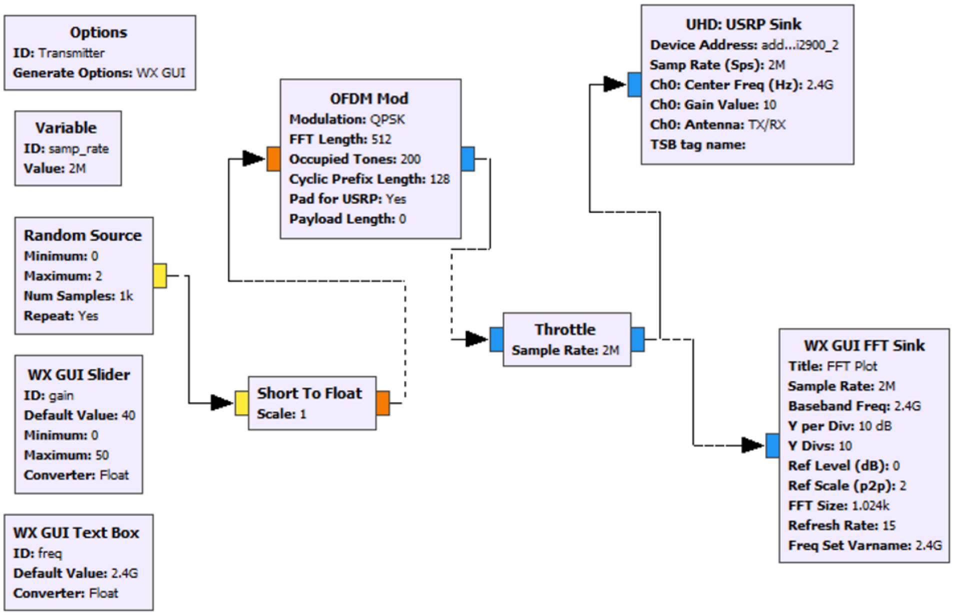


Figure 4.2: PU transmitter and SU receiver realized by means of USRP-2900 and personal computer.

4.2 Transmitter

4.2.1 OFDM Modulation

At the transmitter side the wideband multicarrier signal based on orthogonal frequency division multiplexing (OFDM) was generated. In the former case the composite radio signal has been created and frequency modulated before sending to USRP board via USB port. The spectrum of the multicarrier signal is assumed to be wider -the OFDM symbol with $N_OFDM = 512$ subcarriers has been used. As mentioned above the baseband processing has been realized on the PC computer using GNU Radio Companion, where the whole system is built from blocks. The screen-shot from the GRC illustrating the OFDM transmitter side is shown in **Figure. 4.3**. One can observe the presence of the signal source block ($Randomsource$) that generates repeatedly random data, which are mapped to QPSK symbols and then are subject to OFDM modulation block realized in $OFDMModblock$. Only 200 subcarriers from 512 available has been occupied. Finally, after the power is proper adjust, the signal is sent to the local spectrum analyzer ($FFTplot$) and to the USRP block ($USRPSink$), responsible for sending data. It can be noticed that the complex sampling frequency has been set to 2MHz and the center frequency was set to 2.4 GHz. There is interference at neighboring access points, which can interfere with the USRP frequencies. Therefore, by setting the frequency center at 2.4 GHz it can the interference and jamming with these access points can be avoided. The power signal spectrum is shown in **Figure. 4.4**.



Options
ID: Transmitter
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

Random Source
Minimum: 0
Maximum: 2
Num Samples: 1k
Repeat: Yes

WX GUI Slider
ID: gain
Default Value: 40
Minimum: 0
Maximum: 50
Converter: Float

WX GUI Text Box
ID: freq
Default Value: 2.4G
Converter: Float

Short To Float
Scale: 1

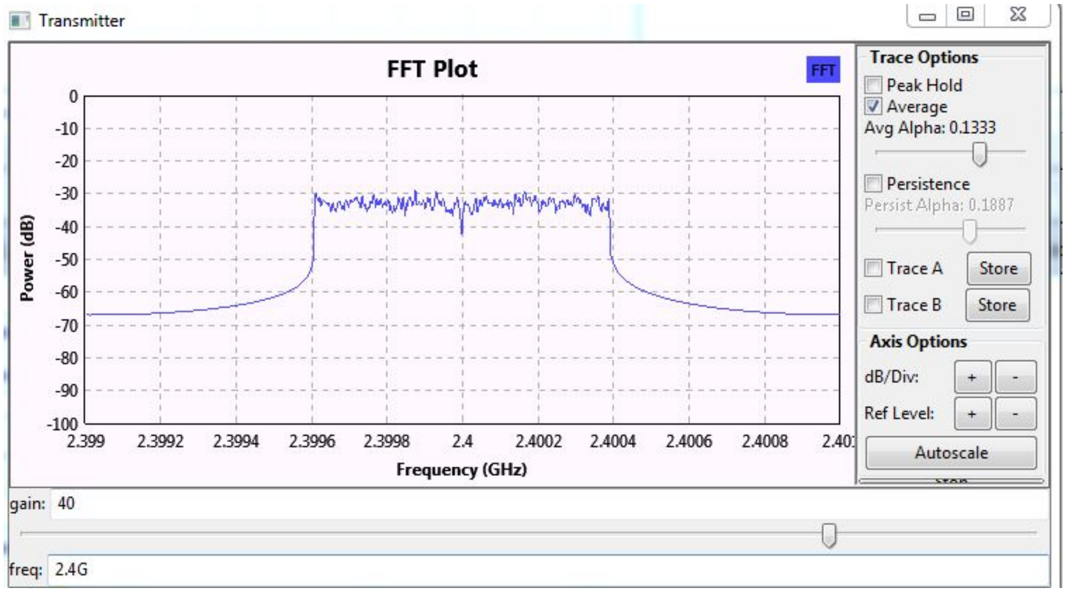
OFDM Mod
Modulation: QPSK
FFT Length: 512
Occupied Tones: 200
Cyclic Prefix Length: 128
Pad for USRP: Yes
Payload Length: 0

Throttle
Sample Rate: 2M

UHD: USRP Sink
Device Address: add...i2900_2
Samp Rate (Sps): 2M
Ch0: Center Freq (Hz): 2.4G
Ch0: Gain Value: 10
Ch0: Antenna: TX/RX
TSB tag name:

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 2.4G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: 2.4G

Figure 4.3: Block diagram of the PU transmitter with OFDM modulation in GRC.



☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1887
☐ Trace A
☐ Trace B

gain: 40
freq: 2.4G

Figure 4.4: PU transmitted OFDM power signal spectrum at 2.4 GHz.

4.2.2 Channel transmission

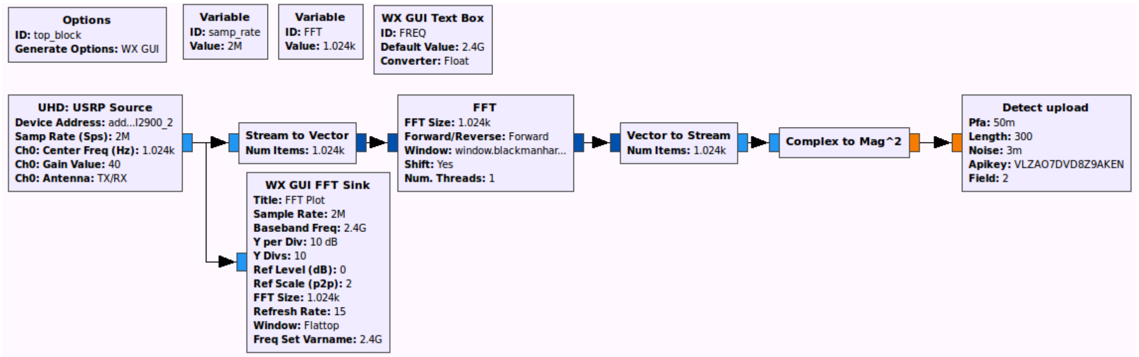
The channel models refer to effects as delay, fading and path-loss caused by the transmission medium between transmitter and receiver. The characteristics of transmitted signal changes whiles propagating through the medium, depending on the distance and the path of transmission paths environment [32].The mathematical model of the medium effect is called the transmission channel model. Channel model are combined to the original generated signal to make the received signal model similar to the real-world situation.

4.3 Receiver

4.3.1 SU Receive Block Diagram with ED

Analogously to the transmitter side the whole base-band processing that will be performed by the SU receiver has been realized in the laptop computer with $NIUSRP-2900$ using GRC. The schematic diagram of the receiver is shown in **Figure. 4.5**. One can observe the USRP Source block responsible for delivering data from the RF spectrum to the computer. It also operates at the center frequency equal to 2.4 GHz. In order to evaluate the influence of noise on the performance of selected spectrum sensing algorithms, additional block of noise generation has been used and the noise-signal of appropriate power has been added to the signal produced by the $USRPSource$ block. Afterward, the signal goes to through the chain with ED. One can observe the receive signal through the $FFTsink$ block used for displaying receive signal on the computer screen. The signal disturbed by additive white Gaussian noise is then transformed to frequency domain by means of $FFTblock$. The decisions are then transferred to the graphical sink. All the collected data are stored into the files. Then these data can be analyzed and stored in the detect upload block which is used for processing and storing data into data file and then we connect MATLAB on

GNU for estimated the performance of ED algorithm and connected to Thingsnetwork for uploading the detection probability. To upload the detection performance , the block used the API key ”VLZA07DVD8Z9AKEN” for connecting into Thingsnetwork.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

Variable
ID: FFT
Value: 1.024k

WX GUI Text Box
ID: FREQ
Default Value: 2.4G
Converter: Float

UHD: USRP Source
Device Address: add...I2900_2
Samp Rate (Sps): 2M
Ch0: Center Freq (Hz): 1.024k
Ch0: Gain Value: 40
Ch0: Antenna: TX/RX

Stream to Vector
Num Items: 1.024k

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 2.4G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Window: Flattop
Freq Set Varname: 2.4G

FFT
FFT Size: 1.024k
Forward/Reverse: Forward
Window: window.blackmanhar...
Shift: Yes
Num. Threads: 1

Vector to Stream
Num Items: 1.024k

Complex to Mag^2

Detect upload
Pfa: 50m
Length: 300
Noise: 3m
Apikey: VLZAO7DVD8Z9AKEN
Field: 2

Figure 4.5: schematic of SU receiver realized in GRC with energy detection.

4.3.2 Receiver signal power spectrum with FFT Plot

The out of the OFDM signal received by SU with ED algorithm is shown in **Figure. 4.6** and in **Figure. 4.7** respectively by observing two scenarios. In the first scenario, we tested the detection algorithm while the transmitter (PU) is turn off, and the scanned results are plotted in **Figure. 4.6**. We observed that the receiver cannot see any active signal when the transmitter is turn off. The displayed readings are just noise signal which do not have bandwidth greater than the threshold. In the second scenario when the transmitter is turn on, we observed that the signal is detected as active. It is clear that, the USRP was able to detect the signal when PU is on.

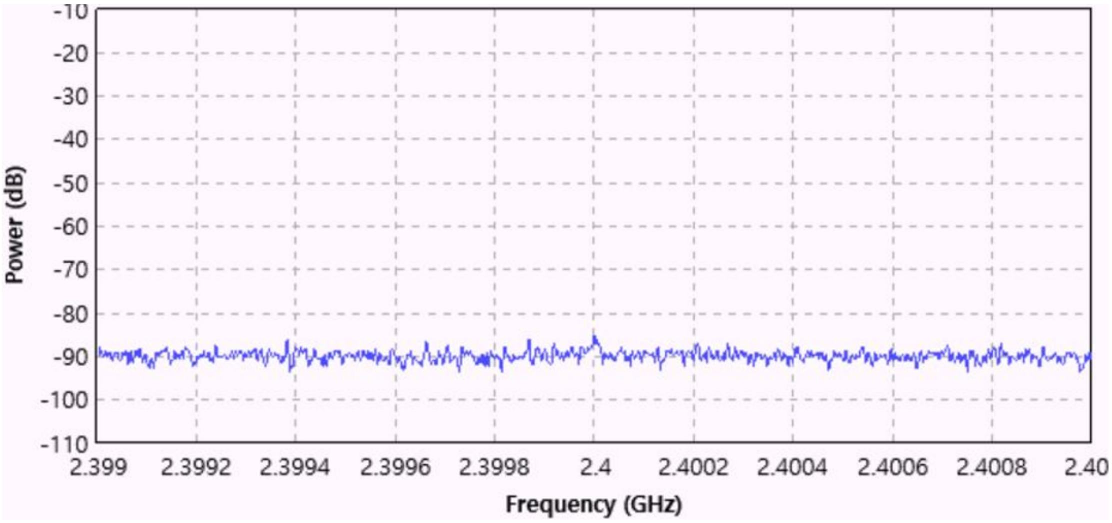


Figure 4.6: SU receiver OFDM power signal spectrum at 2.4 GHz with ED when TX is off.

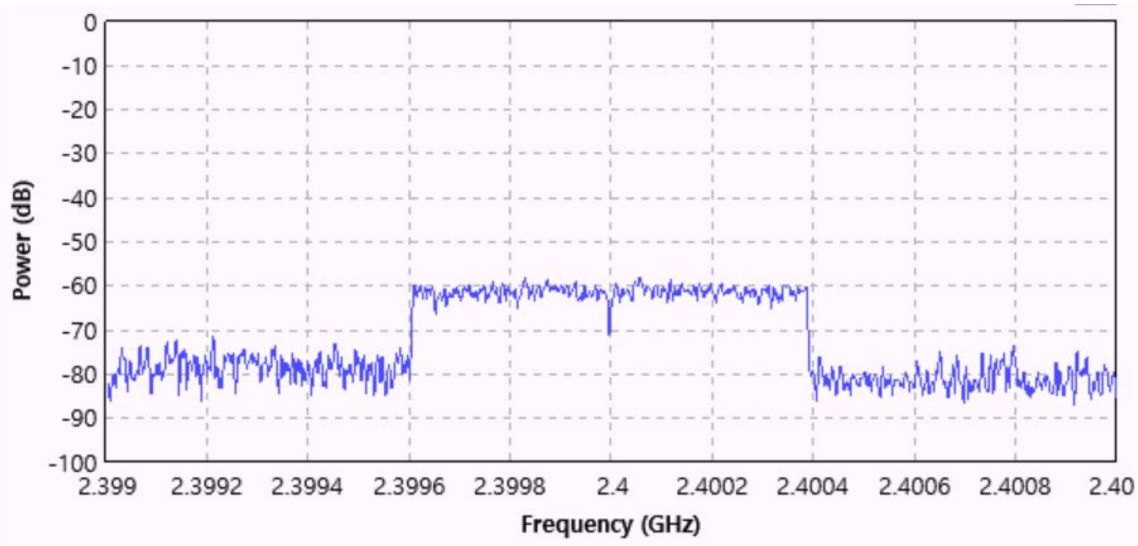


Figure 4.7: SU receiver OFDM power signal spectrum at 2.4 GHz with ED when PU is on.

4.4 Real-time video streaming using USRP

The development of wireless communication system includes the transfer of big data, video, images etc. This impose an additional capacity to the system and require more comprehensive structure to deal with these types of signal processing. In this section we demonstrate the basic concepts of real-time video streaming using USRP and VLC media player through GNU platform. We demonstrate how the video can be implemented on GNU radio by using USRP devices to transmit/receive the video in real-time streaming [36]. This experiment is divided into two parts. The first part is about the transmitter side and the second part is the receiver side implementation on GNU radio software. The input of the system is a real-time video signal which is processed by GNU radio. The video stream for transmitter is produced by VLC player. The overall system block diagram is shown in **Figure. 4.8**.

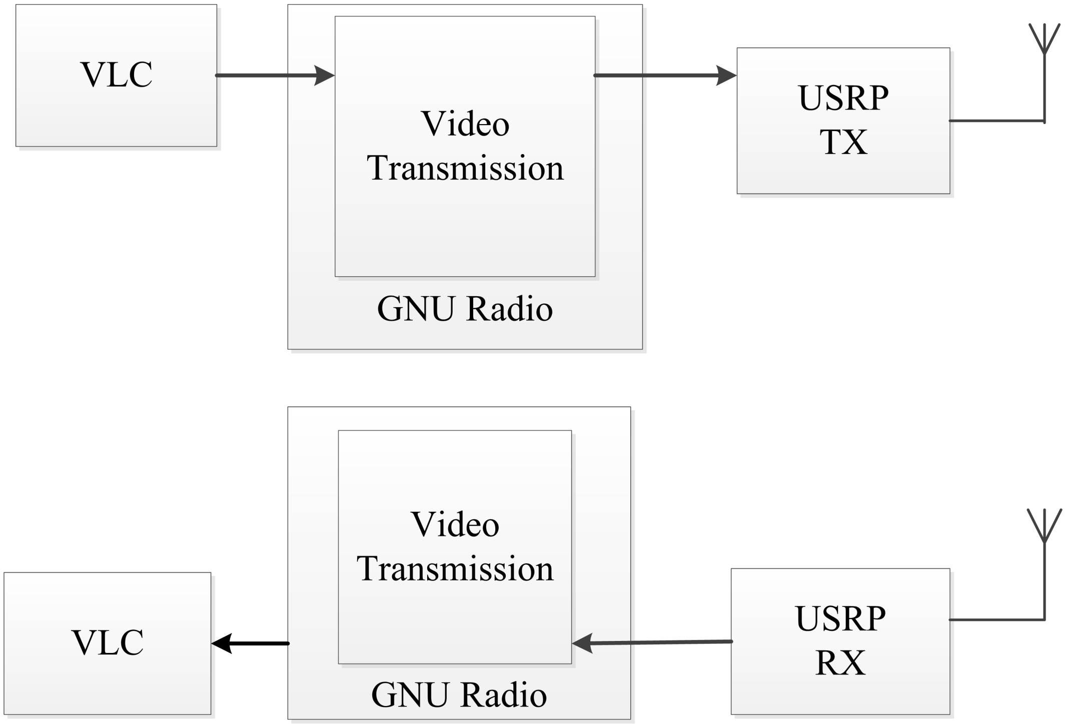
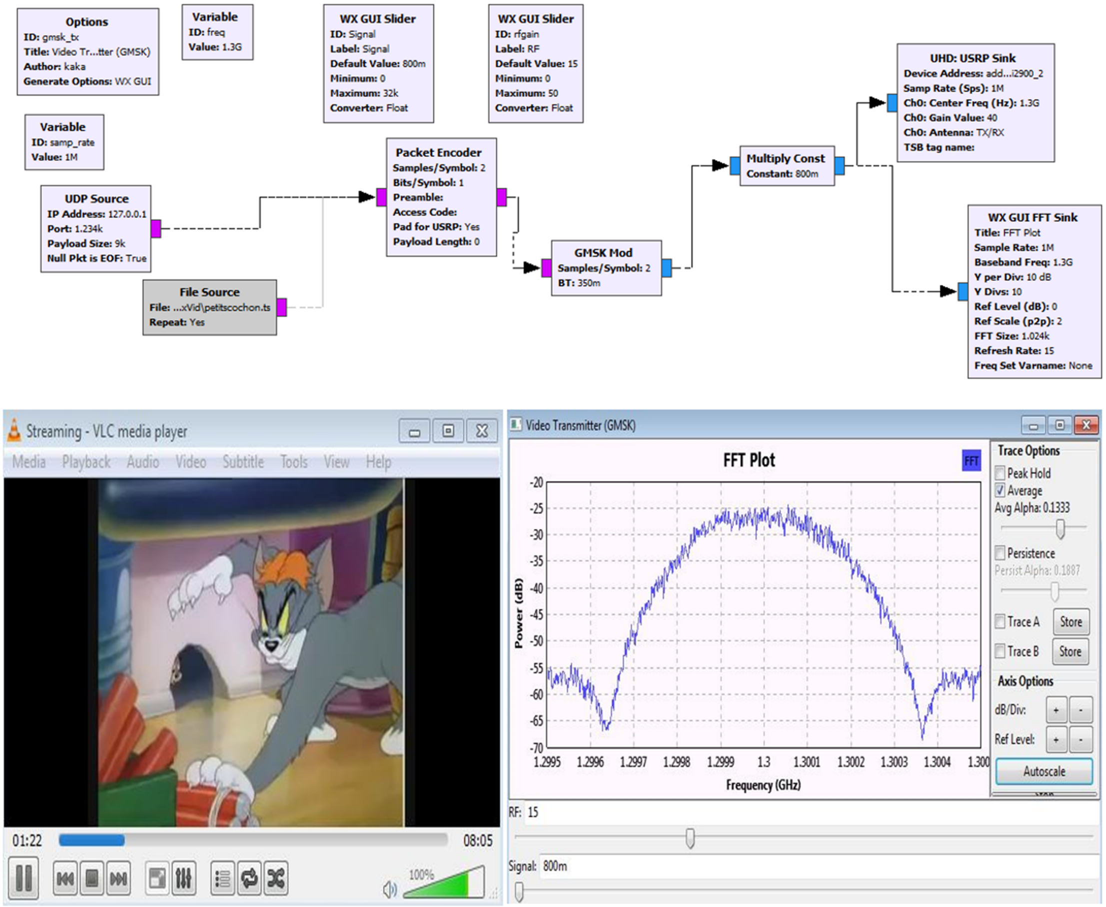


Figure 4.8: System diagram for video transmission with GNU Radio.

4.4.1 Video transmission side



Options
ID: gmsk_tx
Title: Video Tr...tter (GMSK)
Author: kaka
Generate Options: WX GUI

Variable
ID: freq
Value: 1.3G

WX GUI Slider
ID: Signal
Label: Signal
Default Value: 800m
Minimum: 0
Maximum: 32k
Converter: Float

WX GUI Slider
ID: rfgain
Label: RF
Default Value: 15
Minimum: 0
Maximum: 50
Converter: Float

Variable
ID: samp_rate
Value: 1M

UDP Source
IP Address: 127.0.0.1
Port: 1.234k
Payload Size: 9k
Null Pkt is EOF: True

File Source
File: ...xVid\petitscochon.ts
Repeat: Yes

Packet Encoder
Samples/Symbol: 2
Bits/Symbol: 1
Preamble:
Access Code:
Pad for USRP: Yes
Payload Length: 0

GMSK Mod
Samples/Symbol: 2
BT: 350m

Multiply Const
Constant: 800m

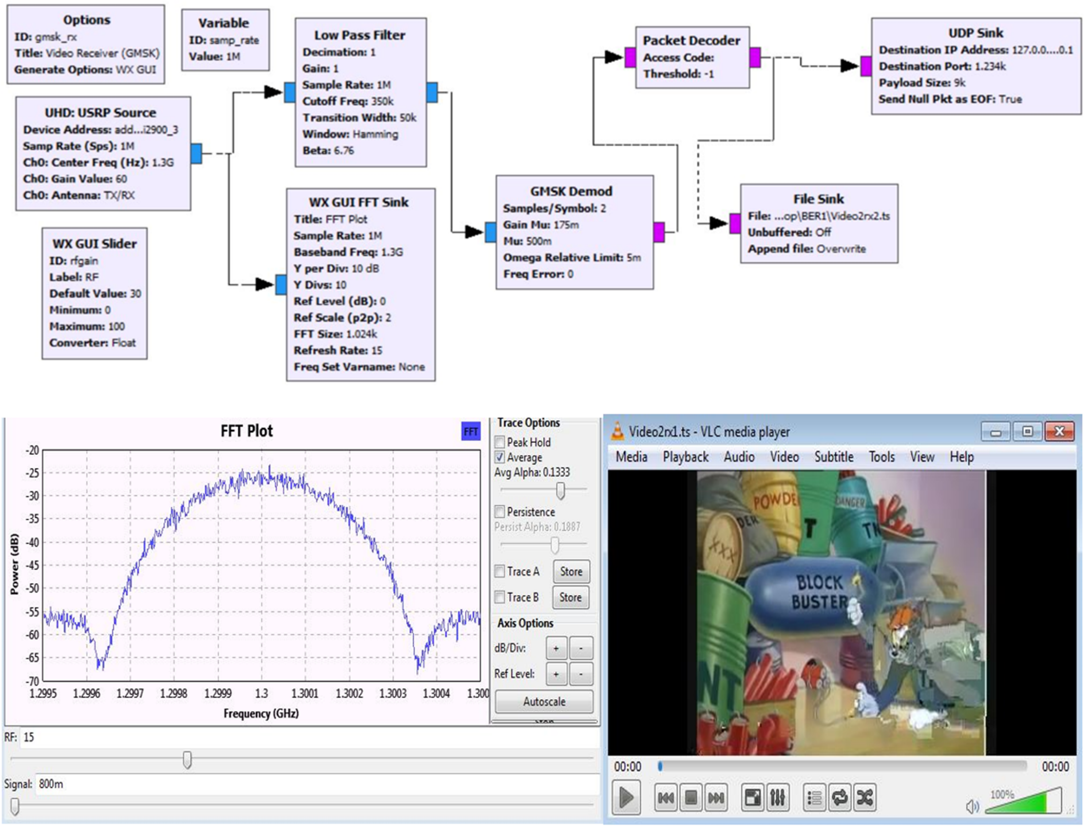
UHD: USRP Sink
Device Address: add...i2900_2
Samp Rate (Sps): 1M
Ch0: Center Freq (Hz): 1.3G
Ch0: Gain Value: 40
Ch0: Antenna: TX/RX
TSB tag name:

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1M
Baseband Freq: 1.3G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Figure 4.9: Flowgraph for video transmission.

The block diagram for video transmission is shown in **Figure. 4.9**. The first block in the flowgraph is a UDP source that receives the video input stream which is encoded in H.264 format by VLC player. Afterward the VLC sends the video stream to UDP port 1234. After the encoder video is received it is passed to encoder to be encoded in packets. We use 1bit/symbol and number of sample per symbol is 2. After forming packets, the packetized data is modulated using GMSK modulation following by multiplying the signal in order to boost the amplitude of modulated signal to be easily detected. Finally, the samples are sent out to the USRP through USB port. After performing DAC, the USRP sink sends to the daughter board for transmission in the RF environment.

4.4.2 Video reception



Options
ID: gmsk_rx
Title: Video Receiver (GMSK)
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 1M

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 1M
Cutoff Freq: 350k
Transition Width: 50k
Window: Hamming
Beta: 6.76

Packet Decoder
Access Code:
Threshold: -1

UDP Sink
Destination IP Address: 127.0.0....0.1
Destination Port: 1.234k
Payload Size: 9k
Send Null Pkt as EOF: True

UHD: USRP Source
Device Address: add...i2900_3
Samp Rate (Sps): 1M
Ch0: Center Freq (Hz): 1.3G
Ch0: Gain Value: 60
Ch0: Antenna: TX/RX

GMSK Demod
Samples/Symbol: 2
Gain Mu: 175m
Mu: 500m
Omega Relative Limit: 5m
Freq Error: 0

File Sink
File: ...op\BER1\Video2rx2.ts
Unbuffered: Off
Append file: Overwrite

WX GUI Slider
ID: rfgain
Label: RF
Default Value: 30
Minimum: 0
Maximum: 100
Converter: Float

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1M
Baseband Freq: 1.3G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Avg Alpha: 0.1333
Persist Alpha: 0.1887
RF: 15
Signal: 800m

Figure 4.10: Flowgraph for video Reception.

The flowgraph for the video reception is shown in **Figure. 4.10**. The first block is the USRP source block which captures the transmitter data from the environment and converted for processing. The captured data is demodulated using GMSK demodulator and forwarded to the packet decoder which performs the inverse operation of packet encoder to generate a bit stream and this stream is forwarded to UDP sink port so that it can be sent to the VLC player displaying the video. The quality of stream video received, performs well at the receive side with some delay and there was some packet loss due to the inability of spontaneous synchronization between transmitter and receiver frequency. The performance degradation was not high enough to cause visible distortion in the received data.

4.5 Simulation Results and Discussion

In this section, the implementation results of OFDM signal with the ED sensing algorithm is studied. The results will be presented according to their channel models. In this implementation a QPSK signal is tested with the carrier frequency of 2.4 GHz, the signal bandwidth is around 5 MHz and the sensing bandwidth is 8 MHz. the number of FFT points is 512. in this measurement, we observe the sensing performance of the sub-channels at the FFT sink block that contains the QPSK signal we transmit. In the process of detection the signal is store in a file which can be used for analysis in MATLAB. The detection probability performance according to the number of sensing sample under different SNR conditions for the indoor channel is shown in **Figure. 4.11**. In this figure it show that the detection performance must be better if more sample are used. **Figure. 4.12** illustrates the simulation and implementation result for the probability of false alarm and probability of detection according to different SNR condition. We can observe that when the SNR is high, the performance of detection and false alarm is improved. Afterward we integrated data uploading of the probability of detection versus the number of sample to the Thingsnetwork cloud in **Figure. 4.13**

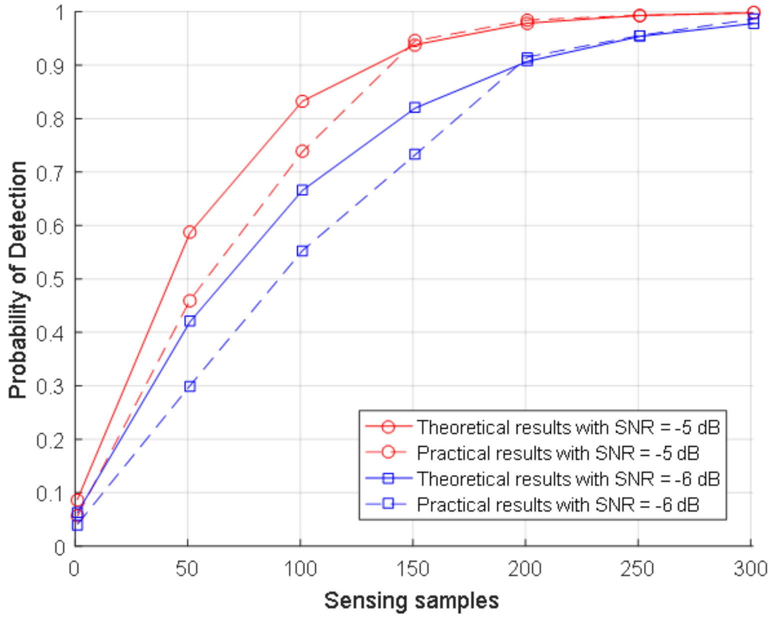


Figure 4.11: Detection Probability according to the number of sensing sample with $P_{(}f_a) = 0.05$.

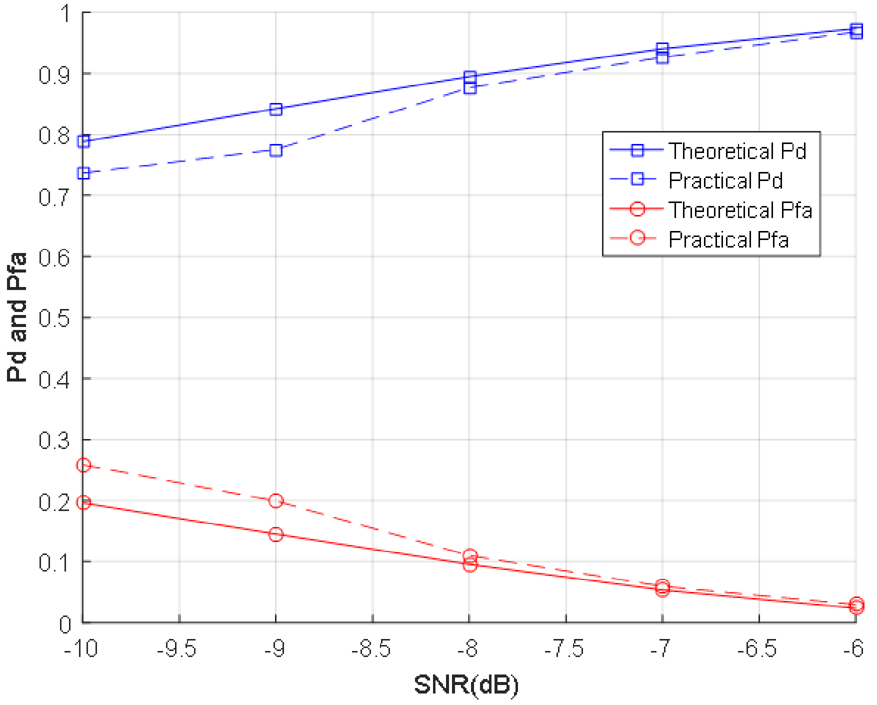


Figure 4.12: Probability of detection and probability of fals alarm according to SNR.

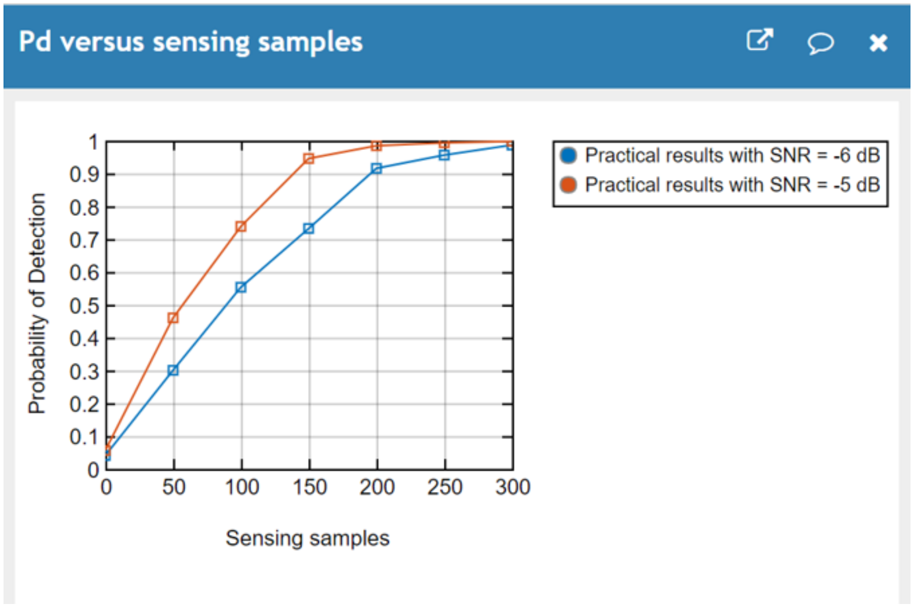


Figure 4.13: Detection probability uploaded in Thingsnetwork cloud.

Chapter 5

Testbed implementation of real LoRa network based on LoRaWAN Protocol

5.1 Introduction

Recently new transceiver technologies have emerged which enable power efficient communication over very long distances. Such Low-Power Wide Area Network (LPWAN) technologies are LoRa [37], Sigfox and Weightless. These news transceivers target applications where thousands of devices are used in a large geographic area to collect sensor readings. These systems are used in a setup where simple devices send data in one hop networks which then forwards data over a fixed wired infrastructure to a data collection point. These transceivers are potentially very useful to build more generic Internet of Thing (IoT) networks incorporating multi-hop bidirectional communication enable sensing and actuation.

In this section, we do performance testing on such IoT network based on Semtechs $Lora^TM$ technology. We perform our tests by implementing a real LoRa network and then observing the traffic between IoT LoRa nodes and base stations (LoRa gateway) based on LoRaWAN protocol.

5.2 Fundamental of LoRa and LoRaWAN concept

LoRa (Long Range) is a proprietary spread spectrum modulation technique by Semtech. It is the physical layer or the wireless modulation utilized to create a long range communication link. Many legacy wireless systems use frequency shifting keying (FSK) modulation as the physical layer because it is a very efficient modulation for achieving low power. LoRa is based on chirp spread spectrum (CSS) which maintains the same low power characteristics as FSK modulation but significantly increases the communication range. CSS has been used in military and space communication for decades due to the long communication distances that can be achieved and robustness to interference, but LoRa is the first low cost implementation for commercial usage. The advantage of LoRa is in the technologys long-range capability which a single gateway can cover entire cities or hundreds of square kilometers. The range is highly depending on the environment or obstruction, but LoRa and LoRaWAN have a link budget greater than any other standardize communication technology. The link budget is given in (dB) and is the primary factor in determining the range in a given environment. The throughput and the range depend on tree main LoRa parameters which are the bandwidth (BW), coding rate (CR),and spreading factor (SF). The BW is the physical bandwidth for the RF modulation (e.g 125KHz),higher bandwidth allows for higher effective data rate, thus reducing transmission time at the expense reduced sensitivity. CR is for forwarding error detection and correction, such coding incurs a transmission overhead and the lower coding rate, while the higher CR overhead ratio e.g. with $CR = 4/(4 + CR)$ and the overhead ratio is 1.25 for $CR = 1$ which is the minimum value. Finally SF,which can be set from 6 to 12, is the ratio between the symbol rate and the chip rate. A higher SF increases the SNR, sensitivity and range, but also increases the air time of the packet. The LoRa physical layer may be used with any MAC layer, however, LoRaWAN is the currently proposed MAC which operates a network in a simple star topology, Mesh, P2P [38].

LoRaWAN(Long Range Wide Area Network) is a MAC layer protocol designed for large-scale public networks with a single operator. LoRaWAN defines the communication protocol and system architecture for the network. It is built using the Semtechs LoRa modulation scheme. Nodes transmit directly to a gateway which is powered and connected to a backbone infrastructure. Gateways are powerful devices with powerful radios capable to receive and decode multiple concurrent transmissions (up to 50). A network architecture for LoRa topology is shown in **Figure. 5.1**.

In a LoRaWAN network nodes are not associated with a specific gateway. Instead, data transmitted by a node is typically received by multiple gateways. Each gateway will forward the received packet from the end-node to the cloud-based network server via some backhaul (either cellular, Ethernet, satellite, or Wi-Fi). The intelligence and complexity is pushed to the network server, which manages the network and will filter redundant received packets, perform security checks, schedule acknowledgments through the optimal gateway, and perform adaptive data rate, etc. If a node is mobile or moving there is no handover needed from gateway to gateway, which is a critical feature to enable asset tracking applications a major target application vertical for IoT.

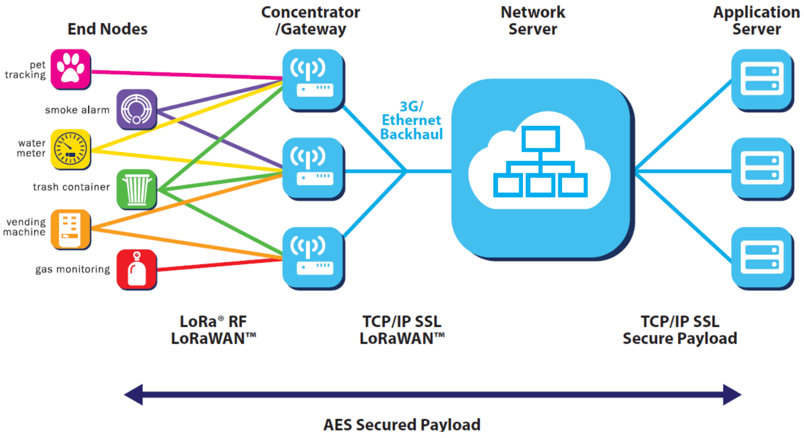


Figure 5.1: Advanced LoRaWAN Network Topology. [38].

5.2.1 Battery Lifetime

The nodes in a LoRaWAN network are asynchronous and communicate when they have data ready to send whether event-driven or scheduled. This type of protocol is typically referred to as the Aloha method. In a mesh network or with a synchronous network, such as cellular, the nodes frequently have to wake up to synchronize with the network and check for messages. This synchronization consumes significant energy and is the number one driver of battery lifetime reduction. In a recent study and comparison done by GSMA of the various technologies addressing the LPWAN space, LoRaWAN showed a 3 to 5 times advantage compared to all other technology options.

5.2.2 Network Capacity

In order to make a long-range star network viable, the gateway must have a very high capacity or capability to receive messages from a very high volume of nodes. High network capacity in a LoRaWAN network is achieved by utilizing adaptive data rate and by using a multichannel multi-modem transceiver in the gateway so that simultaneous messages on multiple channels can be received. The critical factors effecting capacity are the number of concurrent channels, data rate (time on air), the payload length, and how often nodes transmit. Since LoRa is a spread spectrum-based modulation, the signals are practically orthogonal to each other when different spreading factors are utilized. As the spreading factor changes, the effective data rate also changes. The gateway takes advantage of this property by being able to receive multiple different data rates on the same channel at the same time. If a node has a good link and is close to a gateway, there is no reason for it to always use the lowest data rate and fill up the available spectrum longer than it needs to. By shifting the data rate higher, the time on air is shortened opening up more potential space for other nodes to transmit. Adaptive data rate also optimizes the battery lifetime of a node. In order to make adaptive data rate work, symmetrical up link and down link is required with sufficient downlink capacity. These features enable a LoRaWAN network to have a very high capacity and make the network scalable.

A network can be deployed with a minimal amount of infrastructure, and as capacity is needed, more gateways can be added, shifting up the data rates, reducing the amount of overhearing to other gateways, and scaling the capacity by 6-8x. Other LPWAN alternatives do not have the scalability of LoRaWAN due to technology trade-offs, which limit downlink capacity or make the downlink range asymmetrical to the uplink range.

LoRa End-devices serve different applications and have different requirements. In order to optimize a variety of end application profiles, LoRaWAN utilizes different device classes such as class A, class B and class C.

Class A end-node devices: The End-devices of Class A allow for bi-directional communications whereby each end-devices uplink transmission is followed by two short downlink receive windows. The transmission slot scheduled by the end-device is based on its own communication needs with a small variation based on a random time basis

(ALOHA-type of protocol). This Class A operation is the lowest power end-device system for applications that only require downlink communication from the server shortly after the end-device has sent an uplink transmission. Downlink communications from the server at any other time will have to wait until the next scheduled uplink. Class B end-node devices with scheduled receive slots: The node behaves like a Class A node with additional receive windows at scheduled times. Gateway beacons are used for time synchronization of end-node devices which allows the server to know when the end device is listening. Class C end-node devices with maximal receive slots: these nodes are continuous listening which makes them unsuitable for battery powered operations.

5.3 Environment setup for LoRaWAN network

In this section we presents the implementation details for testing the Real LoRaWan networks. In order to implement LoRa network, microchip technology evolution kit hardware such as microchip LoRa gateway and mote are used. Microchip have introduced Evaluation kits for developers to integrate sensors into low-power wide-area network (LPWAN). Each kit includes two motes (LoRaWAN sensor board), a LoRaWAN gateway and a local LoRaWAN server application. Using LoRa semthec technology allows the modules to achieve a rang of 10 miles which means that relatively few gateways are required to implement a wide-area network of sensor and provide reliable internet connectivity. the microchip evolution is shown in **Figure. 5.2**. The microchip LoRa evolution include two kits, the DV164140-1 evolution kit which is suitable for European 868 MHz band and the DV164140-2 evolution kit which operates on the North American 915 MHz band. The kits contain two RN2903 Motes boards (type DV164140-1 and DV164140-2) fitted with a PIC18LF25K50 microcontroller, an OLED display, a light sensor, temperature sensor. It also includes a 6-channel Gateway (DV164140-1) or an 8-channel Gateway (DV164140-2), a local LoRaWAN Network/Application Server, Ethernet Cable, three USB Cables (Type-A to Micro-B),three Antennas and an SD Card. The purpose of this testbed is to show how sensors in different locations, the sensor data will be sent to the gateway/server via LoRa technology. Our testbed consist of one gateway and one node which sends temperature data and light data to the gateway then the gateway forwards the data to a LoRa server. A LoRa system can be implemented in many different ways, the operation of our evaluation system will be configured as shown in **Figure. 5.3**.

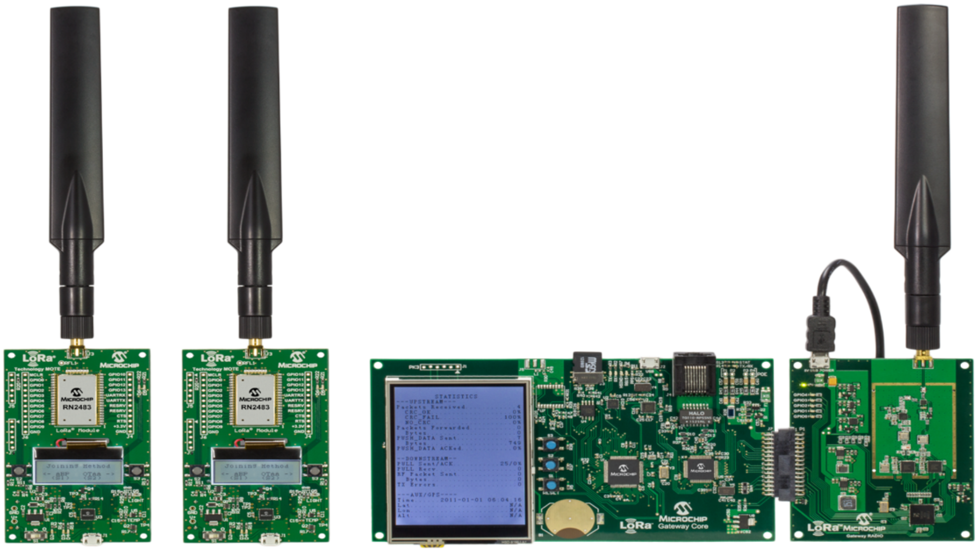


Figure 5.2: Microchip LoRa evolution kits .

The LoRa Gateway board, consisting of the Core board, and Radio board attachment, is connected to the Host PC through USB, supplying both power and Serial communication. Additionally, an Ethernet cable is connected between the Core board and the PCs Local Area Network (LAN) connector; this is used for communication between Gateway and Server. The RN Module populated development board (Mote or PICtail/PICtail

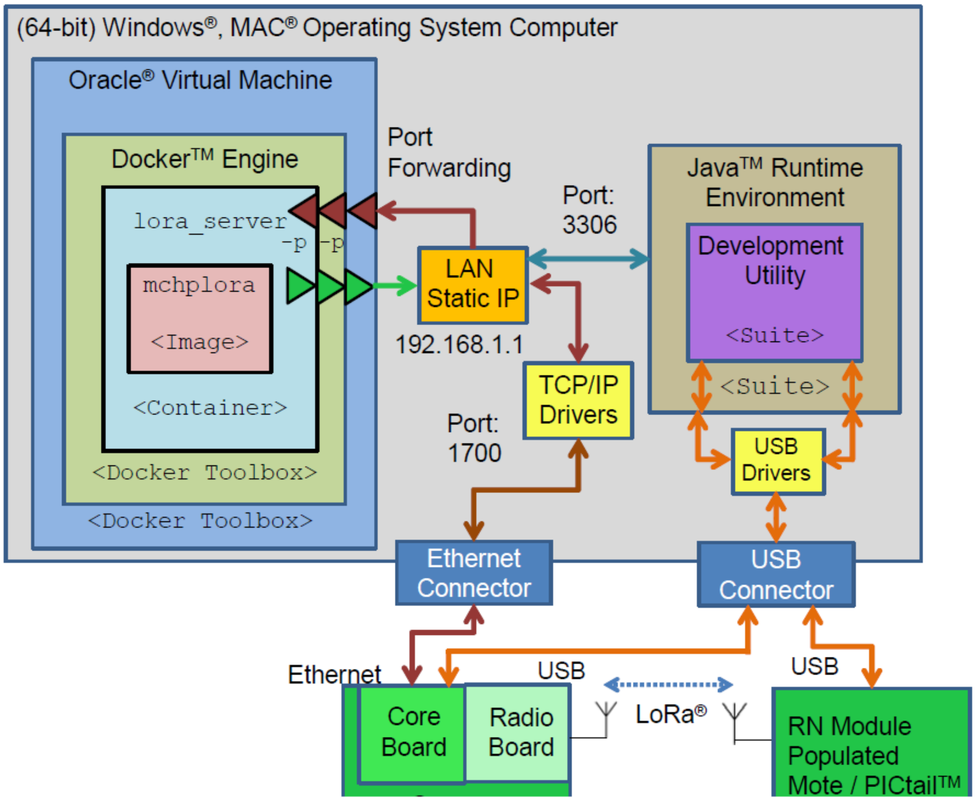


Figure 5.3: Microchip LoRa network evaluation kit implement a complete LoRaWAN network application. [39]

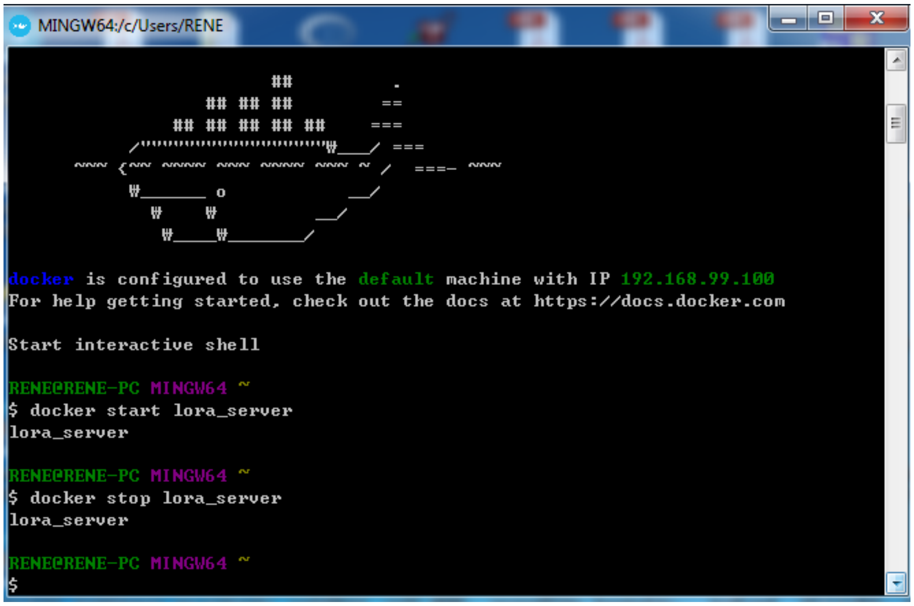
Plus) is connected to the same computer through its own USB connection, supplying both power and Serial communication.

From the Host PC, the program Docker Toolbox will be used. Using the Docker Toolbox allows the Oracle Virtual Machine to operate, hosting the Docker Engine used to run the LoRa evaluation server. A static IP address is assigned to the LAN adapter settings, allowing for Gateway Traffic to be forwarded through the PC (1700) port to the Docker hosted evaluation server. The LoRa Development Utility runs within the Java Runtime Environment. From the Utility commands and configurations, it can be issued from the PC to connected LoRa Technology Evaluation Kit boards. Database information is exchanged between the LoRa Utility and Docker hosted evaluation server, through MySQL traffic

(3306) port forwarding.

5.3.1 Server, Gateway and RN Mote configuration

On the software side, the kit uses the Microchip LoRa technology evaluation Suite which provides all the software components required. The Suite provides a network server (mchplora) as a docker container designed to run on virtual machine in a development system. The gateway board connects to the development system through USN and communicates wirelessly with the Mote board. The whole setup of LoRa server, gateway and RN node with their operation system based LoRaWan Network are shown in **Figure. 5.4**,**Figure. 5.5**, **Figure. 5.6**, **Figure. 5.7**, respectively.



```
                        ##         .
                  ## ## ##        ==
               ## ## ## ## ##    ===
           /""""""""""""""""""₩___/ ===
      ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
           ₩______ o           __/
             ₩    ₩         __/
              ₩____₩_______/

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Start interactive shell

RENE@RENE-PC MINGW64 ~
$ docker start lora_server
lora_server

RENE@RENE-PC MINGW64 ~
$ docker stop lora_server
lora_server

RENE@RENE-PC MINGW64 ~
$
```

Figure 5.4: LoRa server setup.

When you connect a node to a LoRa gateway, we need some amount of security and trust to be established among them. Due that there are two connection modes, namely activation by personalization (ABP) and over the air activation (OTAA) respectively, and we

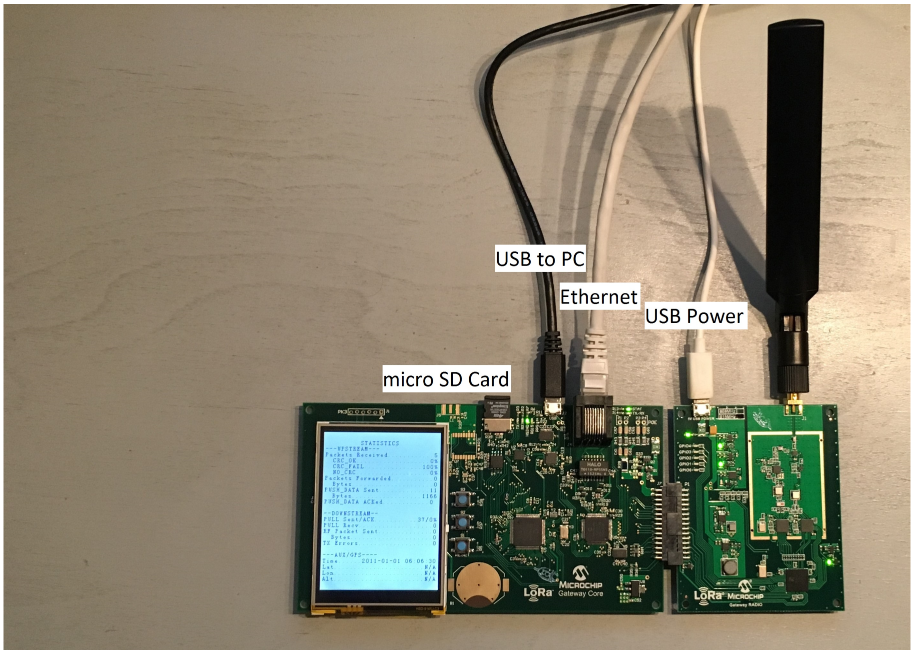
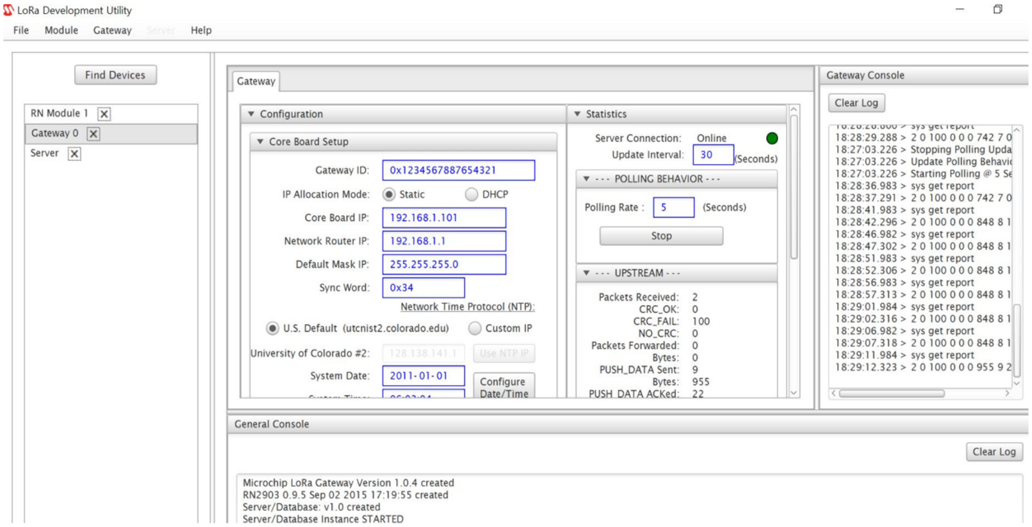


Figure 5.5: LoRa gateway connecting to host Pc.

distinguish between them using the criteria of security and ease of implementation. In ABP activation the device can be made operational in little time, but the drawback of this mode is that the encryption keys enabling communication with the network are preconfigured in the device, the key can be stolen, the device's identity retrieved and the collected data corrupted. In order to properly set up the connection to the network and identify the object, we need some information as follows: AppEUI which is a unique application identifier used to group objects. This address, 64 bit, is used to classify the peripheral devices by application. The second information is DevEUI, this identifier factory set, makes each object unique. In principle this setting cannot be adjusted. the thirst one is the AppKey which means a secret key shared between the peripheral and device and the network. It is used to determine the session keys and can be adjusted. the forth one is DevAdd this is a logical address used to identify the object on the network. The next is NetSKey (Network session key), encryption



Gateway ID: 0x1234567887654321

IP Allocation Mode: ● Static ○ DHCP

Core Board IP: 192.168.1.101

Network Router IP: 192.168.1.1

Default Mask IP: 255.255.255.0

Sync Word: 0x34

Network Time Protocol (NTP):

● U.S. Default (utcnist2.colorado.edu) ○ Custom IP

University of Colorado #2: 128.138.141.1

System Date: 2011-01-01

Server Connection: Online

Update Interval: 30 (Seconds)

--- POLLING BEHAVIOR ---

Polling Rate : 5 (Seconds)

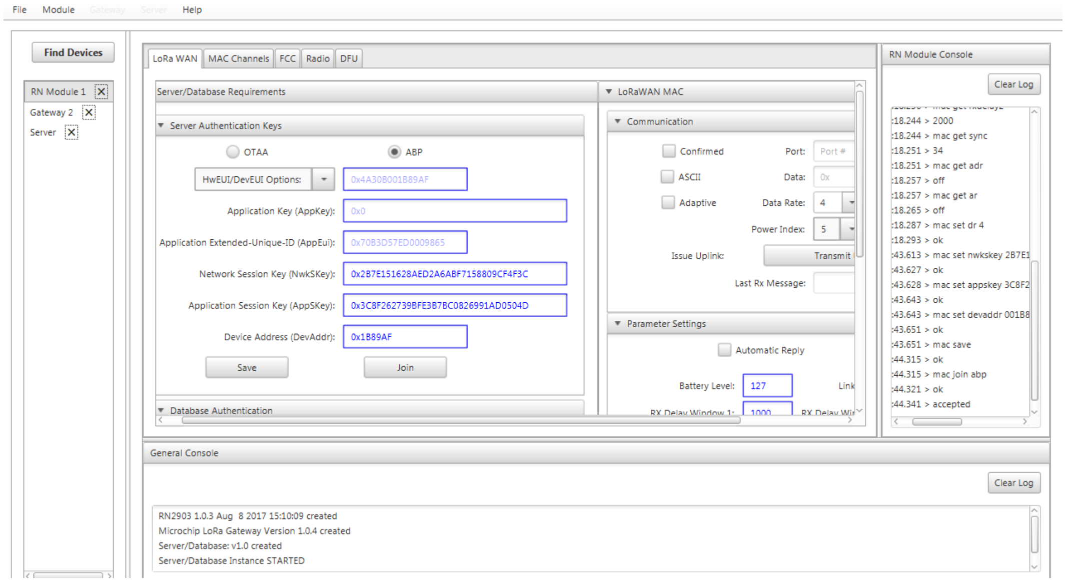
--- UPSTREAM ---

Packets Received: 2
CRC_OK: 0
CRC_FAIL: 100
NO_CRC: 0
Packets Forwarded: 0
Bytes: 0
PUSH_DATA Sent: 9
Bytes: 955
PUSH_DATA ACKed: 22

```
18:28:29.288 > 2 0 100 0 0 0 742 7 0
18:27:03.226 > Stopping Polling Upda
18:27:03.226 > Update Polling Behavi
18:27:03.226 > Starting Polling @ 5 Se
18:28:36.983 > sys get report
18:28:37.291 > 2 0 100 0 0 0 742 7 0
18:28:41.983 > sys get report
18:28:42.296 > 2 0 100 0 0 0 848 8 1
18:28:46.982 > sys get report
18:28:47.302 > 2 0 100 0 0 0 848 8 1
18:28:51.983 > sys get report
18:28:52.306 > 2 0 100 0 0 0 848 8 1
18:28:56.983 > sys get report
18:28:57.313 > 2 0 100 0 0 0 848 8 1
18:29:01.984 > sys get report
18:29:02.316 > 2 0 100 0 0 0 848 8 1
18:29:06.982 > sys get report
18:29:07.318 > 2 0 100 0 0 0 848 8 1
18:29:11.984 > sys get report
18:29:12.323 > 2 0 100 0 0 0 955 9 2
```

```
Microchip LoRa Gateway Version 1.0.4 created
RN2903 0.9.5 Sep 02 2015 17:19:55 created
Server/Database: v1.0 created
Server/Database Instance STARTED
```

Figure 5.6: LoRa gateway configuration.



○ OTAA ● ABP

HwEUI/DevEUI Options: 0x4A30B001B89AF

Application Key (AppKey): 0x0

Application Extended-Unique-ID (AppEui): 0x70B3D57ED0009865

Network Session Key (NwkSKey): 0x2B7E151628AED2A6ABF7158809CF4F3C

Application Session Key (AppSKey): 0x3C8F262739BFE3B7BC0826991AD0504D

Device Address (DevAddr): 0x1B89AF

☐ Confirmed Port: Port #

☐ ASCII Data: 0x

☐ Adaptive Data Rate: 4

Power Index: 5

Issue Uplink:

Last Rx Message:

☐ Automatic Reply

Battery Level: 127

RX Delay Window 1: 1000

```
:18.244 > 2000
:18.244 > mac get sync
:18.251 > 34
:18.251 > mac get adr
:18.257 > off
:18.257 > mac get ar
:18.265 > off
:18.287 > mac set dr 4
:18.293 > ok
:43.613 > mac set nwkskey 2B7E1
:43.627 > ok
:43.628 > mac set appskey 3C8F2
:43.643 > ok
:43.643 > mac set devaddr 001B8
:43.651 > ok
:43.651 > mac save
:44.315 > ok
:44.315 > mac join abp
:44.321 > ok
:44.341 > accepted
```

```
RN2903 1.0.3 Aug  8 2017 15:10:09 created
Microchip LoRa Gateway Version 1.0.4 created
Server/Database: v1.0 created
Server/Database Instance STARTED
```

Figure 5.7: LoRa RN mote configuration.

key between the object and the operator used for transmission and validate the integrity of the message. finally the AppSkey (Application Session Key) which is the encryption key between the object and the user (via the application) used for transmission.

In OTAA activation the network generates and sends the encryption key which makes for more security, it is the most frequently used method in IOT/LoRaWAN. The drawback is that the object must implement this connection mechanism which adds an additional layer complexity. The OTAA also need some information in order to establish the connection network like the ABP.

5.4 LoRa gateway registration on The Things Network cloud

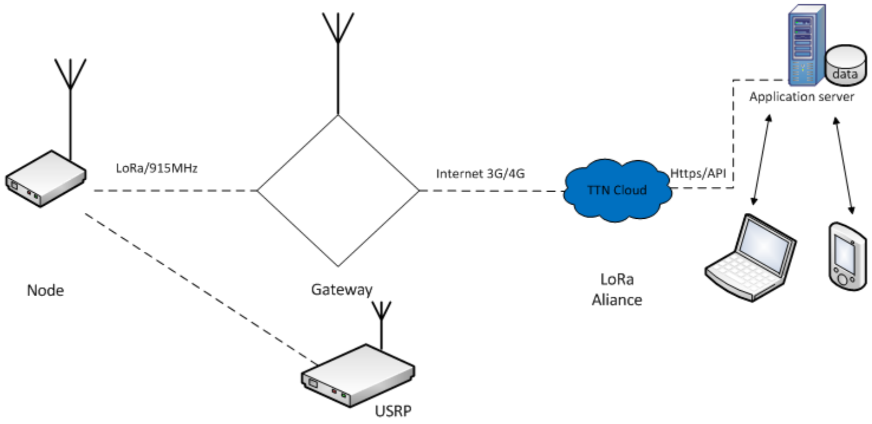
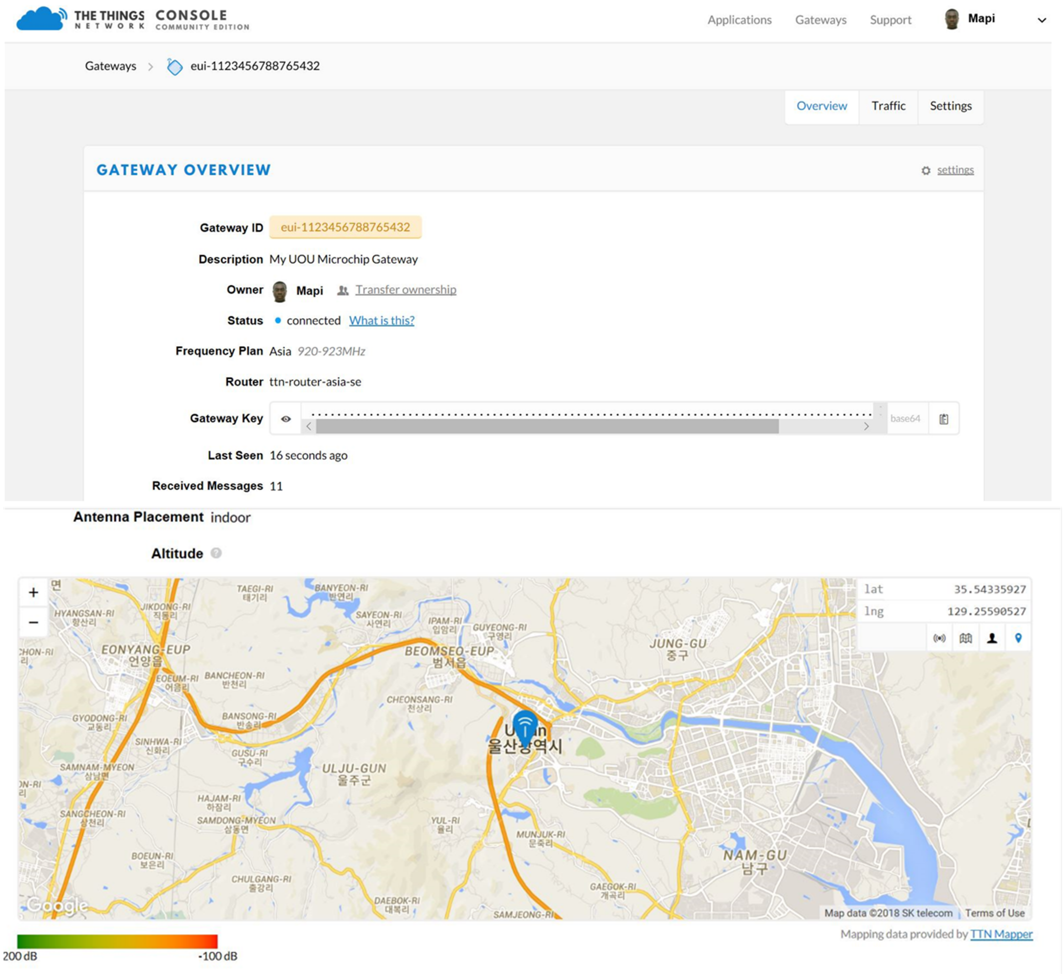


Figure 5.8: LoRaWAN network architecture with the things network representing a cloud interface.

In this section we integrate data uploading on the things network (TTN) console without running our own server. The things network is a global, crowdsourced, open free and decentralized internet of thing network. The network allows for things to connect to the internet using little power and little data. It makes use of the LoRaWan technology. Before processing a post data to TTN we need first to register our gateway which will be able to capture the data from the sensor node and create an application for LoRa device node. To use the default Over The Air Activation we need to register our device with its Device EUI. According to TTN webside, API works as shown in **Figure. 5.8**, the gateway registration is shown in **Figure. 5.9**, **Figure. 5.10** In this figure, we can observe the



Gateways > eui-1123456788765432

GATEWAY OVERVIEW

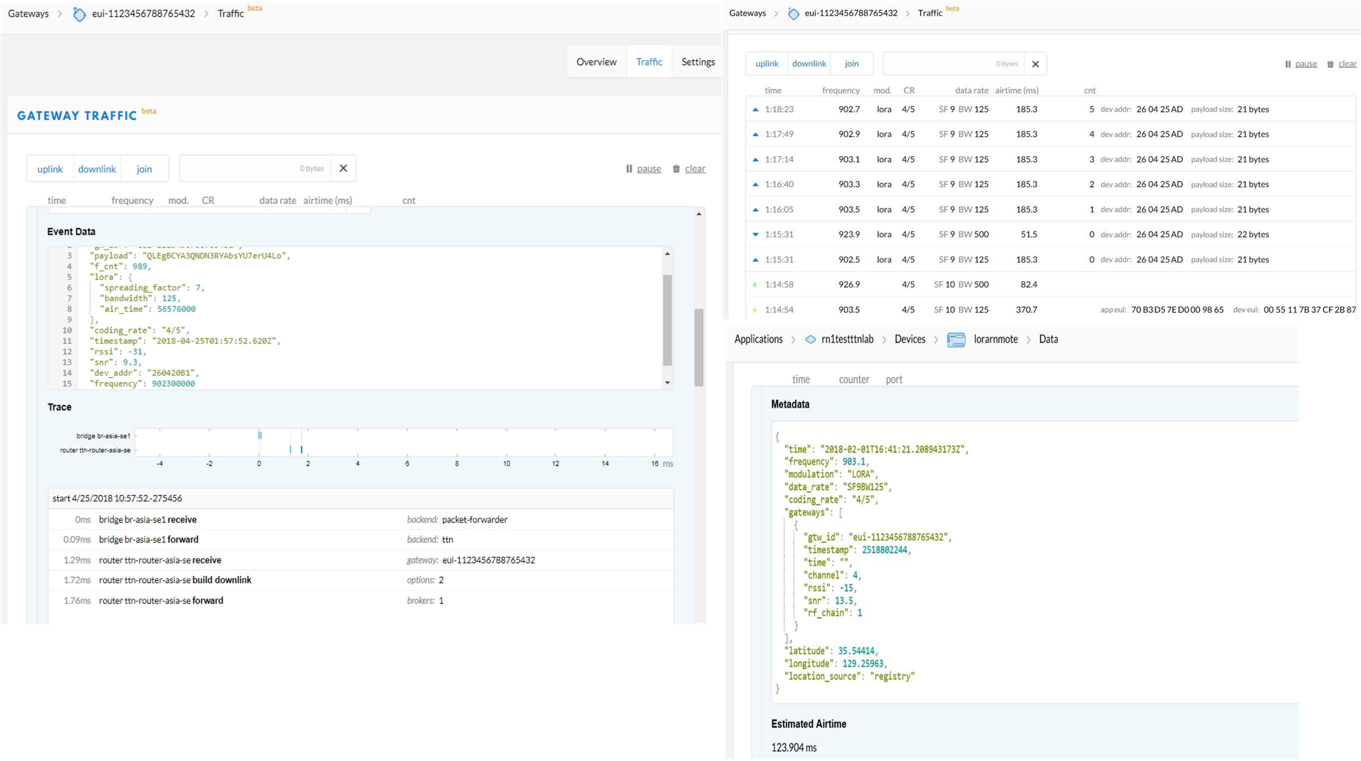
Gateway ID eui-1123456788765432
Description My UOU Microchip Gateway
Owner Mapi
Status connected
Frequency Plan Asia 920-923MHz
Router ttn-router-asia-se
Gateway Key ••
Last Seen 16 seconds ago
Received Messages 11
Antenna Placement indoor
Altitude

lat 35.54335927
lng 129.25590527

Mapping data provided by TTN Mapper

Figure 5.9: Gateway Registration on TTN.

data traffic from the sensor node captured by our gateway. As shown in the figure the data even shown the main parameter explained above. The SF 9, bandwidth 125 MHz,frequency 903.1 MHz and the CR was 4 was used by the node and most important is the RSSI and SNR for each received packet which can be evaluated. For this first measurements, we chose to focus on testing only the uplink from IoT node to the LoRa gateway, as this is the more common case in IoT. to perform better the LoRa range more measurements are needed using



```
"payload": "QLEgBCYA3QNDN3RYAbsYU7erU4Lo",
"f_cnt": 989,
"lora": {
  "spreading_factor": 7,
  "bandwidth": 125,
  "air_time": 56576000
},
"coding_rate": "4/5",
"timestamp": "2018-04-25T01:57:52.620Z",
"rssi": -31,
"snr": 9.3,
"dev_addr": "260420B1",
"frequency": 902300000
```

start 4/25/2018 10:57:52.-275456

0ms	bridge br-asia-se1 **receive**	*backend:* packet-forwarder
0.09ms	bridge br-asia-se1 **forward**	*backend:* ttn
1.29ms	router ttn-router-asia-se **receive**	*gateway:* eui-1123456788765432
1.72ms	router ttn-router-asia-se **build downlink**	*options:* 2
1.76ms	router ttn-router-asia-se **forward**	*brokers:* 1

	time	frequency	mod.	CR	data rate	airtime (ms)	cnt	
▲	1:18:23	902.7	lora	4/5	SF 9 BW 125	185.3	5	dev addr: 26 04 25 AD payload size: 21 bytes
▲	1:17:49	902.9	lora	4/5	SF 9 BW 125	185.3	4	dev addr: 26 04 25 AD payload size: 21 bytes
▲	1:17:14	903.1	lora	4/5	SF 9 BW 125	185.3	3	dev addr: 26 04 25 AD payload size: 21 bytes
▲	1:16:40	903.3	lora	4/5	SF 9 BW 125	185.3	2	dev addr: 26 04 25 AD payload size: 21 bytes
▲	1:16:05	903.5	lora	4/5	SF 9 BW 125	185.3	1	dev addr: 26 04 25 AD payload size: 21 bytes
▼	1:15:31	923.9	lora	4/5	SF 9 BW 500	51.5	0	dev addr: 26 04 25 AD payload size: 22 bytes
▲	1:15:31	902.5	lora	4/5	SF 9 BW 125	185.3	0	dev addr: 26 04 25 AD payload size: 21 bytes
⚡	1:14:58	926.9		4/5	SF 10 BW 500	82.4		
⚡	1:14:54	903.5		4/5	SF 10 BW 125	370.7		app eui: 70 B3 D5 7E D0 00 98 65 dev eui: 00 55 11 7B 37 CF 2B 87

```
{
  "time": "2018-02-01T16:41:21.208943173Z",
  "frequency": 903.1,
  "modulation": "LORA",
  "data_rate": "SF9BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-1123456788765432",
      "timestamp": 2518802244,
      "time": "",
      "channel": 4,
      "rssi": -15,
      "snr": 13.5,
      "rf_chain": 1
    }
  ],
  "latitude": 35.54414,
  "longitude": 129.25963,
  "location_source": "registry"
}
```

123.904 ms

Figure 5.10: Data traffic captured by the gateway when the sensor node started his transmission.

outdoor antenna. Using the API of TTN server, an application has been developed called ”lorarnmote”, it obtains record from the TTN server and shows the data in a graphical way. The application can subscribe a several number of sensor nodes (motes IDs) and show the temperature and light intensity curve. **Figure. 5.11** show the screenshot of the application one, we can observe the radio parameters such as SNR and RSSI. In **Figure. 5.12**, all the data are store in the Data base. We did outdoor test to evaluate the performance of LoRa according to certains distance.one, show in **Figure. 5.13** with one node that was periodically sending packets. We can also observe that the RSSI and the SRN given by the gateway in graph as shown in **Figure. 5.14** and in **Figure. 5.15** This figure show how the RSSI decreases as the distance increases, the SFs were chosen from set of 7, 8,9, 10. .

We also perform our test by using GNU Radio and USRP to capture and decode the LoRa signal in real time as shown in **Figure. 5.16**. We observe that our gateway and the the USRP can capture and receive data and signal in the same time. In this case we fixed the frequency centered at 902.3 for the the gateway and USRP.

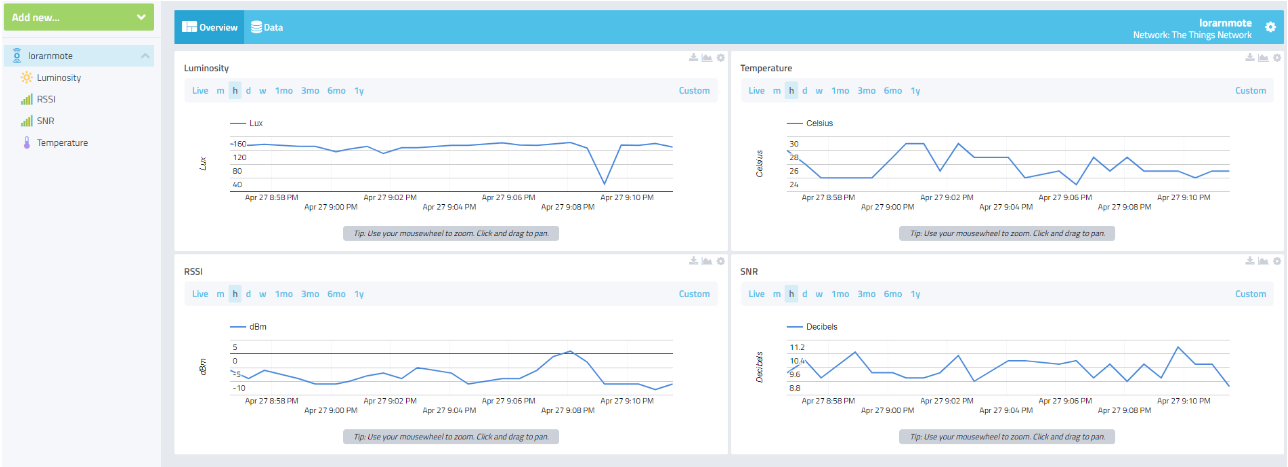
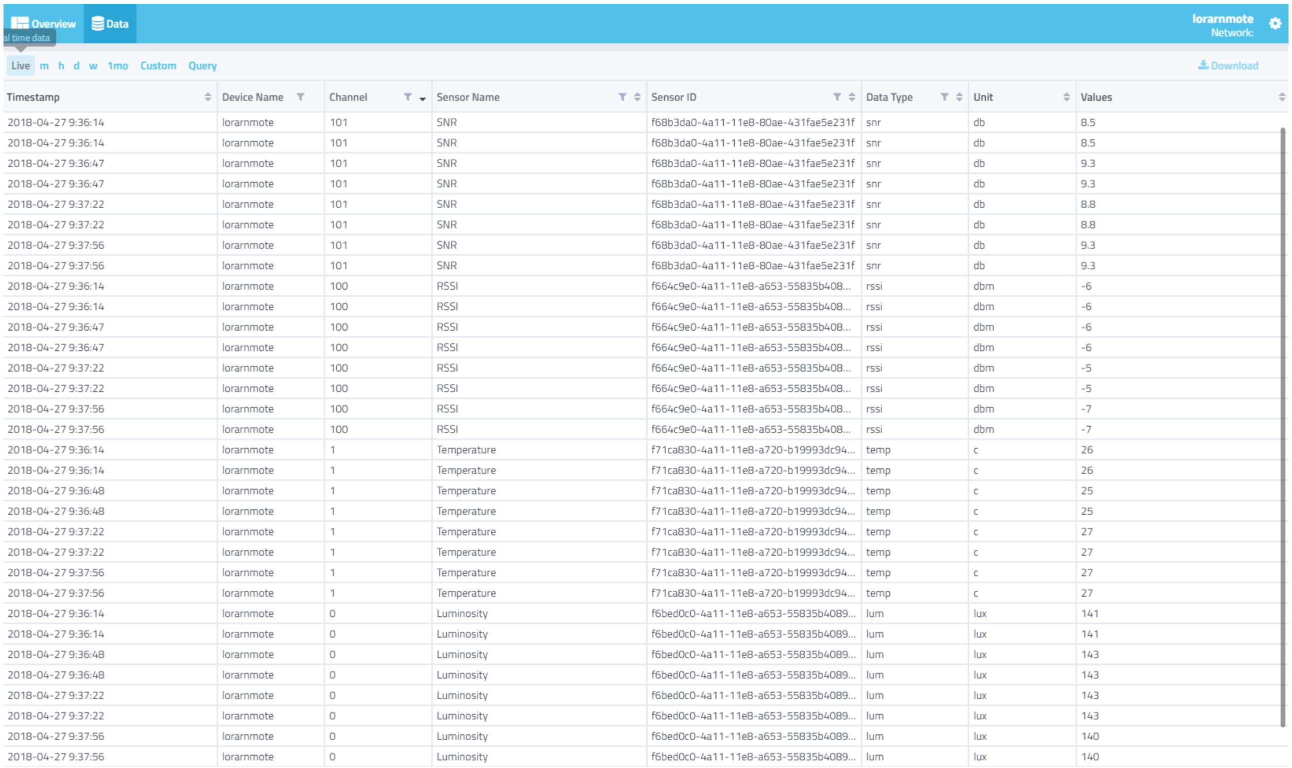


Figure 5.11: received measurement of RSSI, SNR, Temperature and Light intensity in real time on TTN cloud. .



Timestamp	Device Name	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2018-04-27 9:36:14	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	8.5
2018-04-27 9:36:14	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	8.5
2018-04-27 9:36:47	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	9.3
2018-04-27 9:36:47	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	9.3
2018-04-27 9:37:22	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	8.8
2018-04-27 9:37:22	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	8.8
2018-04-27 9:37:56	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	9.3
2018-04-27 9:37:56	lorarnmote	101	SNR	f68b3da0-4a11-11e8-80ae-431fae5e231f	snr	db	9.3
2018-04-27 9:36:14	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-6
2018-04-27 9:36:14	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-6
2018-04-27 9:36:47	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-6
2018-04-27 9:36:47	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-6
2018-04-27 9:37:22	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-5
2018-04-27 9:37:22	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-5
2018-04-27 9:37:56	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-7
2018-04-27 9:37:56	lorarnmote	100	RSSI	f664c9e0-4a11-11e8-a653-55835b408...	rssi	dbm	-7
2018-04-27 9:36:14	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	26
2018-04-27 9:36:14	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	26
2018-04-27 9:36:48	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	25
2018-04-27 9:36:48	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	25
2018-04-27 9:37:22	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	27
2018-04-27 9:37:22	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	27
2018-04-27 9:37:56	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	27
2018-04-27 9:37:56	lorarnmote	1	Temperature	f71ca830-4a11-11e8-a720-b19993dc94...	temp	c	27
2018-04-27 9:36:14	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	141
2018-04-27 9:36:14	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	141
2018-04-27 9:36:48	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	143
2018-04-27 9:36:48	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	143
2018-04-27 9:37:22	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	143
2018-04-27 9:37:22	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	143
2018-04-27 9:37:56	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	140
2018-04-27 9:37:56	lorarnmote	0	Luminosity	f6bed0c0-4a11-11e8-a653-55835b4089...	lum	lux	140

Figure 5.12: Received data transmitted by the LoRa node on TTN cloud.

Latest packet:
2018-05-08 11:56:54
Received by: eui-1123456788765432
RSSI: -112.0dBm
SNR: 0.3dB
Distance: 0m

60 packets

Mapping

Figure 5.13: Received packets at UOU campus.

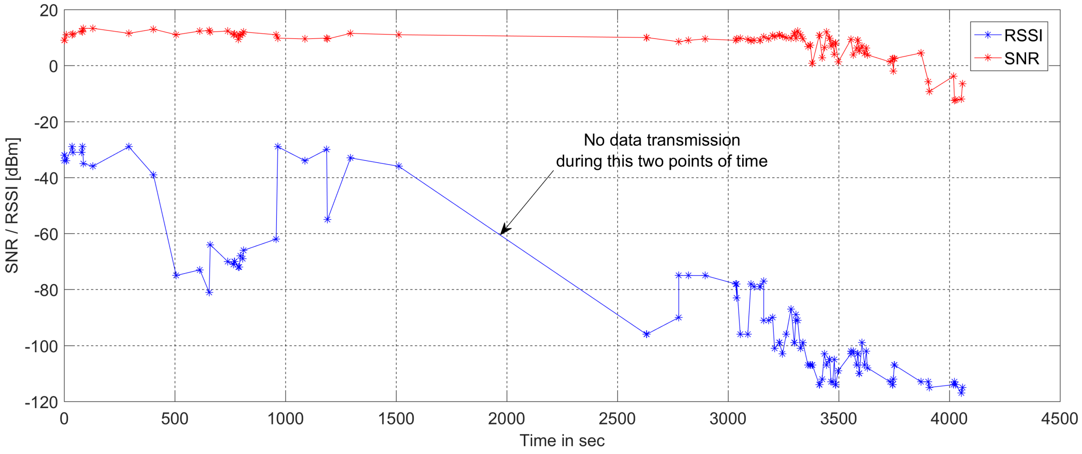


Figure 5.14: Measured average RSSI and SNR as a function of time by the gateway with outdoor antenna at UOU campus.

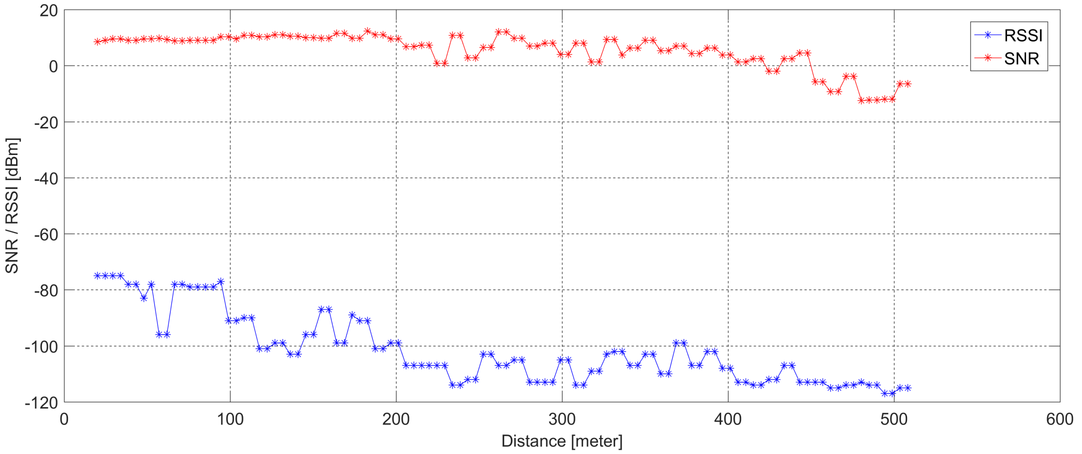
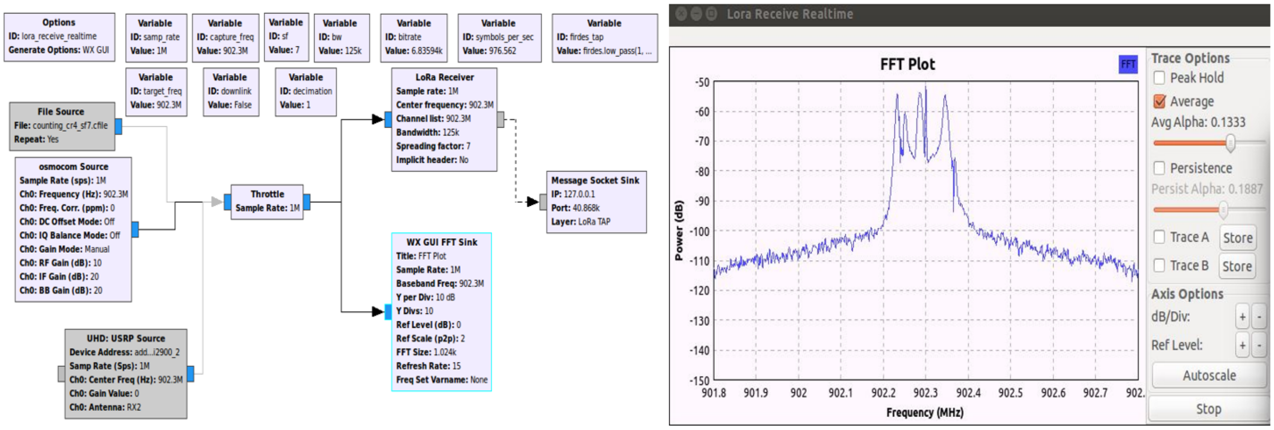


Figure 5.15: Measured average RSSI and SNR as a function of distance by gateway.



Options
ID: lora_receive_realtime
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 1M

Variable
ID: capture_freq
Value: 902.3M

Variable
ID: sf
Value: 7

Variable
ID: bw
Value: 125k

Variable
ID: bitrate
Value: 6.83594k

Variable
ID: symbols_per_sec
Value: 976.562

Variable
ID: firdes_tap
Value: firdes.low_pass(1, ...

Variable
ID: target_freq
Value: 902.3M

Variable
ID: downlink
Value: False

Variable
ID: decimation
Value: 1

File Source
File: counting_cr4_sf7.cfile
Repeat: Yes

osmocom Source
Sample Rate (sps): 1M
Ch0: Frequency (Hz): 902.3M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

UHD: USRP Source
Device Address: add...i2900_2
Samp Rate (Sps): 1M
Ch0: Center Freq (Hz): 902.3M
Ch0: Gain Value: 0
Ch0: Antenna: RX2

Throttle
Sample Rate: 1M

LoRa Receiver
Sample rate: 1M
Center frequency: 902.3M
Channel list: 902.3M
Bandwidth: 125k
Spreading factor: 7
Implicit header: No

Message Socket Sink
IP: 127.0.0.1
Port: 40.868k
Layer: LoRa TAP

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1M
Baseband Freq: 902.3M
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Trace Options
☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1887
☐ Trace A
☐ Trace B

Figure 5.16: Capturing LoRa spectrum signal centered on 902.3MHz using USRP device and GNU Radio when packets are being transmitted.

Chapter 6

Conclusion and Future Work

In this thesis, the aim was to implement a spectrum sensing system-based energy detection algorithm using NI USRP-2900 hardware and GNU Radio platform. The concept of CR was briefly introduced in the beginning of this thesis and the basic information about the hardware and the software used in our implementation environment also were introduced. Then the details information according to SS algorithm using NI USRP and GNU were explained. In this implementation two main performance metric were measured in real time environment, which are the probability of detection and the probability of false alarm. The P_D is used to determine the desired performance while P_{f_a} is used to determine the sensing threshold. According to the implementation test result presented, the ED based spectrum sensing can achieve higher detection performance when the sample sensing number is larger. This performance can be improved as well when a higher SNR is used. Thereafter we use a Thingspeak cloud to upload the data measured in GNU radio such that we can store and share the sensing information. In addition, on this thesis we designed and implement a testbed for a real LoRa network for IoT based on LoRaWAN technology. We described and analysed our LoRa setup in MSCL lab of University of Ulsan.

Then we upload the data measured by the gateway on TTN cloud.

Efficient connectivity is a fundamental requirement for IoT networks comprised of massive numbers of low-power IoT devices. LoRaWAN offers an effective IoT connectivity solution that offers long-range operation with minimal power requirements. As with any connectivity option, implementation can prove a major undertaking in itself, distracting developers from their primary focus on the IoT application itself. Based on Microchip Technology's RN2903 LoRa module, Microchip Technology's LoRa Network Evaluation Kit and accompanying LoRa Development Suite provide a complete LoRaWAN application. Using this combination of pre-certified hardware and software, developers can quickly bring up an IoT connectivity solution able to achieve a 15 km wireless range and 10-year battery life.

To perform these LoRa measurements we have to move on a trajectory(e.g going up to roughly 2km or 3km) for outdoor antenna then make a frequency map which the results will be shown in my final thesis report. We are also going to keep studying on various communication technology to offer services in a broaden on this research.

Publications

International journal

[1] 1. Huynh Thanh Thien, Rene Tendeng, Hiep Vu-Van, Insoo Koo, Implementation of Spectrum-Sensing for Cognitive Radio Using USRP with GNU Radio and a Cloud Server, J. lnf. Commun. Converg. Eng. (Scopus indexed), Vol. 16(1), pp. 23-30, Mar. 2018.

Bibliography

[1] J. Mitola "software radio survey, critical evaluation and future directions," *IEEE National Telesystem conference,*, pp.13/15 - 13/23, May. 1992.

[2] Haykin D. J. Thomson J. H. Reed "Spectrum sensing for cognitive radio," *Proceedings of the IEEE ,* vol. 97, no. 5, pp.849-877, 2009.

[3] Yucek and H. Arslan "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys and Tutorials,* vol. 11, no. 1, pp.116-130, First Quarter. 2009.

[4] E. Axell G. Leus E. G. Larsson H. V. Poor "Spectrum sensing for cognitive radio: State-of-the-art and recent advances,"*IEEE Signal Processing Magazine ,* vol. 29, no. 3, pp.101-116 , 2012.

[5] Y.C. Liang, A. T. Hoang Y. Zeng and R. Zhang "A Review on Spectrum Sensing for Cognitive Radio: Challenges and Solutions," *EURASIP Journal on Advances in Signal Processing,*, pp.1-15, 2010.

[6] J.Mitola and G.Q Maquire"Cognitive radio: making sofware radio more personal," *IEEE Personal Communications ,* vol. 6, no. 4, pp.13-18, Aug. 1999.

[7] Khaled Ben Letaief "Cooperative Spectrum Sensing ," *Cognitive Wireless Communication Network,* 2007.

[8] H. Urkowitz "Energy detection of unknown deterministic signals," *Proceedings of the IEEE,* vol. 55, no. 4, pp.523-531 , April. 1967.

[9] R. Tandra and A. Sahai "SNR walls for feature detectors," *in Proc. IEEE Int. Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland,* pp.559570, Apr. 2007.

[10] E. Blossom "Exploring GNU Radio, Nov. 2004," Available; http//www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html, Access date: Jan. 5, 2012.

[11] Gnu radio "[11] Gnu radio- the free and open-source software radio ecosystem," *[Online].Available: http://gnuradio.org,*

[12] Matt Ettus "Universal software radio peripheral," *http://www.ettus.com,*

[13] E.Blossom "Exploring GNU radio," *Avalaible: http:/www.gnu.org/software/gnuradio/doc/exploring-gnu radio.html,* [Access date: jan. 5, 2012]

[14] Ettus Research LLC "http://www.ettus.com," *IEEE National Telesystem conference,* April 2011.

[15] Shahzad A. et. al. "Comparative Analysis of Primary Transmitter Detection Based Spectrum Sensing Techniques in Cognitive Radio Systems," *Australian Journal of Basic and Applied Sciences, pp.4522-4531,* INSInet Publication. 2010.

[16] Matt Ettus, "Universal software radio peripheral," *http://www.ettus.com,*

[17] S.Dikmese, S.Srinivasan, M. Shaat, F. Bader and M.Renfors"Spectrum sensing and resource allocation for multicarrier cognitive radio systems under interference and power constraints,"*EURASIP Journal on Advances in Signal Processing* , 2014, 2014:68.

[18] Y.C. Liang, A. T. Hoang Y. Zeng and R. Zhang "A Review on Spectrum Sensing for Cognitive Radio: Challenges and Solutions,"*EURASIP Jor. on Ad. in Sig. Proc,*, pp. 1-15, January 2010.

[19] S. Dikmese and M. Renfors"Performance analysis of eigenvalue based spectrum sensing under frequency selective channels,"*in Proc. CROWNCOM 2012, Stocholm, Sweden,*, pp.356-361, 2012

[20] Dikmese, A. Gokceoglu, M. Valkama and M. Renfors"Reduced Complexity Spectrum Sensing Based on Maximum Eigenvalue and Energy,"*in Proc. ISWCS 2013, Ilmenau, Germany,* pp.1-5, 2013.

[21] S. Dikmese, J. L. Wong, A. Gokceoglu, E. Guzzon, M. Valkama and M. Renfors "Reducing computational complexity of eigenvalue based spectrum sensing for cognitive radio," *in Proc. CROWNCOM 2013, Washington DC, USA,* pp.61-67, 2013.

[22] S. Dikmese, P. C. Sofotasios, M. Renfors, M. Valkama "Reduced Complexity Energy and Eigenvalue Based Spectrum Sensing under Frequency-Selective Spectral Characteristics," Submitted to IEEE Transaction on Signal Processing 2014.

[23] S. Dikmese, P. C. Sofotasios, T. Ihalainen, M. Renfors and M. Valkama "Efficient Energy Detection Methods for Spectrum Sensing under Non-Flat Spectral Characteristics,"*accepted with minor revision for IEEE Journal on Selected Areas in Communications (JSAC),* Nov. 2014.

[24] S. Dikmese, S. Srinivasan and M. Renfors, Maquire“FFT and Filter Bank Based Spectrum Sensing and Spectrum Utilization for Cogntive Radios,” *in Proc. ISCCSP 2012,Rome, Italy,* pp.1-5, 2012.

[25] S. Dikmese, M. Renfors and H. Dincer“FFT and Filter Bank Based Spectrum Sensing for WLAN Signals,” *in Proc. 20th European Conference on Circuit Theory and Design (ECCTD), Linkoping, Sweden,*, pp.781-784, 2011.

[26] S. Dikmese, M. Renfors“Optimized FFT and filter bank based spectrum sensing for Bluetooth signal,” *2012 IEEE Wireless Communications and Networking Conference, (WCNC2012), Paris, France,* pp.803-807, 2012.

[27] R.Tandra and A.Sahai“SNR walls for signal detection,” *IEEE J.Selected Topics Signal Processing,* vol. 6, no. 4, pp.4-17, February. 2008

[28] GNU Radio“GNU Radio Community,” *[Online].Available:http://ww w.gnuradio.org,* accessed 05 June 2014.

[29] National Instrument Corporation“Labview,” *[Online].Available: http://www.ni.com/labview,*.

[30] Matlab“Matlab Simulink,” *[Online]. Available: http://www.mathworks.se/products /simulink,*accessed 05 June 2014.

[31] National Instrument Corporation*http://www.ni.com/documentation/en/software-defined-radio-device/latest/usrp-2900/block-diagram/,IEEE Pers. Commun,*.

[32] R. Jain“Channel Models: A Tutorial,” *WiMAX Forum AATG,* February 2007

[33] Sener Dikmese; Paschalis C. Sofotasios; Markku Renfors; Mikko Valkama“Subband Energy Based Reduced Complexity Spectrum Sensing Under Noise Uncertainty and

Frequency-Selective Spectral Characteristics," *IEEE Transactions on Signal Processing,* vol. 64, no. 1, pp.131 - 145 , September. 2016.

[34] A. Goldsmith, *Wireless Communications, Cambridge University Press, 2005,.*

[35] Federal Communications Commission"Part 74 Experimental radio, auxiliary, special broadcast and other program distribution services," *[Online]. Avaliable:http://www.gpo.gov/fdsys/pkg/CFR-2009-title47-vol4/pdf/CFR-2009-title47-vol4-part74.pdf,* accessed 05 June 2014

[36] Mohammad Tahir; Hafizal Mohamad; Nordin Ramli; Sigit P. W. Jarot "Experimental implementation of dynamic spectrum access for video transmission using USRP," *International Conference on Computer and Communication Engineering,*, pp.228 - 233, August 2012

[37] LoRa "lora-alliance," *[Online].Available: https://www.lora-alliance.org,* accessed 2015-11-07.

[38] LoRa"lora-alliance," *[Online].Available:https://lora-alliance.org/resource-hub/what-lorawantm,*

[39] Microchip"http://www.microchip.com/support/hottopics.aspx,"