Spring 2016

Design, Analysis, Implementation and Evaluation of Real-time Opportunistic Spectrum Access in Cloud-based Cognitive Radio Networks

Nimish Sharma

DESIGN, ANALYSIS, IMPLEMENTATION AND EVALUATION OF REAL-TIME OPPORTUNISTIC SPECTRUM ACCESS IN CLOUD-BASED COGNITIVE RADIO NETWORKS

by

NIMISH SHARMA

(Under the Direction of Danda B. Rawat)

ABSTRACT

Opportunistic spectrum access in cognitive radio network is proposed for remediation of spectrum under-utilization caused by exclusive licensing for service providers that are intermittently utilizing spectrum at any given geolocation and time. The unlicensed secondary users (SUs) rely on opportunistic spectrum access to maximize spectrum utilization by sensing/identifying the idle bands without causing harmful interference to licensed primary users (PUs). In this thesis, Real-time Opportunistic Spectrum Access in Cloud-based Cognitive Radio Networks (ROAR) architecture is presented where cloud computing is used for processing and storage of idle channels. Software-defined radios (SDRs) are used as SUs and PUs that identify, report, analyze and utilize the available idle channels. The SUs in ROAR architecture query the spectrum geolocation database for idle channels and use them opportunistically. The testbed for ROAR architecture is designed, analyzed, implemented and evaluated for efficient and plausible opportunistic communication between SUs.

Index Words: cognitive radio, primary user, secondary user, USRP, cloud computing

DESIGN, ANALYSIS, IMPLEMENTATION AND EVALUATION OF REAL-TIME OPPORTUNISTIC SPECTRUM ACCESS IN CLOUD-BASED COGNITIVE RADIO NETWORKS

by

NIMISH SHARMA

B.S., Old Dominion University, 2011

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment
of the Requirement for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

DESIGN, ANALYSIS, IMPLEMENTATION AND EVALUATION OF REAL-TIME OPPORTUNISTIC SPECTRUM ACCESS IN CLOUD-BASED COGNITIVE RADIO NETWORKS

by

NIMISH SHARMA

Major Professor:	Danda B. Rawat
Committee:	Fernando Rios-Gutierrez
	Mohammad Ahad

Electronic Version Approved:

May 2016

DEDICATION

To my family

ACKNOWLEDGMENTS

I would like to acknowledge the U.S. National Science Foundation (NSF) for their support in this research funded under the grant CNS-1405670. However, the research expressed in this thesis is accredited to me as the author and do not reflect the views of NSF.

Most importantly, I would like to thank my advisor Dr. Danda B. Rawat for his invaluable guidance in this research. Without Dr. Rawat, my involvement in the NSF funded research would not have been possible. He has provided supervision throughout my advance engineering degree and has helped me in every way possible to achieve my academic goals.

I would also like to acknowledge my friends Ms. Swetha Reddy, Mr. Tanjil Amin, Mr. Robin Grodi, Mr. Nickolas S. Holcomb, Mr. Isaac Cushman and Mr. Kishan Neupane for their support in my research. Without such helpful and noble friends, the research would not have been possible.

My advisor's and my friends' insights and experiences have given me various outlooks toward achieving the final result and have made the process of conducting research and preparing this thesis both educative and fun.

It has been a wonderful experience with faculties, staffs and my friends here at Georgia Southern University. A few words to acknowledge their selfless and unconditional contributions towards my degree would not be enough to portray all the wonderful experiences shared and cherished with them.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Communication systems have introduced plethora of technological advancements in information sharing. Every innovation in communication has propounded yet another revolutionary idea in the field of technology. Similarly, the birth of Internet can be accredited to contributions from communication systems. With the advent of Internet, many other technologies in communications are born. One such technologies is wireless-local-area-network (Wi-Fi) [1]. And not just Wi-Fi, but cellular technologies have evolved as well. These profound technologies use wireless media to transmit information to longer distances using electromagnetic waves and are entitled to mathematical models for wave-propagation proposed by many renowned scientists and mathematicians such as Maxwell, Hertz, Marconi, Tesla, and many more [2].

The advancement in wireless communication technology has pioneered the use of radio-waves to carry information far and wide. Different wireless communicating devices resort to using different frequencies of radio spectrum to transmit and receive information. Wireless frequencies used by simplex, half-duplex and duplex devices are static in nature and are assigned by Federal Communications Commission (FCC) [3]. Furthermore, FCC decides licensing for exclusive users and does not tolerate if any user circumvents its policies [4]. As a result, there is scantiness in publicly available frequencies as more and more Internet-of-Things (IoT) devices operating at FCC assigned frequencies crowd the congested and near saturated electronic device consumer market.

From the Figure 1.1, it can be seen that almost all of the radio spectrum from 3KHz to 300GHz has been assigned [5]. Due to such exclusive licensing, it becomes very hard to accommodate wireless devices that are expected to hit the market in the future. Each new wireless communicating device will then add problem to the preexisting wireless devices in operation. It has been estimated that by the end of 2020, there will be 50 billion wireless devices connected to the IoT [6]. When there are many wireless devices actively transmitting and receiving at the same frequencies, poor services, service blackouts and security vulnerabilities are inevitable. Scenarios such as these are realizable in day-to-day basis at crowded places such as airports, stadiums and colleges where there are higher demands for coverage and faster data-rate.

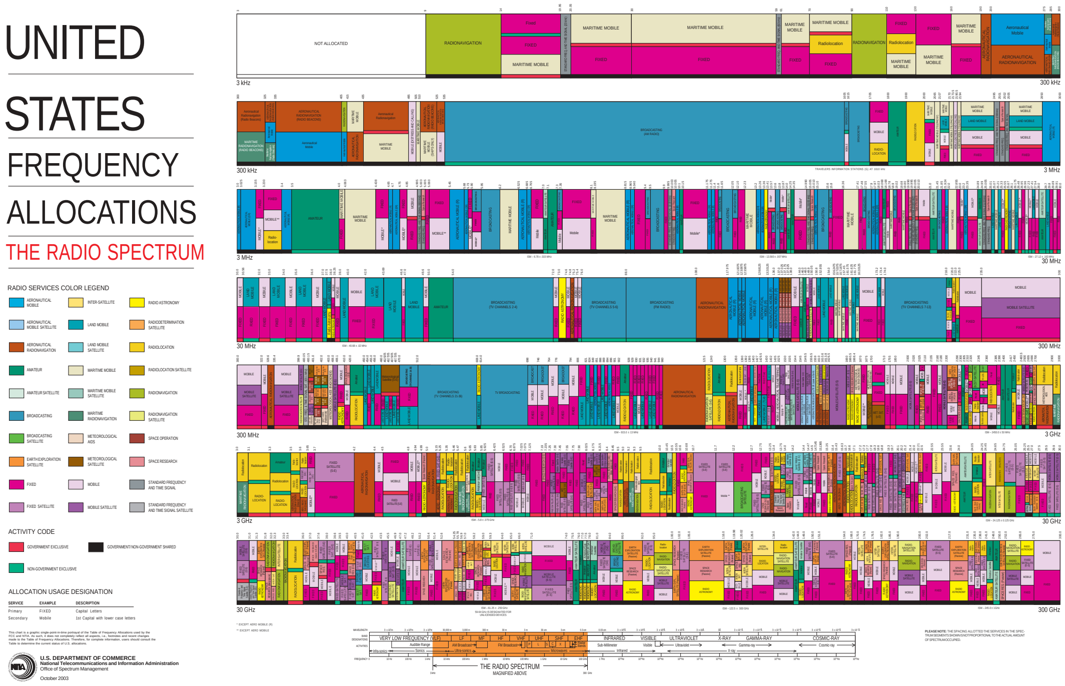


Figure 1.1: The U.S. frequency allocation from the U.S. Department of Commerce National Telecommunication and Information Administration

However, even though the frequencies are exclusively licensed, they are heavily under-utilized. In the Figure 1.2, it can be seen that in Chicago and New York City (urban areas), the licensed bands are under-utilized [7]. The maximum occupied spectrums are TV 2-6, RC: 54-88MHz and TV 7-13, 174-216MHz. Some of these highly utilized spectrums are not even 75% utilized. Even though there exists FCC exclusivity on licensed frequencies, these bands are completely under-utilized. Moreover, if the spectrums are under-utilized in

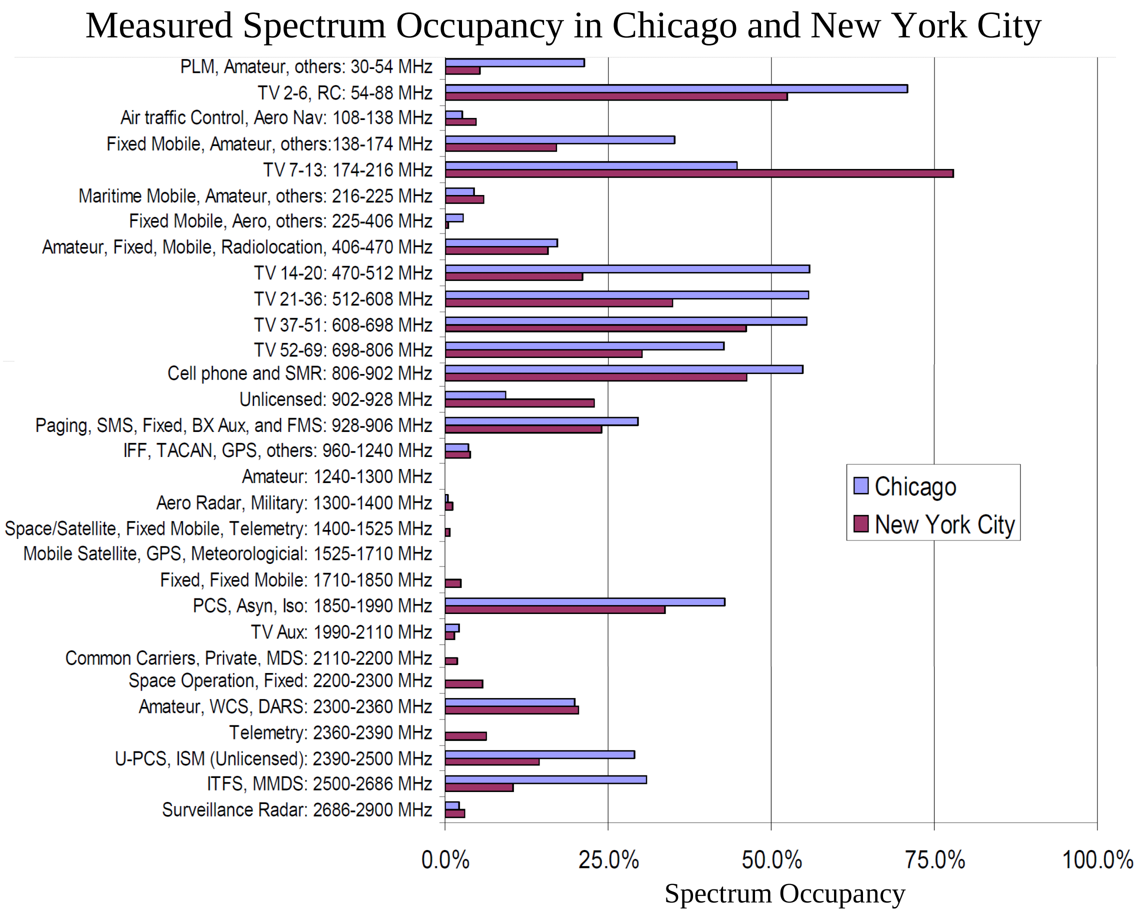


Figure 1.2: Measured spectrum occupancy information in New York City and Chicago (both urban areas)

urban areas, then the sub-urban and even the country side would be less utilized than the city.

Therefore, to address the issues of spectrum scarcity, better quality-of-service (QoS) (in crowded places like stadiums, airports), better quality-of-information (QoI) and sustained wireless communication, research in cognitive radio (CR) has become important.

What is Cognitive Radio?

The idea of CR is first proposed by Joseph Mitola III in his publications “Cognitive Radio: Making Software Radios More Personal” [8], “Cognitive Radio for Flexible Mobile Multimedia Communications” [9] and his doctoral dissertation “Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio” [10]. He introduced the idea of CR as the smart radio technology that is capable of machine learning and self-improvising techniques. Implementing cognition abilities in radios could provide them the capacity to decide the frequencies that are suitable for successfully communicating with multiple participants with minimal interference [11].

Aside from providing better services, CR prioritizes low interference communication. By being aware of its surroundings where there are multiple channels used, CR uses sophisticated algorithms to avoid any interference to the preexisting devices [12]. The radio does not just operate on programs running on them but in turn, it has the ability to learn about the presence or absence of idle spectrum in the environment and improve its spectrum sensing and selection process based on past experiences [10].

The initial idea proposed in CR offers a digitized bandwidth where modulation and demodulation are carried out in digital domain [13]. As shown in Figure 1.3, the CR uses analog-to-digital conversion (ADC) and digital-to-analog (DAC) conversion in software radio for transceivers. J. Mitola III shared his ideas on the transceiver in cognitive radios where the receiver carries out signal analysis when it demodulates the signal in digital domain. The transmitter on the other hand uses analog media to transmit after the DAC synthesize the radio frequency (RF) signal in digital domain. The concepts shared by J. Mitola III’s dissertation [10] is also put-forward in [13], that the CR can only be realizable for software radio proposal.

The CR architecture shares the abstract of autonomous RF selection and spectrum awareness capabilities [12] [13] [14]. Dynamic spectrum access (DSA) in CR utilizes the

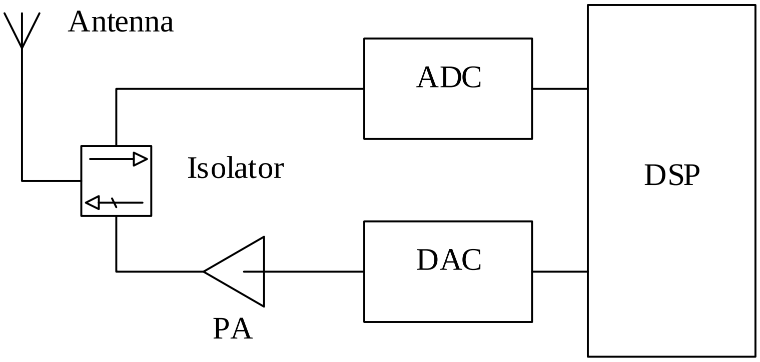


Figure 1.3: Cognitive radio transceiver conceptual design by J. Mitola III

opportunity to access the spectrums based on the awareness and availablity. DSA in CR takes alternative course of action such as selecting another available channel or waiting long enough for channel to be available so that it does not create harmful interference to any licensed primary users (PUs)[14]. This results in lower interference communication and proper utilization of spectrum in the environment.

Similarly, the importance of CR is conceptualized with two principle objectives [12]:

1. CR can efficiently utilize radio spectrum.
2. CR can provide reliable communication.

The requirements mandated by FCC [4] that the significant problem is imposed on efficient sharing of spectrum rather than the scarcity itself is focused in [12]. In summary, the CR is required because it focuses primarily on DSA where the radio learns from the environment and adapts to the interference whenever and wherever caused. That is also the reason the CR platform is the desired solution for spectrum under-utilization.

1.1 Motivation

The motivation behind the research in opportunistic spectrum access using CR system is to efficiently utilize unused RF spectrum. The motivation behind cloud-assisted CR network at the same time is to outsource memory, computational capacity and device power of CR system to infrastructure-based CR networks. There are many researches conducted in CR, but not many are conducted in realization of a system that is able to dynamically access under-utilized spectrums. Some testbeds such as IRIS [15], LOG-a-TEC [16], w-iLab.t [17], VT-CORNET [18] and ORBIT [19] are developed as CR testbeds, but most of these systems lack wideband spectrum sensing and use of distributed database for spectrum access. Because of the lack of wideband spectrum sensing, it becomes important to research CR and create a testbed that is able to sense the entire spectrum to opportunistically access them.

For example, one of the spectrums freely available for general use and authorized by FCC to be used without license is ISM band (2.4GHz and 5.0GHz) [20]. However, if all the wireless communication devices (that are not able to sense wideband) operate in ISM spectrum, it causes huge interference. Instead, if the devices could use licensed spectrum such as TV, FM, satellite and cellular (entire wideband) for temporary communication through opportunistic spectrum access without creating interference to PUs, the spectrum would be efficiently utilized [4]. Thus, the key motivator to the research presented in this thesis is to implement SDRs as CR that can sense wideband spectrum so that the unlicensed secondary users (SUs) (without sensing) can rely on opportunistic spectrum access to communicate based on the spectrum availability stored at distributed database. Due to the estimated devices that will be creating interference in spectrum occupancy, a viable solution sought to remedy the problem is the key objective of this research. As a result, opportunistic and DSA in CR is procured in this thesis.

The secondary motivation in this research is the FCC mandated geolocation database

[21]. It is a requirement endorsed by the FCC to use geolocation database to store the list of available spectrums so that the SUs could use them for communication. FCC prohibits sensing of spectrum by SUs to find idle channels due to security and privacy factors and has proposed the idea of geolocation database that store idle spectrum [22]. The geolocation database in-turn, resides on the cloud infrastructure where it stores idle channels in distributed manner [23]. Thus, the SUs in CR rely on the geolocation database technology to query the idle spectrums, eliminating the need for sensing the spectrums locally. By offloading the responsibilities of SUs to sense for the spectrum based on their geolocation, the FCC proposal not just addresses privacy and security issues, but also treats the problems faced by the devices that are unable to sense for wideband (50MHz to 6GHz). It is apparent that not every wireless communication device has the ability to sense every available spectrum in the environment. Thus, by leveraging the spectrum sensing tasks to CR sensors and reporting the sensed idle spectrum to the geolocation database, the SU devices that would otherwise end up using memory, power and computational capacity to search for spectrum prior to utilization is eliminated [24]. Ever since FCC mandated the use of database, many research has been carried in the field of geolocation database and cloud-based CR networks [25].

Therefore, an architecture is proposed keeping in mind the main purpose of this research, which is to carryout successful communication in wideband spectrum after sensing the available idle channels and sharing the channel information among multiple services and users without causing harmful interference to licensed PUs.

1.2 Problem Statement

The main objectives of this research are to:

1. Design, analyze, implement and evaluate Real-time Opportunistic Spectrum Access

in Cloud-based Cognitive Radio Networks (ROAR) architecture.

2. Sense heterogeneous spectrum for ROAR architecture.

3. Provide distributed computing to process requests from SUs for opportunistic spectrum access.

4. Implement quorum-based common channel selection for DSA.

5. Test and validate the ROAR architecture using SDRs.

1.3 Outline of Thesis

Chapter 2 provides ROAR architecture. In this chapter, detail description of the system model is discussed. Each component of ROAR architecture is explained and elaborated along with the FCC rules, regulations and policies. The device being used as ROAR architecture testbed is tested and configured.

Chapter 3 details the spectrum sensing for ROAR architecture. It provides theoretical idea and practical implementation of spectrum sensing. Various components used in spectrum sensing is designed, analyzed, implemented and evaluated in this chapter.

Chapter 4 discusses the distributed cloud-based computing and storage for ROAR architecture using Storm topology and Cassandra database. The availability of sensed spectrum is reported to the database and heat-map is generated for visualizing available idle spectrum data using Google map API.

Chapter 5 presents opportunistic spectrum access in CR networks where the spectrum availability information is used by SUs to communicate without causing harmful interference to the PUs. Quorum-based common channel selection discussed in this chapter provides secured and faster communication to different SUs.

Chapter 6 discusses and concludes the thesis and provides the future work.

This research has been published in 2015 IEEE 29th International Conference on Advanced Information Networking and Applications with the title “A Testbed Using USRP$^{\text{TM}}$ and LabVIEW$^{®}$ for Dynamic Spectrum Access in Cognitive Radio Networks” [26].

CHAPTER 2

ROAR SYSTEM ARCHITECTURE AND TESTBED VERIFICATION

In this chapter, ROAR architecture is presented that assists to design, analyze, implement and evaluate spectrum sensing and DSA testbed. The SDR USRP is configured and tested prior to designing and implementing in testbed.

2.1 ROAR System Model

The system model for ROAR architecture is shown in Figure 2.1.

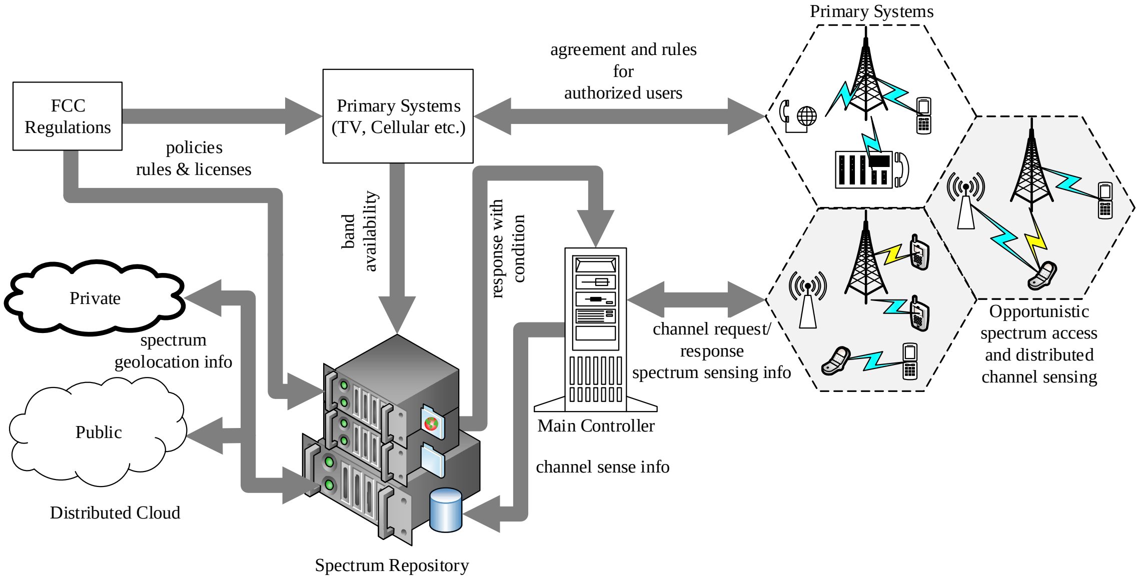

Primary Systems
agreement and rules for authorized users
FCC Regulations
Primary Systems (TV, Cellular etc.)
policies rules & licenses
band availability
response with condition
Private
spectrum geolocation info
Public
Distributed Cloud
Main Controller
channel request/ response spectrum sensing info
channel sense info
Opportunistic spectrum access and distributed channel sensing
Spectrum Repository

Figure 2.1: System model of ROAR architecture

The ROAR system model consists of spectrum sensing, reporting and opportunistic spectrum access. ROAR system model depicts the scenario where SUs are able to query for idle spectrums from the spectrum repository for opportunistic spectrum access to establish communications [27].

The system model can be explained in detail by referring to each components as:

Spectrum Sensing:

In the ROAR architecture, the spectrum sensing play a vital role in determining the idle spectrums. The ROAR architecture for spectrum sensing uses SDRs equipped with 50MHz to 6GHz wideband antenna for sensing available idle channels [27]. The SDR testbeds for SUs in ROAR architecture are the Universal Software Radio Peripheral (USRP) devices that are programmed using LabVIEW.

The primary systems such as TV stations, cellular stations, satellite stations, etc. use specific frequencies at given location that are sensed by the spectrum sensors equipped with GPS units. The channels that are being used by these primary systems are not reported to the spectrum repository. Instead, the sensors report only the available idle spectrums to prevent any SUs from accidentally accessing the channels being used by the primary systems. The sensors are deployed in such a way that the sensing of spectrum occupancy is carried out in a distributed manner and the sensed idle channels are reported to distributed spectrum repository. The spectrum information is later used by the SUs for opportunistic communication.

Spectrum Repository, Distributed-computing and Heat-map Interface:

In the ROAR system model shown in Figure 2.1, after the spectrums are sensed, idle spectrum are stored in the distributed database. The real-time spectrum availability information is encapsulated with time-stamp and geolocation information prior to storing the idle channel information in either private or public distributed cloud-based database. Here, the cloud-based database stores the information and cloud-computing deals with processing the spectrum requests from SUs during opportunistic spectrum access. Upon request from the SUs, the main controller from the system model is responsible for distributed-computing that queries the spectrum repository and provides the idle spectrum information depending on the geolocation and time based request. All of these information is handled in real-time by the means of Storm data processing in cloud-computing and by accessing the information stored in Cassandra database [23]. The stored information in Cassandra database is overlaid on Google maps through application programming interface (API) for visualization of idle spectrums. The detail of Storm topology, Cassandra database and Google maps API is discussed in Chapter 4.

Opportunistic Spectrum Access Based on Geolocation:

The idle spectrum information stored in the distributed cloud-based database is queried and used by the SUs for opportunistic spectrum access. The SUs equipped with their GPS units use location based request for available spectrums from the spectrum repository as shown in the system model. The system model in Figure 2.1 shows the opportunistic spectrum access where the SUs are able to establish communicaton after the main controller provides idle channel information from the spectrum repository [27]. For instance, if any secondary user, (say (SU_1)) is trying to communicate with any other secondary user (say (SU_2)) located in any area, SU_1 first queries for the available spectrums in the distributed database. Similarly, SU_2 queries for the available spectrums in similar manner. The main controller responsible for distributed computing then queries the distributed database with the nature of requests each SUs put. Based on the spectrum repository response depending on the distance of the SUs and the location of idle channels, the distributed computing decides the lists of available frequencies that can be provided and hands them over to the SUs. The SUs in-turn use quorum-based channel selection to find a common channel and establish communication.

FCC Regulations and Policies:

FCC regulations and policies are the key factors in determining the spectrum usability across various services. The FCC regulations and licensures enforce spectrum usability to licensed primary systems (TV stations, cellular stations, satellite stations, etc.) as well as unlicensed spectrum through Integrated Spectrum Auction System (ISAS) [28]. FCC directly intervenes by revocating the license to transmit or receive if it finds any unauthorized access or any circumvention to its policies by either PUs or the SUs [4]. Therefore, FCC is the primary legal body in wireless communication that determines if the system can use any spectrum legally or not to prevent intentional interference. The system repository is constantly updated with FCC regulations and policies. Without such policies massive disruption in services and haphazard spectrum usage can be expected [27].

The FCC policies are incorporated in the ROAR architecture and the ROAR architecture model is highly efficient in determining opportunistic spectrum access for idle channels in the environment without costing the SUs their memory, power and computational capacity [29]. By implementing FCC rules and regulations, the ROAR architecture is able to comply with federal laws involved in wireless communication and is able to provide uninterrupted communication to the primary services, PUs and SUs by sharing under-utilized spectrum.

Prior to developing the ROAR architecture, the testbed devices are tested for accuracy and reliability. The detail on configuring the testbed is provided in the next section.

2.2 Device Configuration for ROAR Architecture

For designing, implementing, analyzing and evaluating the ROAR architecture, USRP devices are used. These devices have different models (2920 and 2921) with different sensing capabilities. USRP 2920 is capable of sensing 50MHz to 2.2GHz and USRP 2921 is capable of sensing 2.4GHz to 6GHz [30]. In order to sense wideband (50MHz to 6GHz), these devices are connected using a MIMO cable. Prior to designing a testbed for ROAR architecture, research is conducted to search for similar prototyping done in the field of spectrum sensing using SDRs and USRPs.

The architecture of SDR system as cognitive wireless cloud (CWC) is introduced in [31]. The idea and innovation involved in this research is user centric and scalable. The new technology to support SDR where CWC hosts multiple services and provide seamless interaction with mobile devices depending on the spectrum requested is being proposed. The hardware consists of multiband antennas (similar to the ROAR architecture), radio frequency unit, signal processing unit, software platform hosting reconfiguration management module highly capable of policy detection, decision making action policy and band selection. The proposed architecture is developed and the performance is tested for 400MHz to 6GHz with 1650ms reconfiguration time and sensing latency. However, the platform required to develop CWC and automate the process of spectrum sensing using programmable language is not covered in [31].

In [15], IRIS testbed is created that is software dependent. Due to layered construct of IRIS testbed, each component can be tweaked using software. This concept is useful for ROAR architecture where if minor components such as gain, data sampling and decision threshold could be dynamically changed using software, better spectrum sensing can be achieved. Similarly, the IRIS system is runtime configurable, supports components based architecture and incorporates physical layer flexibility that are similar to the ROAR architecture. However, unlike the ROAR architecture, IRIS operates in VHF and UHF TV spectrum and does not incorporate entire wideband spectrum.

Similarly, the ORBIT testbed is only capable of 1GHz spectrum sensing and reporting ability [19]. The VT-CORNET [18] uses USRP similar to ROAR architecture. However the testbed in VT-CORNET is an open source testbed whereas the one being used in ROAR architecture is a combination of LabVIEW dependent USRP and GNU based radio. The VT-CORNET is capable of spectrum sensing from 100MHz to 4GHz, however the ROAR architecture is capable of 50MHz to 6GHz. The LOG-a-TEC testbed uses Versatile Sensor Network Application (VESNA) platform as wireless sensor monitoring and re-configuring framework [16]. The LOG-a-TEC testbed is a wireless sensor network and not a fully realizable CR network. The ROAR architecture involves sophisticated SDR with CR abilities when it comes to configurability and programmability. The LOG-a-TEC on the other hand is a wireless network device that has limited spectrum sensing and accessing capabilities.

Based on various speculation of SDRs, the USRP is chosen as a plausible option for ROAR architecture. USRP is capable of various operations guided by the software created in LabVIEW and is able to sense and analyze the received signals. The USRP is easily configurable and provides wide range of spectrum sensing and reporting. The USRP can be separately speculated as hardware and software components synchronized to work in conjunction as discussed below:

Hardware:

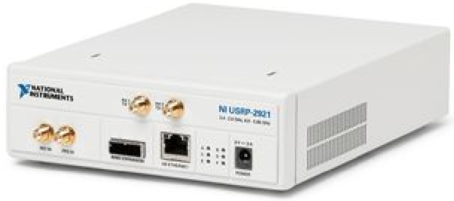


Figure 2.2: National Instruments USRP 2921 hardware as a testbed

For creating a testbed for ROAR architecture, NI-USRP 2921 is used for 2.4-2.5GHz and 4.9-5.85GHz spectrum sensing. The tests are reliable for NI-USRP 2920 that covers 50MHz to 2.4GHz as well. These devices can be connected to each other using MIMO cables.

Frequency range	2.4 to 2.5GHz; 4.9 to 5.9GHz
Frequency step	< 1KHz
Maximum output power	2.4 - 2.5GHz: 50mW to 100mW 4.9 - 5.9GHz: 50mW to 100mW
Gain range	0dB to 35dB
Gain step	0.5dB
Frequency accuracy	2.5ppm
Max. instantaneous real time bandwidth	16-bit sample width - 24MHz 8-bit sample width - 48MHz
Max. I/Q sample rate	16-bit sample width - 25MS/s 8-bit sample width - 50MS/s
DAC converter	2-channels, 400MS/s, 16-bit
DAC spurious-free dynamic range	80dB

Table 2.1: National Instruments USRP 2921 transmitter specifications

As shown in Figure 2.2, the USRP is a rugged designed hardware with two channel dedicated for two different antennas with 50Ω impedance. The Table 2.1 and 2.2 give the overall hardware specification and accuracy of the testbed being used as transmitter and receiver[30].

Along with NI-USRP, laptop and GPS units are being used for spectrum sensing and dynamic spectrum access for ROAR architecture. The transceivers designed uses laptops connected to USRP through gigabit Ethernet for faster reliable connection. The hardware components are individually programmed. The hardware components does not require any soldering or tampering to complete the research work.

Frequency range	2.4 to 2.5GHz; 4.9 to 5.9GHz
Frequency step	< 1KHz
Maximum input power	-15dBm
Gain range	0dB to 92.5dB
Gain step	2dB
Frequency accuracy	2.5ppm
Max. instantaneous real time bandwidth	16-bit sample width - 19MHz 8-bit sample width - 36MHz
Max. I/Q sample rate	16-bit sample width - 25MS/s 8-bit sample width - 50MS/s
Analog to digital converter	2-channels, 100MS/s, 14-bit
ADC SFDR	88dB

Table 2.2: National Instruments USRP 2921 receiver specifications

Software:

After the hardware is connected to the computer, software is setup for programming the hardware. Majority of National Instruments hardware use LabVIEW software as programming tools and USRP devices are not that different either. The list of software packages required for programming testbed is follows:

- Windows 8.1 64-bit operating system
- LabVIEW 2013 32-bit function block programming language
- U-Center ver 1.9.0 for configuring U-Blox LEA-6H GPS unit shown in Figure 2.3

The USRP is programmed using lab generated software package for spectrum sensing and opportunistic opportunistic spectrum access.

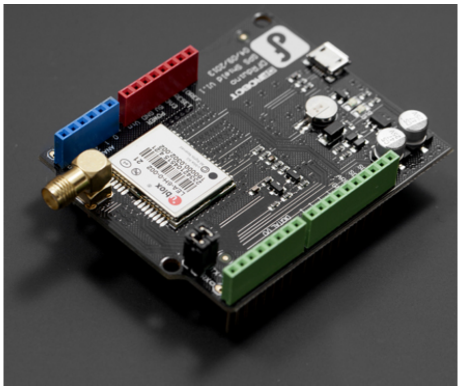


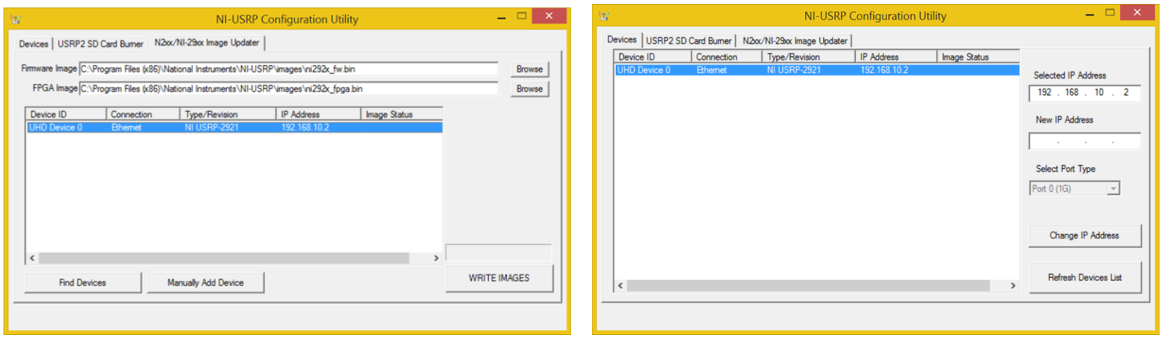
Figure 2.3: U-Blox LEA-6H GPS unit for coordinate acquistion

Configuration:

Next, the hardware and software are configured individually for optimum performance. Hardware configuration does not require tampering with the system. It only consisted of connecting cables, particularly a power source cable and category-5e data cable for communication link between computer and the USRP. GPS unit is connected to the computer using micro-USB cable. There are USRPs that have built-in GPS units in them. However for this research prototype, USRPs that do not have GPS are used. The separate GPS unit gets power from the micro-USB cable. It has a dedicated antenna connected to it.

The GPS unit does not require any configuration. The reliability and accuracy of the GPS unit is tested in Chapter 3. However, for the USRP, software configuration has to be changed which required changing the network configuration, firmware updates and programming the sample transmitter and receiver software.

In Figure 2.4, the USRP IP addressing is set to 192.168.10.2 and firmware version of the device is updated. The GPS firmware is also updated to enable the sensitivity of the GPS units. After successfully configuring the hardware and software, testing the system prior to programming is of utmost importance.



Firmware Image C:\Program Files (x86)\National Instruments\NI-USRP\images\ni292x_fw.bin

FPGA Image C:\Program Files (x86)\National Instruments\NI-USRP\images\ni292x_fpga.bin

Device ID	Connection	Type/Revision	IP Address	Image Status
UHD Device 0	Ethernet	NI USRP-2921	192.168.10.2	

Device ID	Connection	Type/Revision	IP Address	Image Status
UHD Device 0	Ethernet	NI USRP-2921	192.168.10.2	

Selected IP Address 192 . 168 . 10 . 2

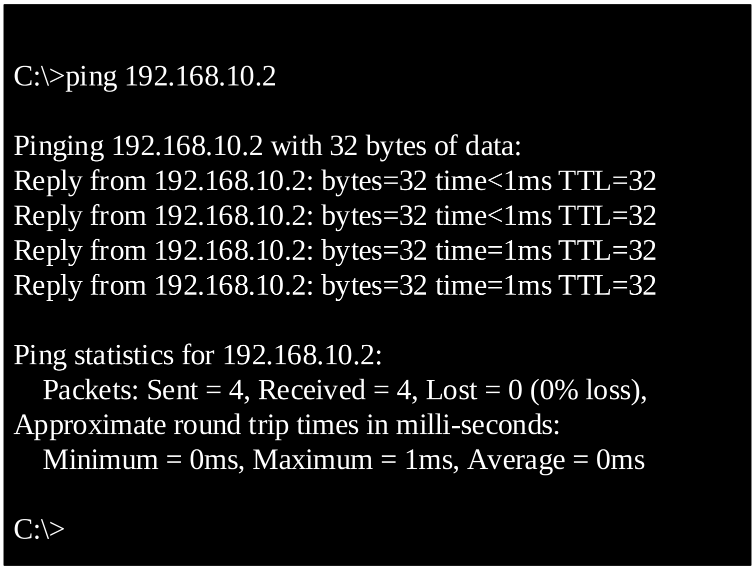
New IP Address . . .

Select Port Type Port 0 (1G)

Figure 2.4: Firmware and network configuration of USRP

Tests:

All the hardware and software utilities are configured. To test for communication between software and hardware, ping request packets are sent from the computer to query and check the link between devices.



```
C:\>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time<1ms TTL=32
Reply from 192.168.10.2: bytes=32 time<1ms TTL=32
Reply from 192.168.10.2: bytes=32 time=1ms TTL=32
Reply from 192.168.10.2: bytes=32 time=1ms TTL=32

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

Figure 2.5: Ping request to verify the hardware is configured and communicating

I/Q Data Test:

Prior to designing ROAR architecture testbed, it is crucial to verify if two USRPs are able to communicate with each other or not. Hence, the I/Q testing is carried out for this purpose. For the hardware designed, National Instruments recommends I and Q data for sine and cosine waveforms to be modulated [32]. Thus I/Q data is modulated at the transmitter and receiver ends that can be successfully handled by the USRPs. The transmitter is configured to transmit I/Q data and the receiver-end is configured to receive the I/Q data as shown in Figure 2.6.

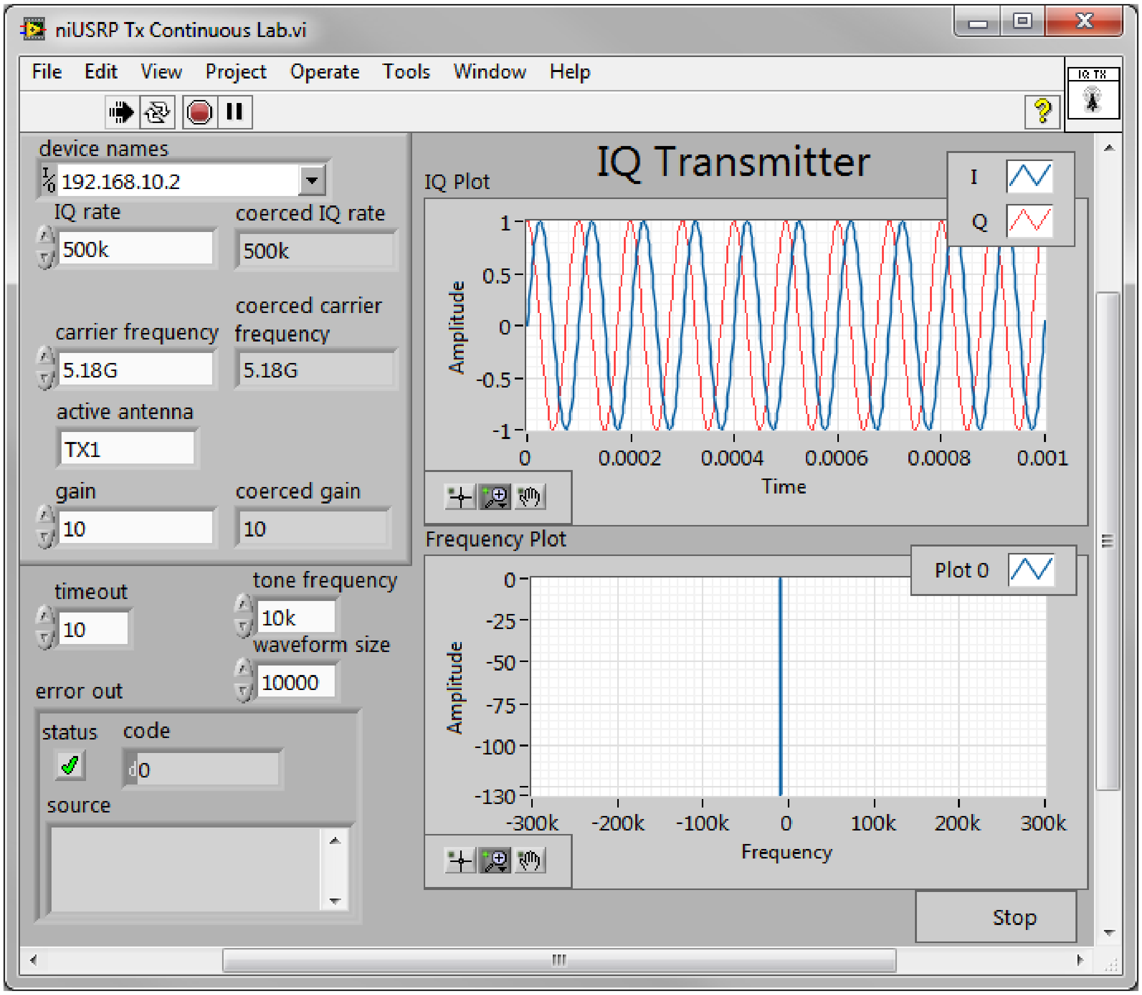
The transmitter is set at IP 192.168.10.2 with 500k I/Q data rate modulated with the carrier frequency at 5.18GHz from the antenna TX1 with 10dB gain. The receiver is set to tune the frequency 5.18GHz and is expecting to receive 500k I/Q sampled data-rate. The test interface for receiver is shown in Figure 2.7.

From Figure 2.7, it can be seen that the received signal is identical to the transmitted signal as shown in Figure 2.6. The gain value at the receiver VERT-2450 antenna is set at 10dB for better reception and to minimize the received noise. Because of the antenna gain and due to adequate performance of the functional block programming, the received I/Q has lower amplitude than the transmitted one. Since the modulation scheme in this test is amplitude modulation, the resulting factor is affected by any variation in the environment due to the amplitude modulation susceptibility to slightest of variation in the environment.

Energy Detection Test:

The graph shown in Figure 2.7 is the result of signal strength measured at the receiver. In the absence of transmitter the received I/Q data and its frequency plot appears as shown in the Figure 2.8.

However, in presence of a transmitter the received signal strength can be measured as



device names: 192.168.10.2

IQ rate: 500k

coerced IQ rate: 500k

carrier frequency: 5.18G

coerced carrier frequency: 5.18G

active antenna: TX1

gain: 10

coerced gain: 10

timeout: 10

tone frequency: 10k

waveform size: 10000

error out

status: ✓

code: 0

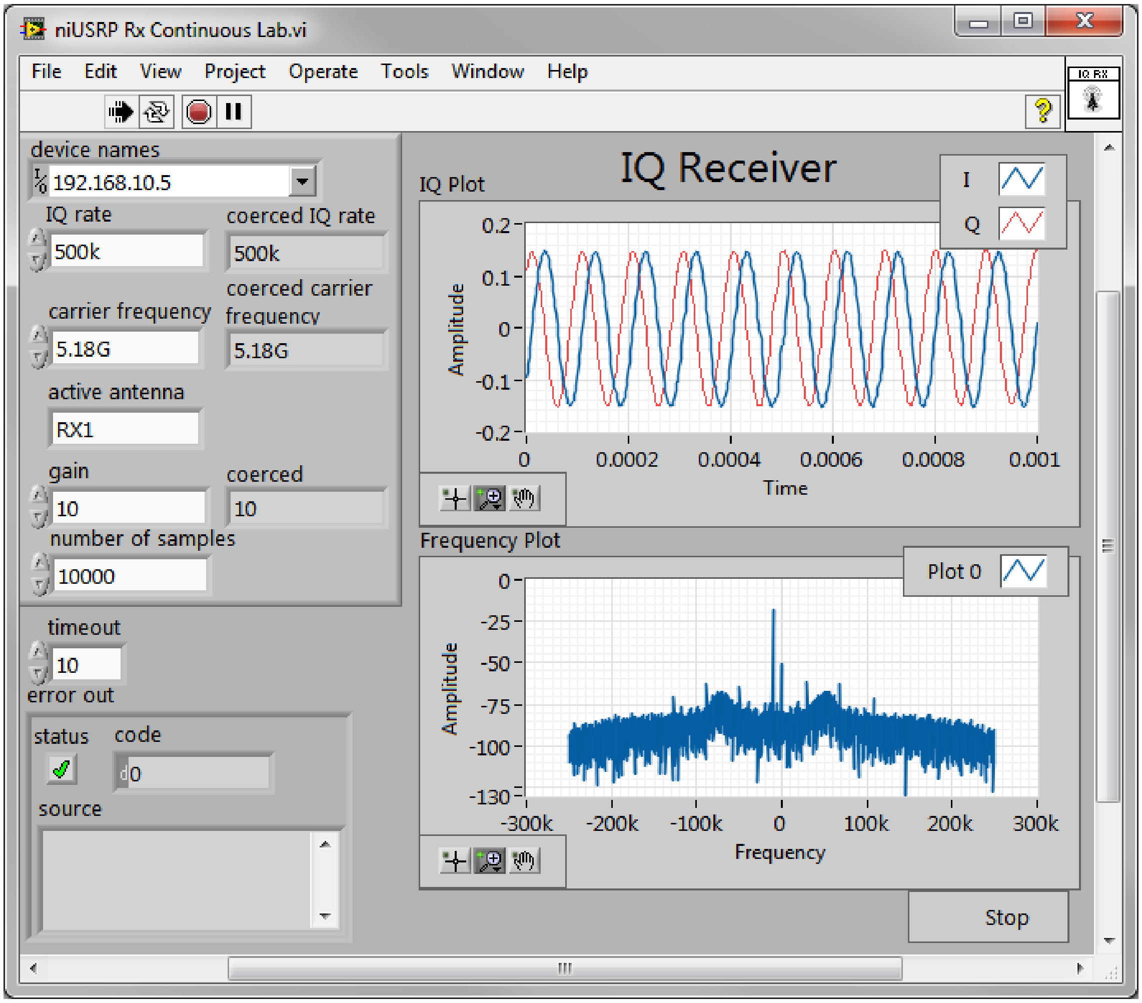
source

Figure 2.6: USRP I/Q testing at transmitter end

shown by the graph in Figure 2.10.

The energy detection implemented in the functional block diagram shown in Figure 3.2 is determining whether there is a spectrum occupancy or vacancy. The implementation of PSD as a function block diagram for energy detection is shown in Figure 2.9 and the detail reference to the use of PSD can also be speculated from the Figure 2.9. The spectrum analyzed from the PSD function block diagram is plotted as frequency plot and the result of frequency plot in the presence of transmitter is provided in Figure 2.10.

The sensors that are reporting to the database use similar (PSD based) signal strength



device names: 192.168.10.5

IQ rate: 500k

coerced IQ rate: 500k

carrier frequency: 5.18G

coerced carrier frequency: 5.18G

active antenna: RX1

gain: 10

coerced: 10

number of samples: 10000

timeout: 10

error out

status: ✓

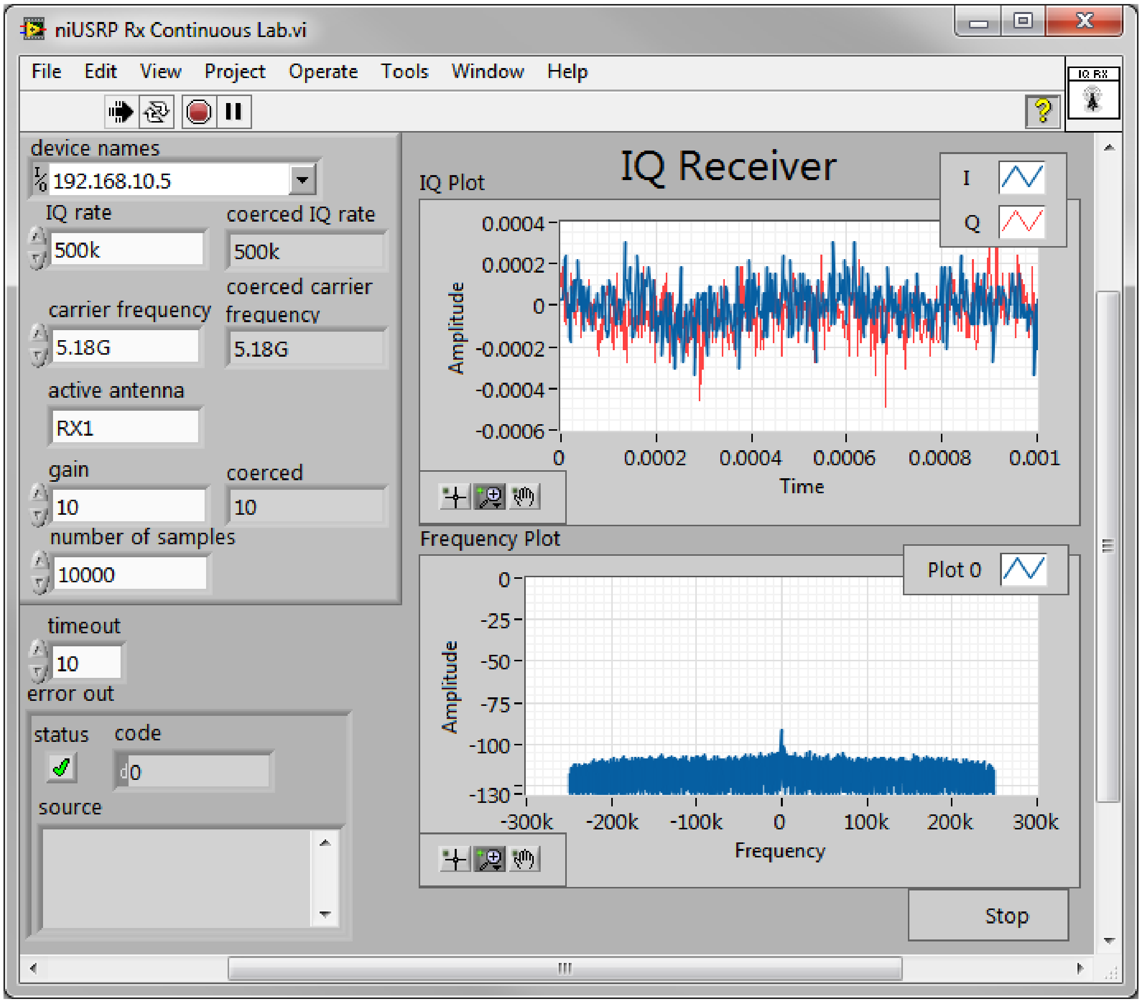
code: 0

source

Figure 2.7: USRP I/Q testing at receiver end

sensing method to search for idle spectrum. The idle spectrum information is used by SUs to communicate with each other during opportunistic spectrum access. The research is based on ISM band as operating in ISM band is not restricted by FCC and is not in violation of any FCC spectrum operating policies [4].

Prior to continuing with development of ROAR architecture, it is also crucial to measure any noise in the environment. Therefore, to validate the I/Q data test, the transmitter is set at the configuration as shown in Figure 2.6 and Agilent MXA N9020A signal analyzer is used to sense the signal strength and analyze at corresponding frequency to verify



device names
192.168.10.5

IQ rate
500k

coerced IQ rate
500k

carrier frequency
5.18G

coerced carrier frequency
5.18G

active antenna
RX1

gain
10

coerced
10

number of samples
10000

timeout
10

error out

status
✓

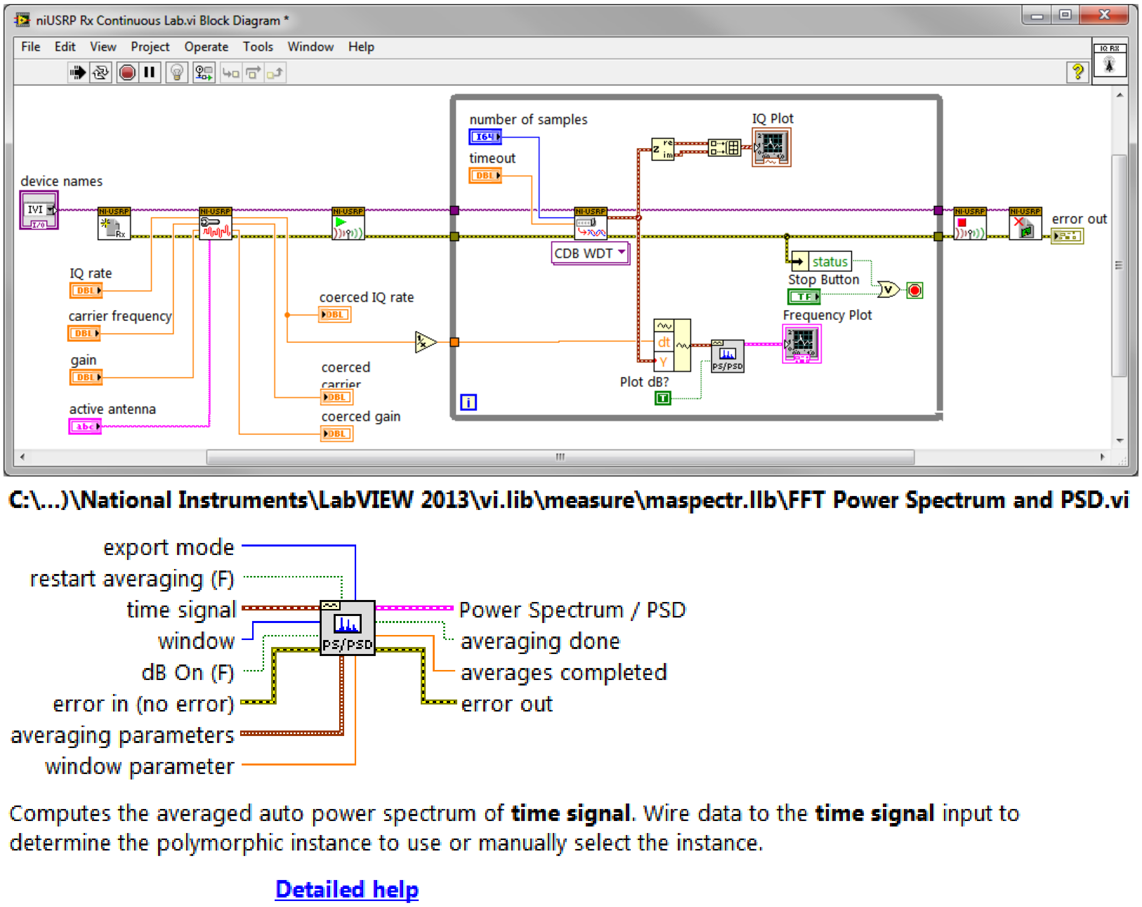
code
0

source

Figure 2.8: USRP I/Q testing at receiver end without transmitter

if the USRP receiver is indeed picking the right transmitter or if it is picking the noise in the environment. Thus, before ROAR architecture is developed, one more test is carried to validate the USRP reliability by introducing Agilent MXA N9020A signal analyzer as shown in Figure 2.11.

The spectrum sensing for ROAR architecture is designed and implemented using USRPs. The detail is presented in Chapter 3.



C:\...)\National Instruments\LabVIEW 2013\vi.lib\measure\maspectr.llb\FFT Power Spectrum and PSD.vi

Computes the averaged auto power spectrum of **time signal**. Wire data to the **time signal** input to determine the polymorphic instance to use or manually select the instance.

Detailed help

Figure 2.9: Power Spectrum / Power Spectral Density (PSD) function block diagram implementation and detail

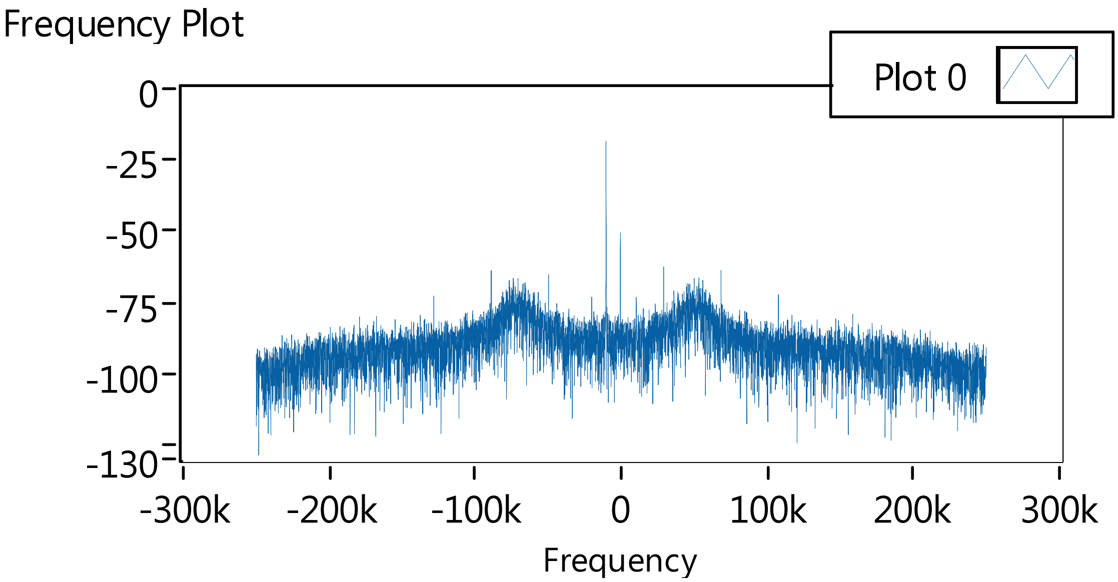
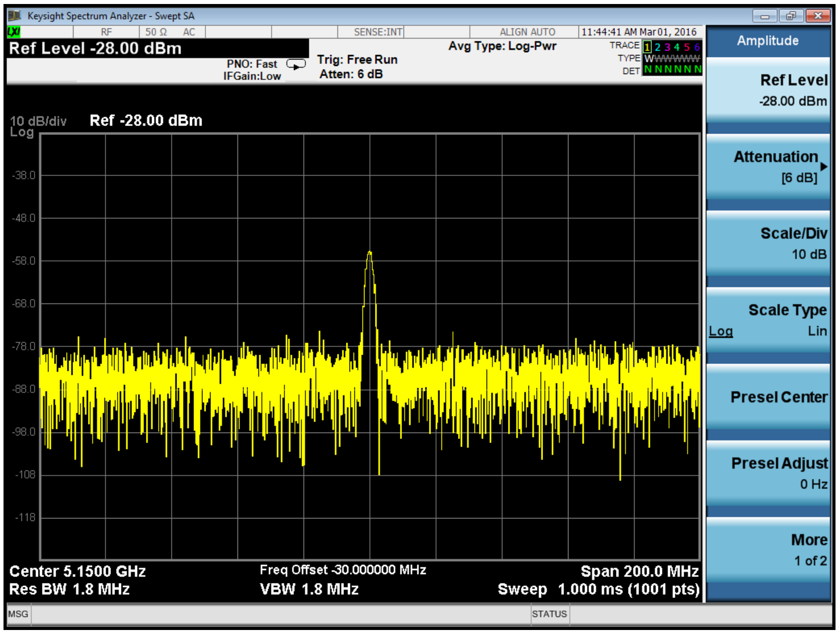


Figure 2.10: USRP I/Q received frequency plot in the presence of transmitter


Keysight Spectrum Analyzer - Swept SA

RF 50 Ω AC SENSE:INT ALIGN AUTO 11:44:41 AM Mar 01, 2016

Ref Level -28.00 dBm

PNO: Fast
IFGain:Low

Trig: Free Run
Atten: 6 dB

Avg Type: Log-Pwr

TRACE 1 2 3 4 5 6
TYPE WWWWWW
DET NNNNNN

10 dB/div Ref -28.00 dBm
Log

Center 5.1500 GHz
Res BW 1.8 MHz

Freq Offset -30.000000 MHz
VBW 1.8 MHz

Span 200.0 MHz
Sweep 1.000 ms (1001 pts)

Figure 2.11: Agilent MXA N9020A signal analyzer reading of transmitter USRP to verify receiver USRP's sensing ability

CHAPTER 3

SPECTRUM SENSING FOR ROAR ARCHITECTURE

In this chapter, spectrum sensing methods are studied and the most suited spectrum sensing technique is presented for ROAR architecture. The spectrum sensing program for USRP is developed using LabVIEW and evaluated for ROAR architecture testbed.

3.1 Spectrum Sensing: A Background

The spectrum sensing for ROAR architecture is the first step towards identifying the availability of idle spectrums prior to reporting them to the spectrum repository. In order to sense the spectrum, the sensors for ROAR architecture use USRP receivers. The receivers are programmed to continuously look for signals. In the Figure 3.1, a signal is considered in frequency domain that has a carrier frequency f_c and the bandwidth B.

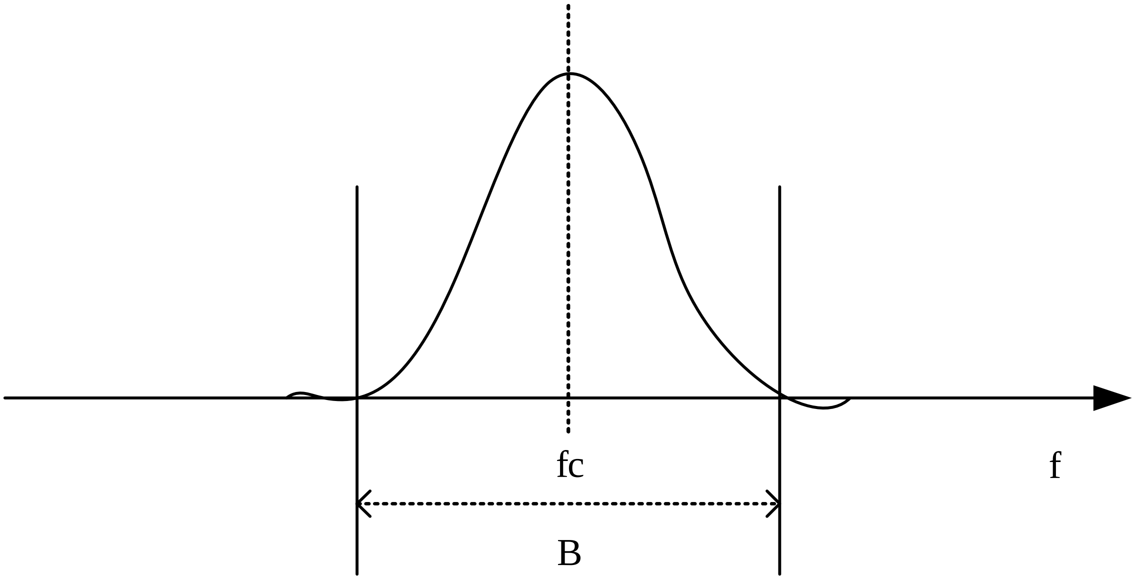


Figure 3.1: Signal with carrier frequency f_c and bandwidth B

If the continuous signal in time domain $y(t) = w(t) + x(t)$ with Gaussian noise $w(t)$

and desired signal $x(t)$ is sampled with sampling frequency f_s such that $f_s > B$, the sampled signal for the presence of any spectrum can be hypothesized as shown in equation (3.1) [33] [34]:

$$\begin{aligned} \mathscr{H}_0 : &\qquad Y[n] = W[n] &\qquad \text{idle spectrum} \\ \mathscr{H}_1 : &\qquad Y[n] = W[n] + X[n] &\qquad \text{spectrum occupied} \end{aligned} \tag{3.1}$$

where, $n = \{1, 2, 3, \ldots, N\}$ is the observation interval.

Here in the equation (3.1), $Y[n]$ is the discrete signal, $W[n]$ is the Additive White Gaussian Noise (AWGN). When there is no desired signal $X[n]$ present in the discrete signal, the noise $W[n]$ is prevalent. But when the signal is present, the overall signal is the combination of noise and the desired signal. In an idle environment, the signal $X[n] = 0$ signifies the presence of idle spectrum. However, whenever $X[n] \neq 0$, given that the readings collected is error free, the idle channel is not available.

In order to detect any signal and to sense the spectrum in cognitive radio, several methods of spectrum sensing are proposed in [33]. The survey on spectrum sensing categorized spectrum sensing techniques into primary transmitter detection, primary receiver detection, cooperative detection, interference temperature management and other miscellaneous detection techniques such as wavelet based detection, multi-taper spectrum sensing and estimation, filter band based spectrum sensing, etc. Among several signal detection techniques, primary transmitter detection is of relevant concern for the ROAR architecture since the information being stored in spectrum repository is related to the idle channel information.

In primary transmitter detection, there are again several methods to detect a signal. Among all the proposed detection methods, energy detection method is chosen for ROAR architecture due to its low computational and implementation complexity and high compatiability with USRP devices [33]. With high complexity algorithms in various other signal detection techniques, there are possibilities of delayed spectrum sensing for ROAR architecture. With each delay in reporting the availability of spectrum in spectrum repository, the ROAR architecture might not meet real-time opportunistic spectrum access and spectrum reporting criteria. Therefore, the advantage of energy based detection over matched filtering based detection, covariance based detection, waveform based detection etc. is that, in energy based detection, prior knowledge of any PUs or SUs presence in that particular spectrum is not needed and that the energy based detection gives faster accurate readings. Energy detection in any spectrum is the most simple form of detection where the energy of any signal at f_c is compared with the threshold value (threshold value depends on the noise floor) conceived in an idle spectrum. The energy detection is calculated based on the equation (3.1) as:

$$E_n = \sum_{n=0}^{N} |Y[n]|^2 \underset{\mathscr{H}_1}{\overset{\mathscr{H}_0}{\lessgtr}} \gamma_{th} \tag{3.2}$$

The energy detected spectrum can then be analyzed for signal present or absent conditions. In energy detection for ROAR architecture, it is inevitable that the energy detected at any given spectrum might falsely detect a signal due probability of false alarm p_{fa} and might miss-detect a signal due to probability of missed detection p_{md}. In p_{fa}, the spectrum sensor detects legitimate value and falsely reports to the system that there is a signal present and the spectrum is being used. In p_{md}, the spectrum sensor does not give any value even if the spectrum is being used and there is a signal present.

In order to prevent the ROAR architecture from receiving any wrong information from the spectrum sensing, a decision threshold γ_{th} is defined for any noise variance $\hat{\sigma}_n^2$ in the spectrum. By optimizing γ_{th}, p_{fa} and p_{md} can be minimized. For lower p_{fa} and p_{md}, the E_n reading has less error. p_{fa} and p_{md} can be expressed as [27]:

$$\begin{aligned} p_{fa} &= Pr(E_n > \gamma_{th}|\mathscr{H}_0) \\ p_{md} &= Pr(E_n < \gamma_{th}|\mathscr{H}_1) \end{aligned} \tag{3.3}$$

The decision threshold, γ_{th} can be defined based on $\hat{\sigma}_n^2$ and p_{fa} under Q-function for N number of samples collected as shown in equation (3.4) [35].

$$\gamma_{th} = \hat{\sigma}_n^2 \cdot Q^{-1}(p_{fa}) \cdot (\sqrt{2N} + N) \tag{3.4}$$

The threshold γ_{th} is then expanded based on the energy readings to sense if there is a legitimate signal present on any given spectrum or not. By categorizing the sensed spectrum under “faint” (γ_{LO}), “medium” or “high” (γ_{HI}) energy levels of signal, the decision threshold can be made adaptive to sense occupied spectrum [35].

From the equation (3.4), it can be observed that if there is low noise in the system $\hat{\sigma}_n^2$, then γ_{th} is less. For, $\hat{\sigma}_n^2 << 0$, $\gamma_{th} = \gamma_{LO}$. For the energy detected in the system, if $E_n \leq \gamma_{LO}$, then there is no PU signal present at any particular spectrum. Thus, the spectrum can be reported as idle for ROAR architecture.

Similarly, if there is high noise in the environment, $\hat{\sigma}_n^2 >> 0$, $\gamma_{th} = \gamma_{HI}$. For $E_n \geq \gamma_{HI}$, then there is a PU signal in that spectrum along with the noise. In this case, the spectrum is being occupied and is not reported to the distributed cloud based database for ROAR architecture.

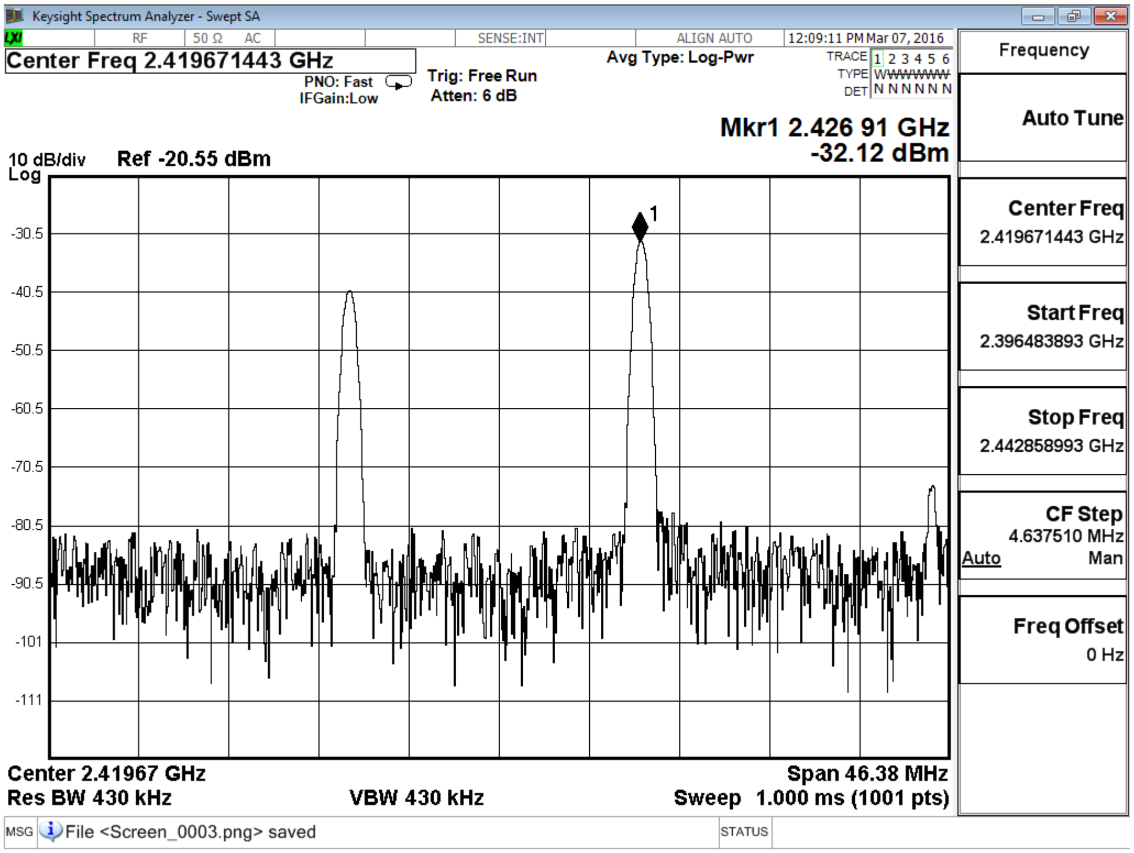
The decision is then made based on the condition if $E_n \leq \gamma_{th}$ or $E_n > \gamma_{th}$. Thus, based on the decision threshold, PU signal present or absent condition can be derived as:

$$\begin{aligned} &E_n \leq \gamma_{LO}, && \text{PU signal not present} \\ &E_n \in (\gamma_{LO}, \gamma_{HI}), && \text{PU signal might be present} \\ &E_n \geq \gamma_{HI}, && \text{PU signal is present} \end{aligned} \tag{3.5}$$

For $E_n \in (\gamma_{LO}, \gamma_{HI})$, there is a possibility that the PU signal might be present or not.

For such cases, adaptive decision threshold is implemented by changing the number of samples N. With higher number of samples, the resolution of the reading is improved and decision threshold is altered as shown in equation (3.4).

Based on the energy detection, the spectrum sensing is speculated for two PUs, PU_A and PU_B administered in the spectrum environment. It can be seen from the Figure 3.2, that for each PU, a signal is perceived. Depending on the decision threshold, if the threshold value is set at -40.5dB, only PU_A (denoted as 1) will be considered as spectrum occupied. Therefore, based on the decision threshold, the energy detection is able to decide whether a spectrum is occupied or is vacant.



Keysight Spectrum Analyzer - Swept SA

RF | 50 Ω AC | SENSE:INT | ALIGN AUTO | 12:09:11 PM Mar 07, 2016

Center Freq 2.419671443 GHz

PNO: Fast IFGain:Low

Trig: Free Run
Atten: 6 dB

Avg Type: Log-Pwr

TRACE 1 2 3 4 5 6
TYPE WWWWWW
DET N N N N N N

Mkr1 2.426 91 GHz
-32.12 dBm

10 dB/div Log Ref -20.55 dBm

Center 2.41967 GHz
Res BW 430 kHz
VBW 430 kHz
Span 46.38 MHz
Sweep 1.000 ms (1001 pts)

Center Freq 2.419671443 GHz

Start Freq 2.396483893 GHz

Stop Freq 2.442858993 GHz

CF Step 4.637510 MHz Auto Man

Freq Offset 0 Hz

MSG File <Screen_0003.png> saved

Figure 3.2: Energy detection of PU_A and PU_B

After sensing the spectrum, the acquired idle spectrums are then encapsulated with

GPS-location information, along with time-stamps and are stored in cloud-based geolocation database for ROAR architecture. The spectrum sensing for ROAR architecture is implemented and discussed in the next section.

3.2 Sensor Implementation

The spectrum sensing model for ROAR architecture is shown in the Figure 3.3. The spectrum sensing architecture is divided into different components and their functionality can be explained individually. The interface developed for the sensor is shown in Appendix A.4 where GPS, spectrum sensor and database components are clearly visible. The explanation of each component is presented as follows:

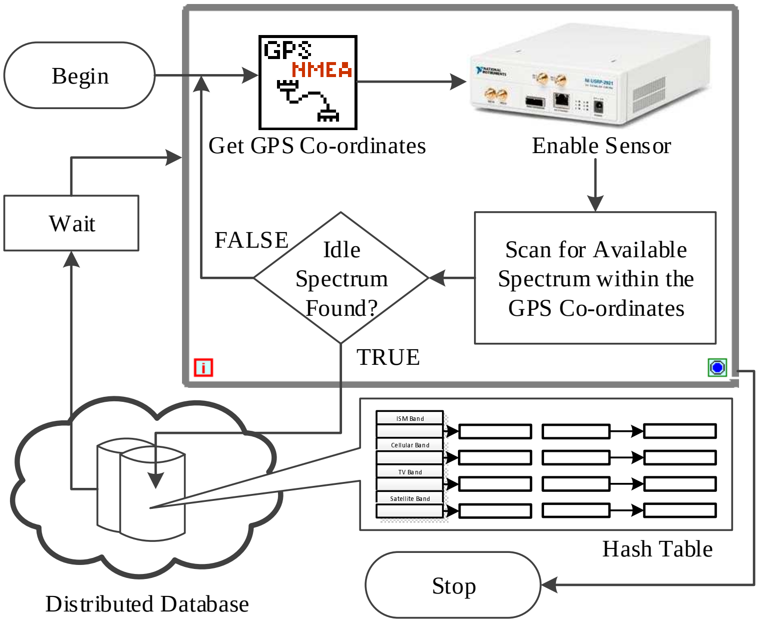


Figure 3.3: Sensor sub-system model consisting of individual components

The spectrum sensing for ROAR architecture in Figure 3.3 shows that the sensing process initiates at GPS and ends after reporting the acquired spectrum occupancy information to the distributed database. Once the GPS location is retrieved, the sensors are enabled and begin searching for available spectrums. If the sensors detect idle spectrums, the spectrums are reported to the database where the information is stored in a hash table. Else, the sensor keeps looking for idle spectrum in a loop. The process repeats until it is terminated by loop counter.

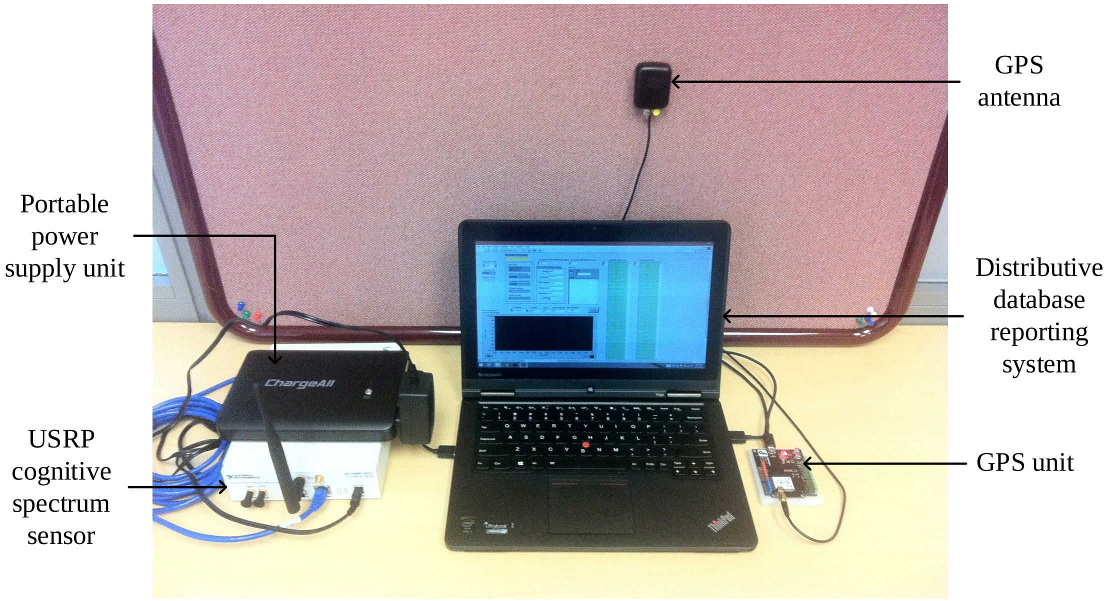

GPS antenna
Portable power supply unit
Distributive database reporting system
USRP cognitive spectrum sensor
GPS unit

Figure 3.4: Spectrum sensing testbed for ROAR architecture

The sensor testbed for ROAR architecture is shown in Figure 3.4. The unit consists of USRP, GPS unit with GPS antenna, remotely located cloud-based database and power supply unit. The USRP being used for the system design does not have its own GPS device built into it. Thus, a separate external GPS unit is used. The sensor reports to distributed database.

GPS in Spectrum Sensing for ROAR Architecture:

The flowchart in Figure 3.5 shows the logical diagram of the GPS unit.

The GPS unit in spectrum sensing for ROAR architecture is programmed to latch to any GPS satellites and triangulate the position of the sensors being deployed. A time-

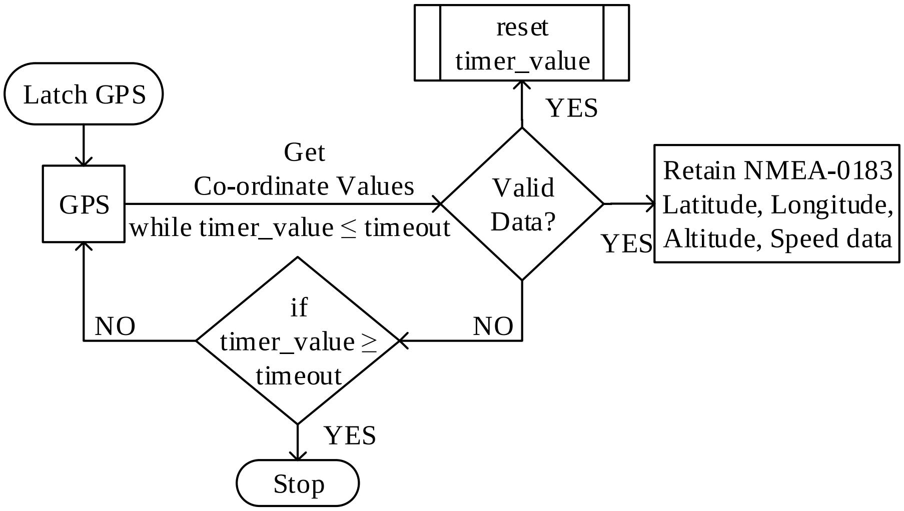


Figure 3.5: Functionality of GPS unit in spectrum sensing for ROAR architecture explained through flowchart

out value is assigned to the GPS as shown in Figure 3.5 which terminates the sensor and initializes the whole process if no satellites are found within certain time frame. The coordinates retrieved from the GPS are used to encapsulate the idle spectrum information prior to reporting to the geolocation database for ROAR architecture. The process continues and repeats as long as valid National Marine Electronics Association (NMEA) 0183 standard data is present. The GPS reports latitude, longitude, altitude and speed along with UTC timestamp. The timeout value is reset for all valid data. The flowchart logic is implemented in LabVIEW program as shown in the Figure in Appendix A.1. The block diagram is shown in the Figure 3.6.

In the Figure 3.6, the Block 1 shows the initialization of the GPS USB V-Port, defining the timeout value and an implementation of timer. Once these values are defined, they are passed to NMEA-0183 coordinate acquisition program to retrieve GPS coordinates. The Block 2 is a conditional block where the NMEA-0183 data is extracted only if the satellite data is valid. The validity of GPS data is checked using the data formatting tool built into

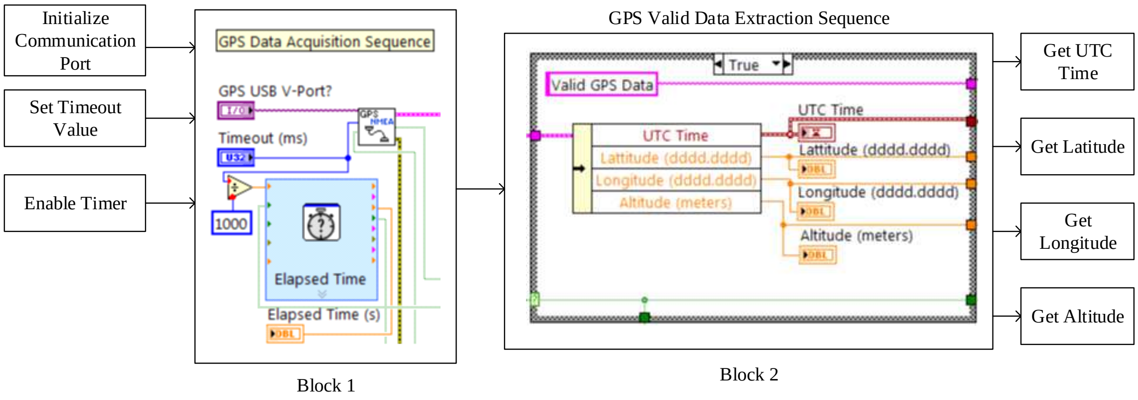


Figure 3.6: Block diagram representation of GPS LabVIEW program

the program. After the valid data is confirmed as shown in Block 2, longitude, latitude, altitude and UTC time are formatted. The formatted data is easy to store in the geolocation database.

Spectrum Sensing:

Spectrum sensing is implemented by programming the USRP in LabVIEW. The LabVIEW function block programming as shown in Figure in Appendix A.2 uses energy detection method to sense the spectrum. The sensing process is only activated after the GPS unit gets valid geolocation data. The whole program is created based on the sequential programming method. Therefore, only after successfully acquiring valid GPS data, the sensor begins sensing for spectrum.

The Figure 3.7 shows the block diagram representation of spectrum sensor for ROAR architecture (the full LabVIEW diagram is shown in Appendix A.2). Here, communication port is first initialized where the IP address is set along with the IQ rate, antenna gain and the sensor channel where the physical antenna is attached. The number of samples is used for resolution of sensed spectrum in order to achieve better spectrum sensing. After these parameters are set, the spectrum sensor for ROAR architecture is enabled and is tran-

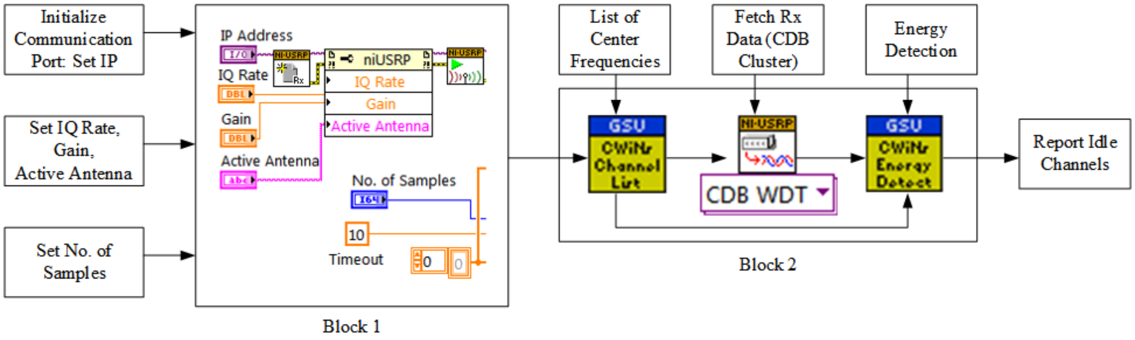


Figure 3.7: Block diagram representation of sensor LabVIEW program

sitioned from Block 1 to Block 2. In Block 2, the CWiNs Channel List provides the list of center frequencies including the wideband (50MHz to 6GHz) which is passed onto the CDB cluster. The CDB cluster is responsible for fetching the received signals in complex, double-precision floating-point data [30]. The CDB cluster reads the data inside the USRP and by using CWiNs Energy Detect sub-program is able to list all those spectrums that are idle. The energy detection sub-program uses γ_{th} set at -88dB. Any signal present in the environment that have lower energy than the threshold are considered to be available, i.e. $E_n \leq \gamma_{LO}$ from equation (3.5). The available channels are acquired into an array and the array is then passed onto cloud-based distributed database to store all the idle channels. Like this way the spectrum sensing is implemented for ROAR architecture. The implemented framework is evaluated in the next section.

3.3 Evaluation of the Spectrum Sensor for ROAR Architecture

The designed and implemented spectrum sensing for ROAR architecture provides idle channel information. The sensor utilizes GPS to identify the areas where the spectrum availability information are sensed. The sensed information is then reported to the cloud-based distributed database. The sensor and its readings are evaluated as follows:

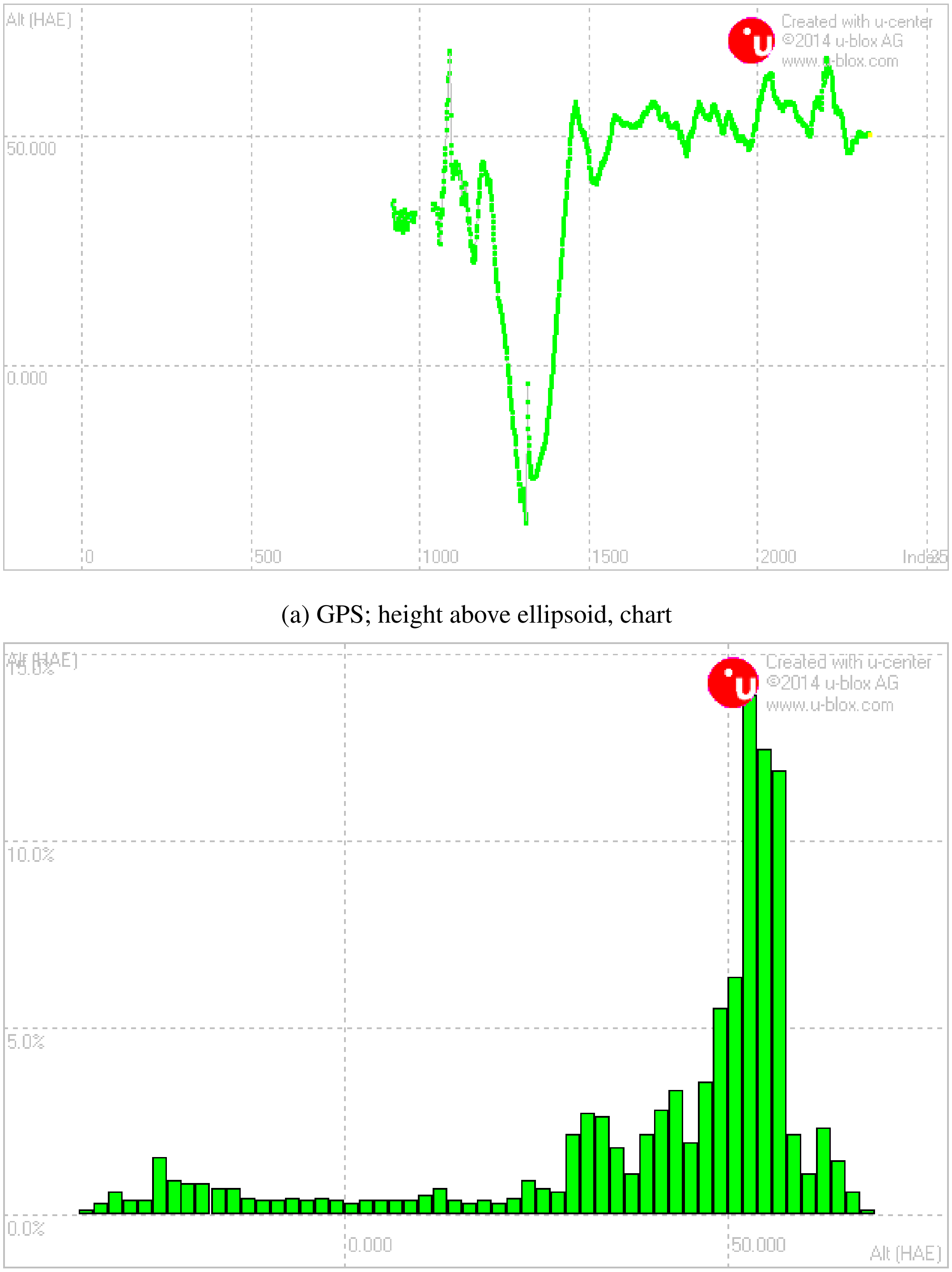
The coordinate acquired by the GPS unit is used in spectrum sensing ROAR testbed. It is necessary to verify the accuracy of the retrieved geolocation data by comparing the NMEA-0183 standard being used in the programming of the sensor in LabVIEW with the manufacturer provided software utility for U-Blox LEA-6H GPS. The difference in reading verifies the accuracy of the programmed sensors in LabVIEW based function block diagram if it is exactly the same as the manufacturer provided location measuring utility.

The outputs from U-Blox utility are shown in the Figures 3.8 and 3.9. The results obtained show all the information required to use the GPS unit to program the spectrum sensor. By speculating the results, it is seen that the GPS unit is precise to be used for the testbed intended for the research.

In the Figure 3.8, the satellite position is triangulated to verify if the GPS unit is properly functioning. The GPS units use ellipsoid to measure the vertical and horizontal positions of the unit on the Earth's surface. The ellipsoid is the approximated elevation based on the sea-level [36]. The measured data is plotted using U-Blox software known as U-Center. The two graphs in Figure 3.8(a) and (b) show the ellipsoidal height in chart and histogram representation which shows the average altitude (height above ellipsoid (HAE)) measurement and the accuracy of the GPS unit being used in testbed for ROAR architecture. Based on these readings, it can be evaluated that the %HAE fluctuation is stable and the accurate reading can be asserted.

In the Figure 3.9, the deviation map and the position of each connected satellite gives the precision of the device. More satellite connection means that the coordinates are highly accurate. Plus, the deviation map reveals the skewed coordinate values that are shifted from the average deviation to determine the error in readings.

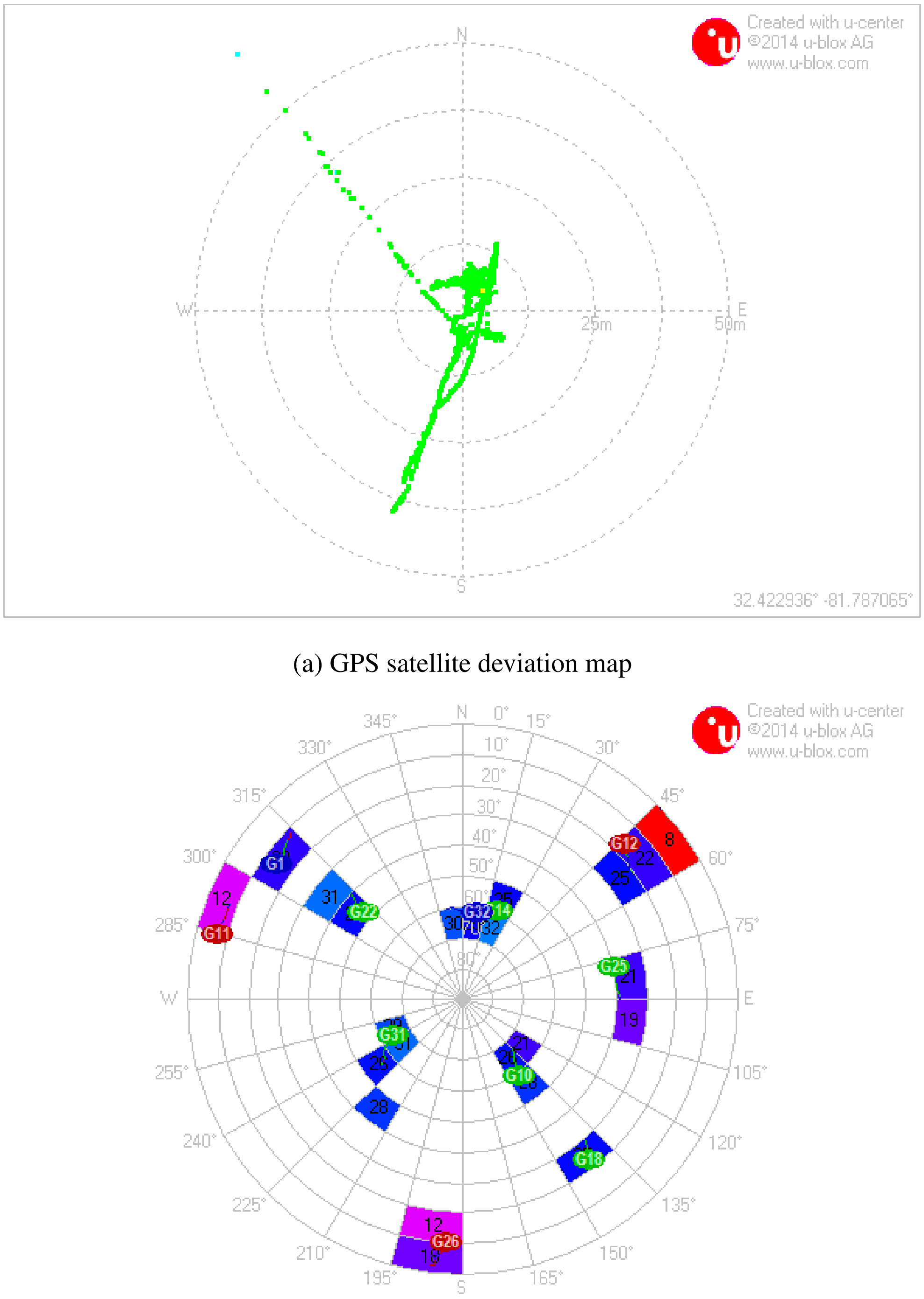
Similarly, in the Figure 3.10, the U-Center software selects the satellites and measures their signal strength to choose for the best ones for stable connection. From the measurement, it is seen that the satellites G31, G18, G10, G25, G14 and G22 from the U.S., has



(a) GPS; height above ellipsoid, chart

(b) GPS; height above ellipsoid, histogram

Figure 3.8: GPS height measurement with reference to ellipsoid



(a) GPS satellite deviation map

(b) GPS satellite sky view

Figure 3.9: GPS satellite position mapping and deviation

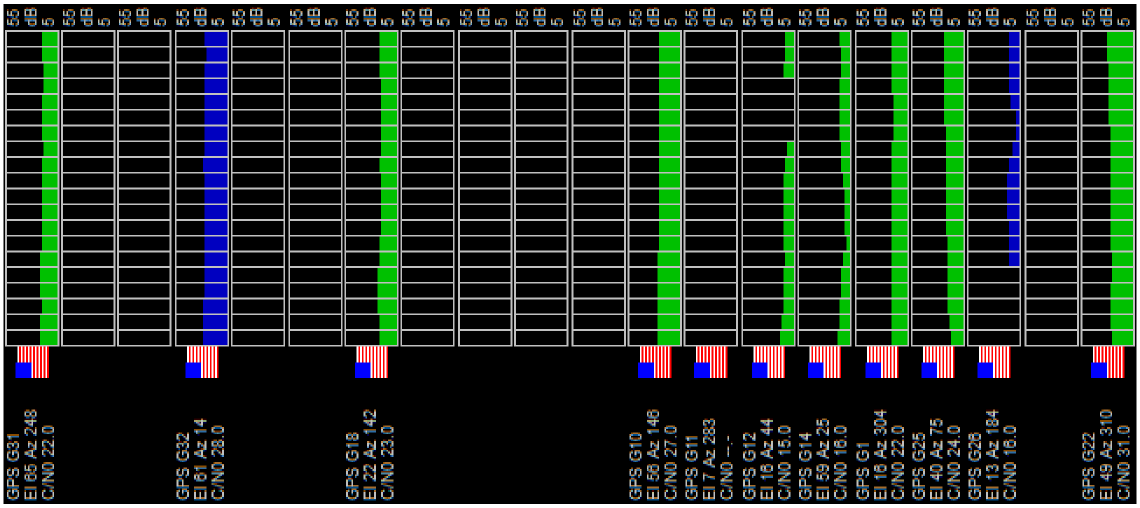
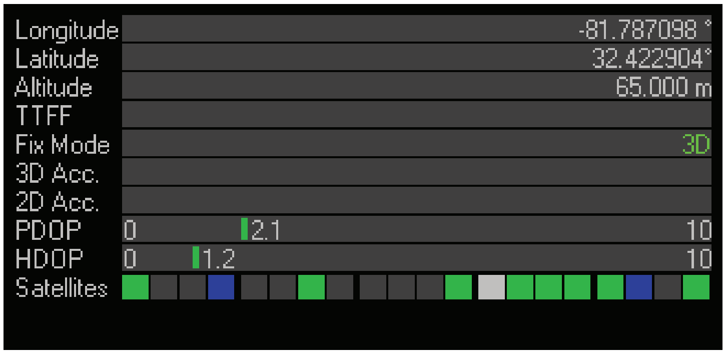


Figure 3.10: Available GPS satellites with signal strength

stable connections and are latched to fixed 3D measurements of the position. The received signal strength is recorded to be roughly 5dBm with very stable signal throughput. Multiple connections to different satellites provide stability as shown in Figure 3.10.

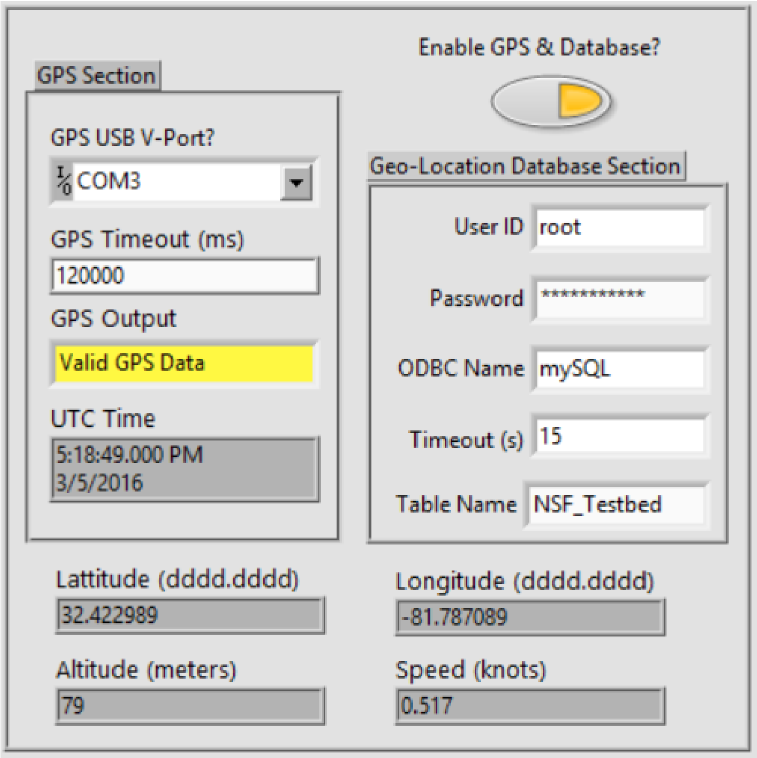
The coordinate of the testbed obtained from the U-center software measurement shown in Figure 3.11 is tallied with the results obtained from sensor program in LabVIEW shown in Figure 3.12.



Longitude	-81.787098 °
Latitude	32.422904°
Altitude	65.000 m
TTFF	
Fix Mode	3D
3D Acc.	
2D Acc.	
PDOP	0 2.1 10
HDOP	0 1.2 10
Satellites	

Figure 3.11: GPS longitude and latitude readings from GPS in U-Center

From the Figures 3.11 and 3.12, it is observed that the latitude values are almost the same. The software U-Center measured the location of testbed at 32.422904°latitude



Enable GPS & Database?

GPS Section

GPS USB V-Port?

COM3

GPS Timeout (ms)

120000

GPS Output

Valid GPS Data

UTC Time

5:18:49.000 PM
3/5/2016

Geo-Location Database Section

User ID	root
Password	***********
ODBC Name	mySQL
Timeout (s)	15
Table Name	NSF_Testbed

Lattitude (dddd.dddd)	Longitude (dddd.dddd)
32.422989	-81.787089
Altitude (meters)	Speed (knots)
79	0.517

Figure 3.12: GPS data acquisition in LabVIEW

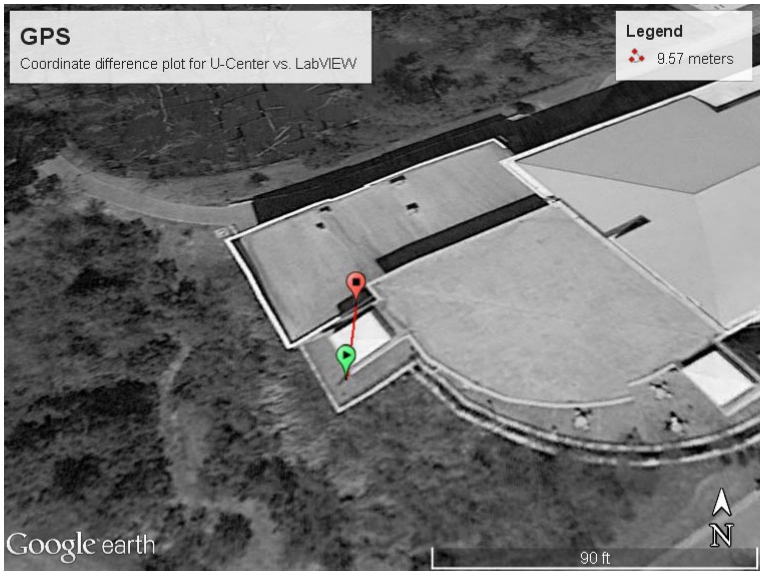


Figure 3.13: GPS coordinate difference between U-Center and LabVIEW Google Map plot

whereas the designed program for ROAR testbed in LabVIEW obtained 32.422989°. Similarly, the longitude difference in readings are -81.787098°and -81.787089°for both U-Center and LabVIEW. The obtained values are not different by larger magnitude, but from real-time Google maps plotting showed that the difference is 9.57 meters as shown in map coordinate plot in the Figure 3.13.

From the result it is verified that the GPS unit being used in spectrum sensing for the ROAR testbed is accurate and provides high resolution coordinate acquisition upto ±10 meters tolerance. The spectrum sensing USRP range is higher than 10m and adapts with antenna gain, therefore the discrepancy of ±10m does not affect the system significantly.

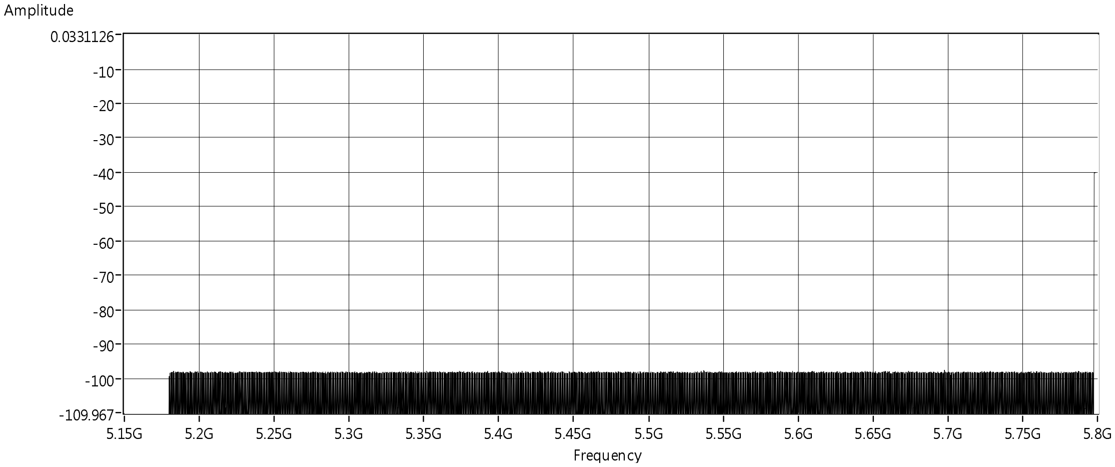


Figure 3.14: 5GHz spectrum in absence of noise and users

The GPS result for accuracy proved that the geolocation data acquired is reliable and valid. Only for valid entry of geolocation data, the sensor then initiates spectrum sensing. The sensed spectrum covers wideband (50MHz to 6GHz). Without any users or the noise in the spectrum, the entire spectrum appears silent as seen in the Figure 3.14.

However, when there are channels available, the ISM band appears as shown on the output from the interface designed on LabVIEW in Figure 3.15.

In the Figure 3.15, the carrier frequency is plotted based on the strength of the signal present in the environment. The decision threshold γ_{th} is assigned and a spectrum is

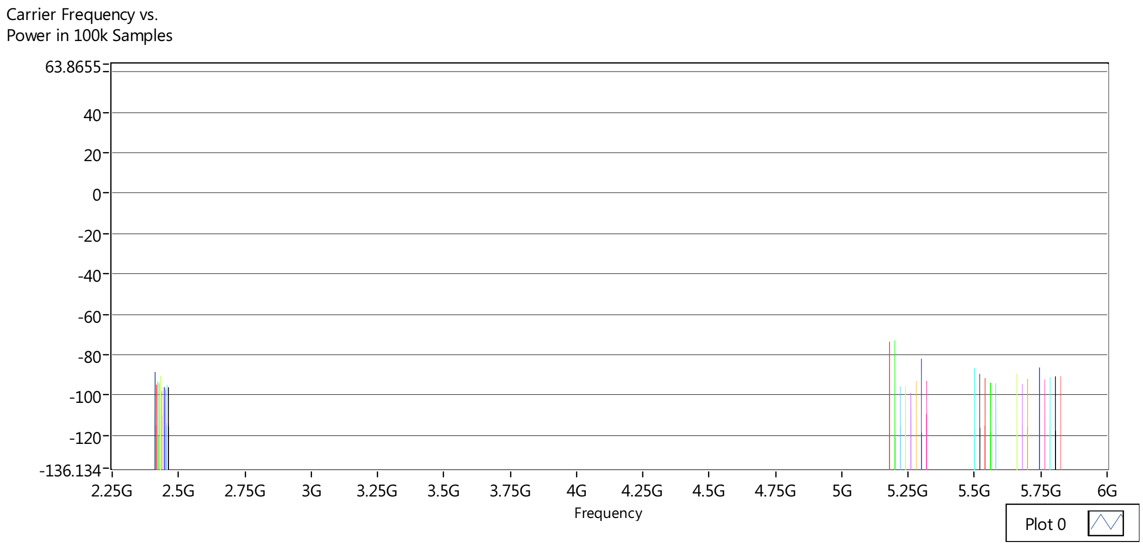
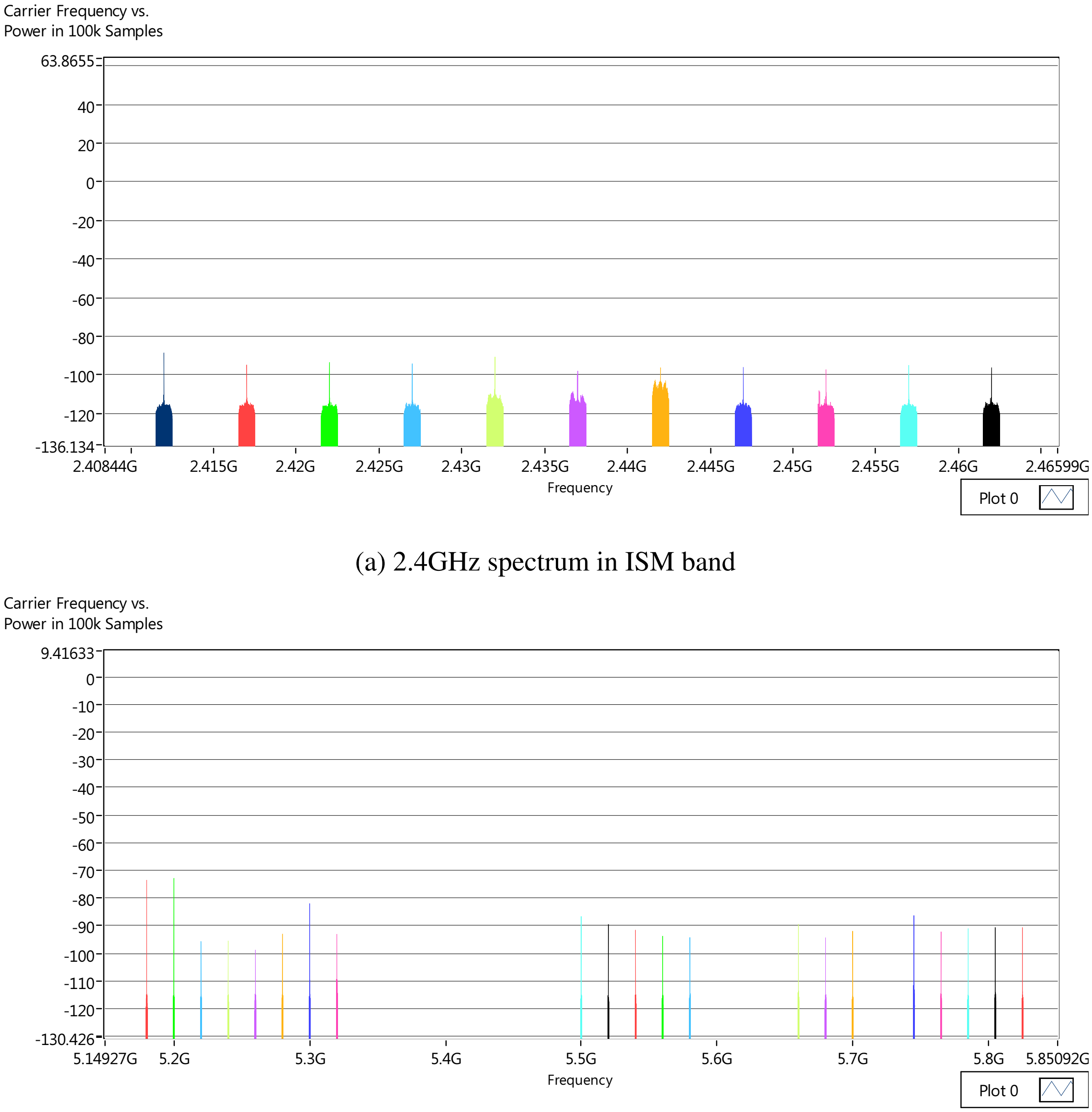


Figure 3.15: ISM band sweep based on the power readings

considered as occupied if there is high energy in that spectrum when compared to γ_{th}. The test is carried for 100k samples for better resolution in identifying signals. The spectrum sensor with its sensing ability is able to sense the entire FCC assigned ISM band within 3.5 seconds. The 2.4GHz spectrum and the 5.0GHz spectrum are individually speculated where 2.4GHz spectrum covers 11 channels and 5.0GHz covers 21 channels.

In the Figure 3.16(a), the channels for 2.4GHz ranges from 2.412GHz to 2.462GHz. Similarly, in Figure 3.16(b), the channels range from 5.18GHz to 5.825GHz in 5.0GHz ISM band. The peak of the spectrum denote the signal strength of the respective spectrum in the environment. The spectrums are sensed for 100k sampling for adequate resolution of the readings.

Each energy of the signal sensed in ISM band for given number of samples denote the energy of the spectrum for that particular channel. Upon closer inspection, it can be seen from the Figure 3.17 that the amount of energy being carried by the spectrum is greater or lower than the decision threshold. The spectrum is focused on 2.412GHz center frequency to inspect the amount of signal strength. From the Figure 3.17, 2.412GHz center



(a) 2.4GHz spectrum in ISM band

(b) 5.0GHz spectrum in ISM band

Figure 3.16: Spectrum sensor sensing power in the ISM band spectrum

frequency, the signal strength reading is roughly -95dB. The γ_{th} value set for the occupancy of the spectrum is -60dB. Therefore, this spectrum is considered to be idle because $(E_n = -95dB) < (\gamma_{LO} = -60dB)$ from the equation (3.5).

Similar analysis is carried out for 5.0GHz spectrum. In the Figure 3.18, the center frequency at 5.18GHz has the signal strength of -75dB. If γ_{th} is set at -88dB, the spectrum is

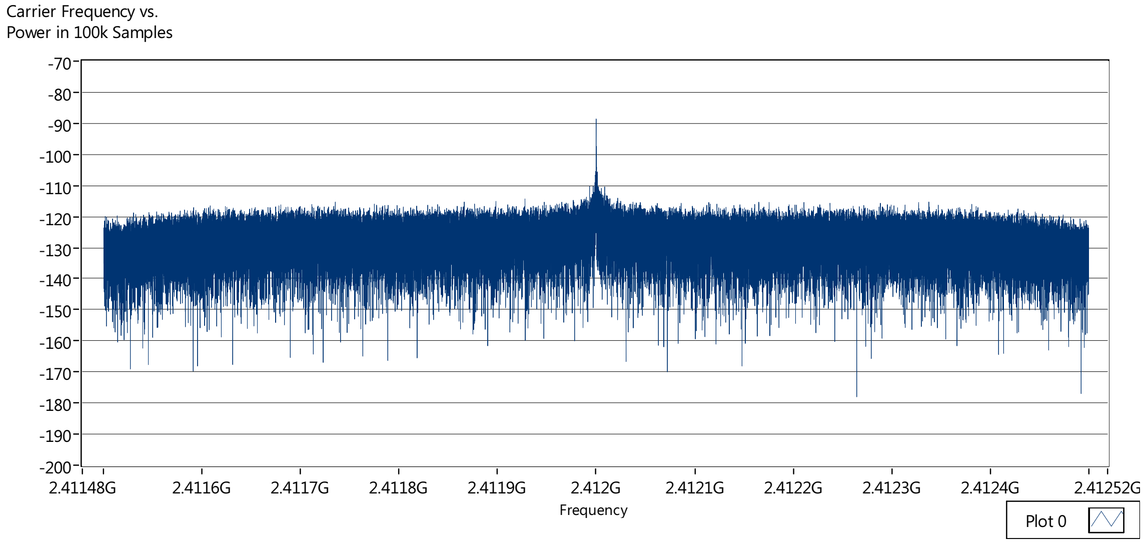


Figure 3.17: Closer inspection of 2.412GHz channel in ISM band

presumed to be occupied and is not reported to the spectrum repository. However, since γ_{th} is set at -60dB for better accuracy in spectrum reporting, the analyzed channel is considered to be unoccupied. The sensed idle spectrum is reported to the spectrum repository. Similar analysis is carried on the entire sensed spectrum.

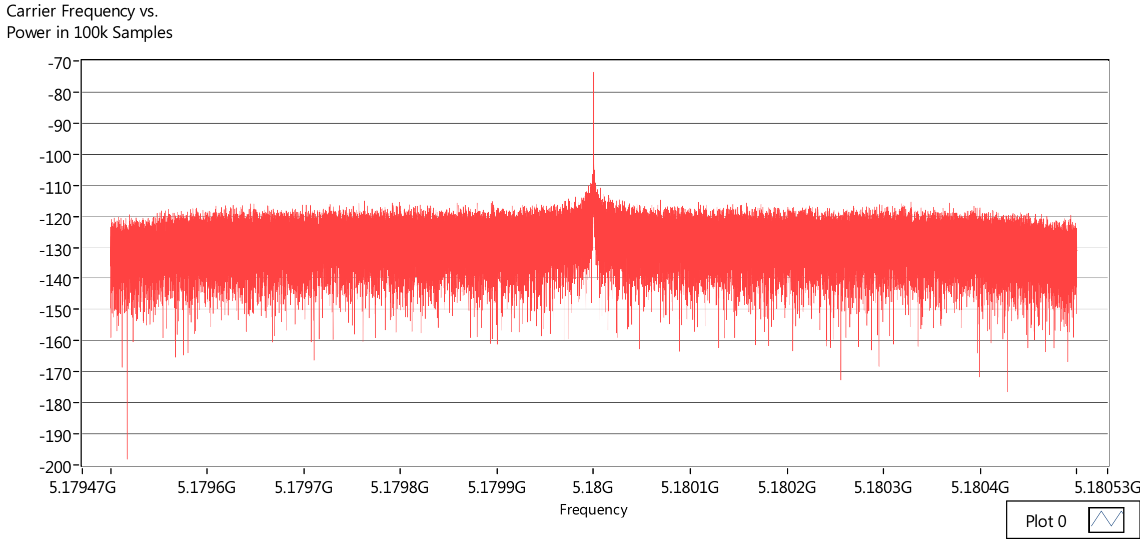
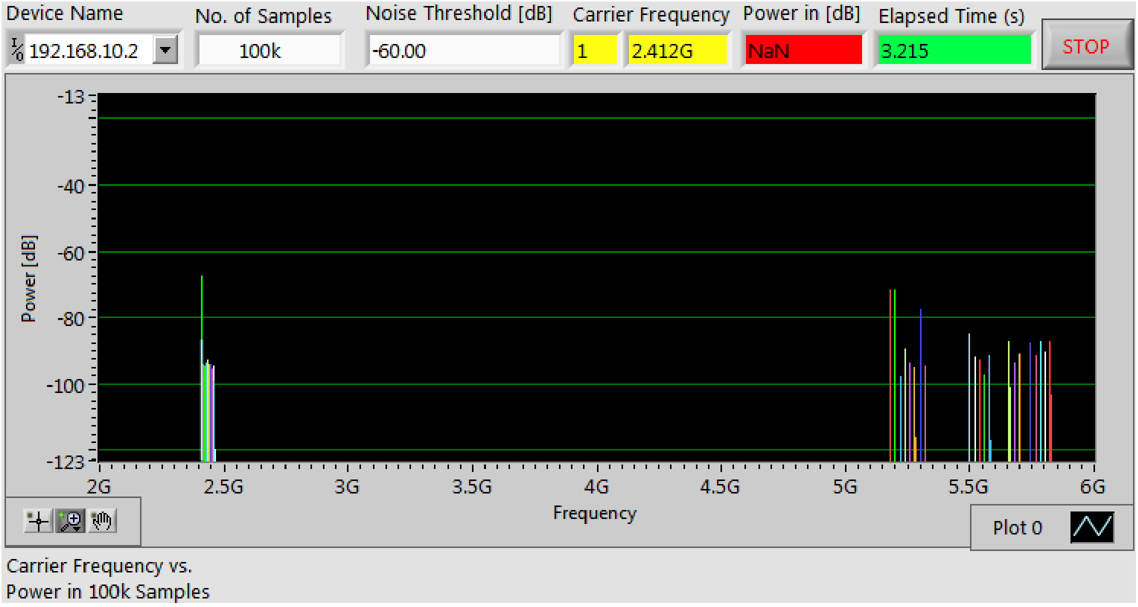


Figure 3.18: Closer inspection of 5.18GHz channel in ISM band

From the results, it is apparent that the spectrum sensor is reporting correctly the presence or absence of any signal or user in the spectrum. In order to avoid any errors in readings and, the resolution of the reading is adjusted to 100k sampling for accuracy without drastic effect to efficiency.



Device Name: 192.168.10.2
No. of Samples: 100k
Noise Threshold [dB]: -60.00
Carrier Frequency: 1 2.412G
Power in [dB]: NaN
Elapsed Time (s): 3.215

Carrier Frequency vs.
Power in 100k Samples

Figure 3.19: Scan time for 100k sampling for all ISM bands

While sensing the spectrum at 100k keeping the I/Q-rate at 1M (million), the spectrum sensing for ROAR architecture takes approximately 3.215 seconds as shown in Figure 3.19. For higher resolution of sensing by keeping higher sampling rate, the signal processing suffers efficiency. In order to find optimum sampling rate, various sampling and scanning times are measured. From the Figure 3.20, it is observed that as the resolution of the samples are increased, the system takes longer to sense for total number of channels in the system.

For deciding the best sampling rate without affecting the efficiency and accuracy, signals are analyzed for both 100k sampling and 1M sampling. The analysis of received signal for 100k for both the 2.4GHz and 5.0GHz spectrum from the Figure 3.17 and 3.18 show that the received signal strength is closer to -75dB. At 1M sampling rate, the spectrum

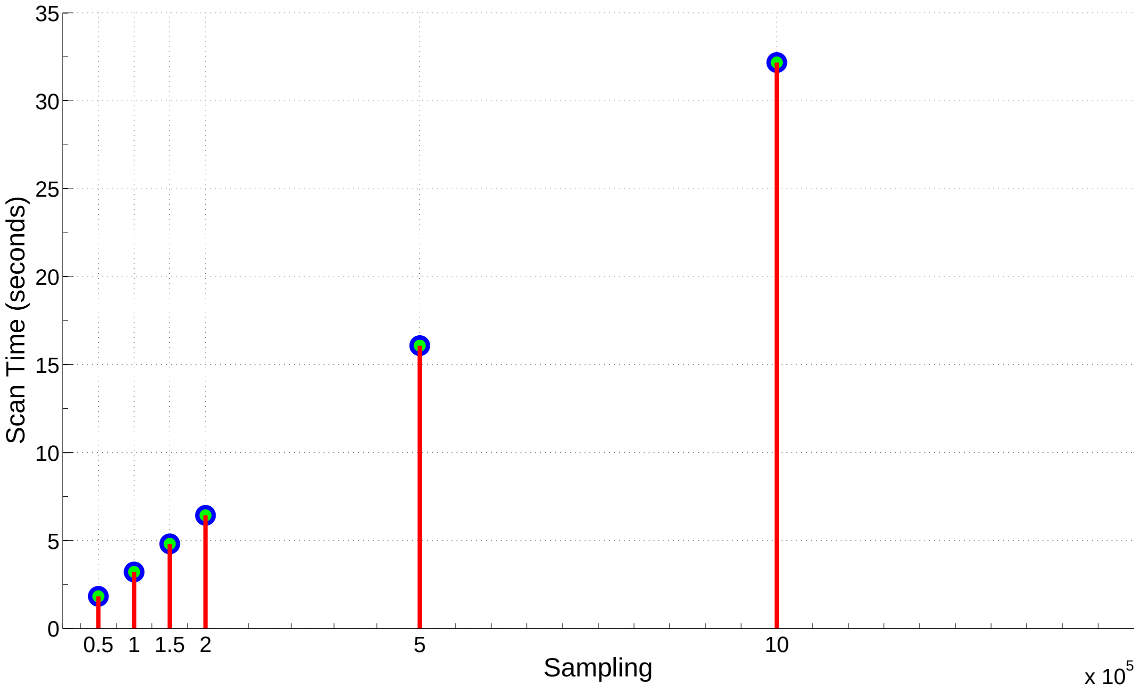


Figure 3.20: Graph of sampling vs. efficiency

in the Figure 3.21 has similar resolution and signal strength (-80dB) as spectrum at 100k sampling rate.

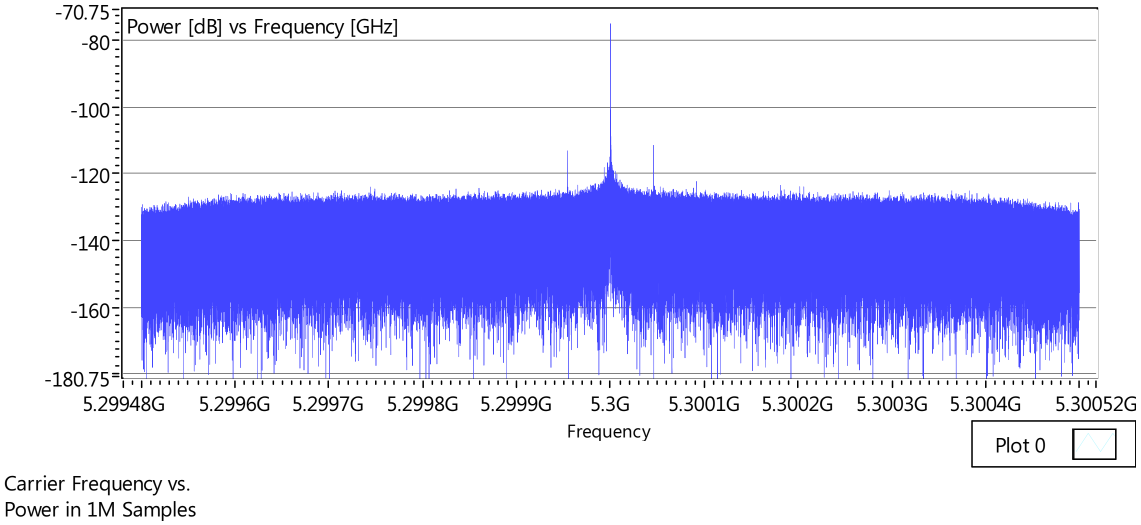


Figure 3.21: Spectrum analysis of higher resolution 1M sampling

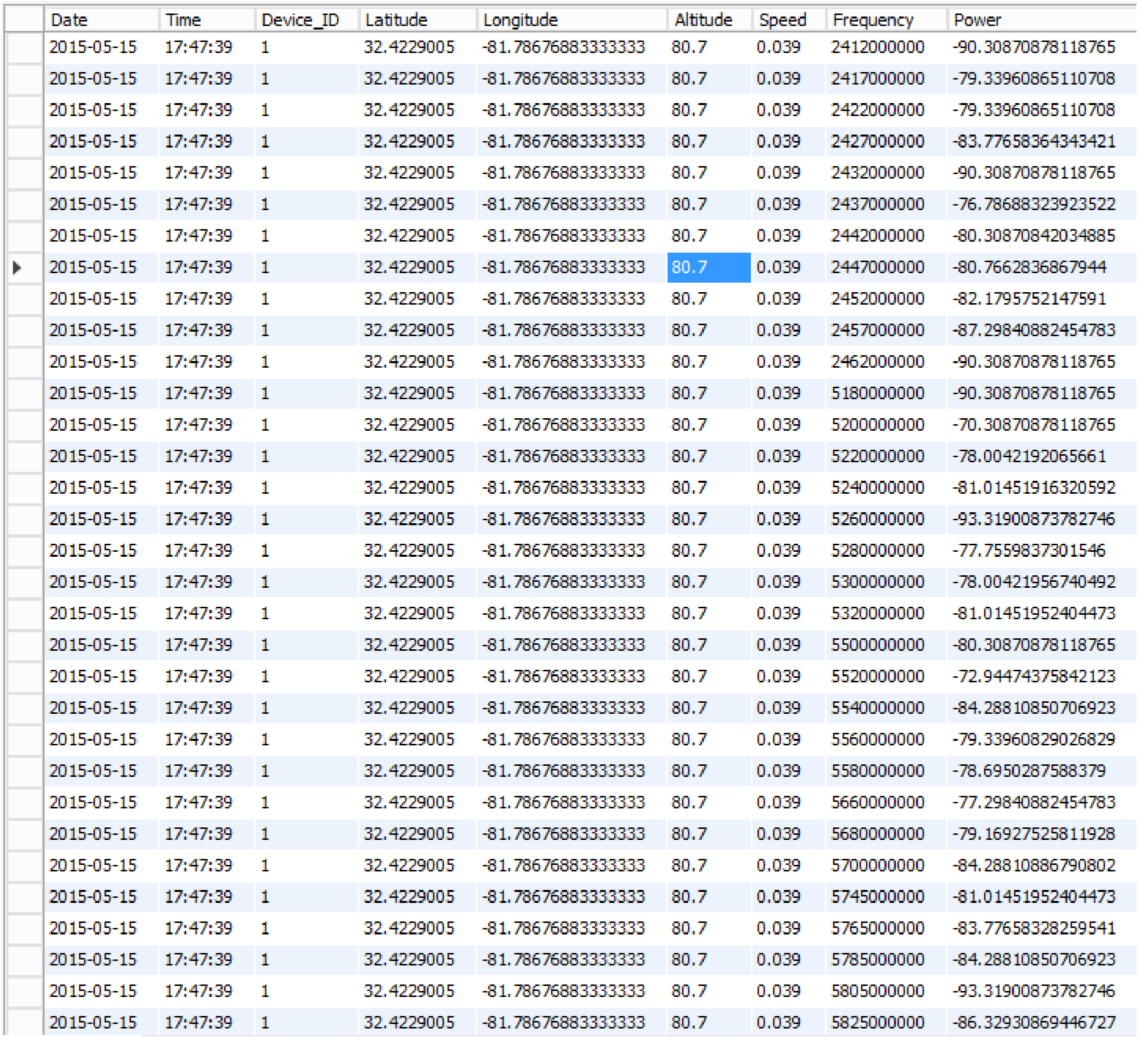
Therefore, the 100k sampling rate is decided to be adequate for spectrum sensing to be able to detect spectrum occupancy information at any given geolocation without adversely affecting the overall efficiency of the spectrum sensing testbed for ROAR architecture. In the Figure 3.21, the center frequency at 5.3GHz recorded the signal strength at -75dB. When comparing the readings obtained at 1M sampling with Figure 3.18, the signal resolution and strength is not significantly different than the one obtained at 100k sampling rate.

The sensed idle spectrum is then passed to distributed cloud-based database for storage. The database is responsible for maintaining and safe-keeping of the available spectrums for SUs when they want to use the idle spectrum opportunistically. The sensed information collected from the spectrum sensing for ROAR architecture is shown in the Figure 3.22. The information shows that it is encapsulated with geolocation and time-stamp information where different columns show different formatted information. Latitude, longitude, altitude, speed, spectrum information and signal strength are constantly updated in a tabular form as shown in the Figure 3.22.

The Figure 3.22 shows the real-time sensor readings ready to be stored in the database. The Figure 3.22 shows the database table for entire ISM band from 2.4GHz to 5.825GHz. The readings are collected on 2015-05-15 at 17:47:39 UTC time. The altitude and speed information based on the GPS are collected but are not being used in this research. However, if considering mobility, these information could be important.

3.4 Chapter Summary

In this chapter a background study is conducted in spectrum sensing where, it is decided that the energy detection is reliable in sensing idle channels in the environment for ROAR architecture. The devices being used for spectrum sensing for ROAR architecture testbeds are USRPs that perform extremely well for lower computational complexity and simplistic implementation achieved during energy based detection techniques. Depending on the



	Date	Time	Device_ID	Latitude	Longitude	Altitude	Speed	Frequency	Power
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2412000000	-90.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2417000000	-79.33960865110708
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2422000000	-79.33960865110708
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2427000000	-83.77658364343421
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2432000000	-90.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2437000000	-76.78688323923522
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2442000000	-80.30870842034885
▶	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2447000000	-80.7662836867944
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2452000000	-82.1795752147591
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2457000000	-87.29840882454783
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	2462000000	-90.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5180000000	-90.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5200000000	-70.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5220000000	-78.0042192065661
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5240000000	-81.01451916320592
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5260000000	-93.31900873782746
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5280000000	-77.7559837301546
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5300000000	-78.00421956740492
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5320000000	-81.01451952404473
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5500000000	-80.30870878118765
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5520000000	-72.94474375842123
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5540000000	-84.28810850706923
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5560000000	-79.33960829026829
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5580000000	-78.6950287588379
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5660000000	-77.29840882454783
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5680000000	-79.16927525811928
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5700000000	-84.28810886790802
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5745000000	-81.01451952404473
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5765000000	-83.77658328259541
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5785000000	-84.28810850706923
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5805000000	-93.31900873782746
	2015-05-15	17:47:39	1	32.4229005	-81.78676883333333	80.7	0.039	5825000000	-86.32930869446727

Figure 3.22: The sensed spectrum information from spectrum sensing for ROAR architecture

strength of the signal sensed in the environment, a decision threshold is defined to identify whether there is a signal in the sensed spectrum or not. If the spectrums sensed have higher energy threshold than the decision threshold, the spectrums are considered to be occupied or are considered to be utilized by either PUs or SUs. These spectrums are not reported to the distributed database. Only the idle spectrums are of interest and are reported to the distributed database.

The spectrum sensing for ROAR architecture is implemented using USRP. LabVIEW is used to program the USRP. A sensor sub-system model for ROAR architecture shows the process of spectrum sensing and idle channel/spectrum reporting with geolocation information to distributed database. The sub-system model is implemented for ROAR architecture and the evaluation is carried out.

Upon evaluating the spectrum sensing for ROAR architecture, each individual components of the sensor is speculated for accuracy and performance. After verifying the accuracy of GPS unit and measuring the effective sampling rate to sense idle spectrum in the environment, the spectrum sensing system is tested. The idle channel information obtained from the testbed is formatted in a tabular form and is stored in distributed cloud-based database. The output of the spectrum sensing for ROAR architecture is plotted on Google maps for visualization; detail is presented in the next chapter.

CHAPTER 4

CLOUD-ASSISTED DISTRIBUTED PROCESSING IN ROAR

The spectrum sensing for ROAR architecture reports the wideband spectrum (50MHz to 6GHz) occupancy information to the cloud-assisted distributed database. The distributed database in roar architecture is handled in real-time for spectrum response/reporting by the distributed cloud based computing. The detail is presented below:

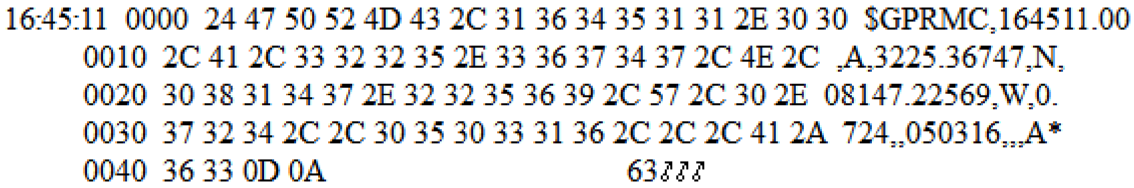
4.1 ROAR Introspective and Implications

ROAR architecture uses distributed cloud-based database to store all the idle spectrums at a given geolocation. The backbone to the research presented in this thesis is the cloud-based distributed computing responsible for handling requests/response and cloud-based distributed database responsible for idle channel storage for ROAR architecture.

4.1.1 Cloud-based Distributed Computing:

The spectrum sensing for ROAR architecture is deployed in distributed manner. These sensors in-turn are reporting idle channel information vigorously at all times. For such large scale real-time data reporting, it is important to include partitioning mechanism of sensed data based on their geolocation to lower the processing time. The topology that is capable of handling the large stream of real-time data is Storm topology. The Storm topology for data processing is currently being implemented by Twitter, Netflix, GitHub etc. for real-time large scale data reporting and faster data processing [23]. Storm topology model is of prime interest in cloud-assisted distributed computing and database storage for ROAR architecture [27].

The Storm model in cloud-based distributed computing is used in ROAR architecture to handle instantaneous idle spectrum requests/response based on the geolocation reported from the spectrum sensors. The Storm model comprises of stream of data defined as the unbounded sequence of tuples (ordered set of values as shown in Figure 4.1).



```
16:45:11  0000  24 47 50 52 4D 43 2C 31 36 34 35 31 31 2E 30 30  $GPRMC,164511.00
          0010  2C 41 2C 33 32 32 35 2E 33 36 37 34 37 2C 4E 2C  ,A,3225.36747,N,
          0020  30 38 31 34 37 2E 32 32 35 36 39 2C 57 2C 30 2E  08147.22569,W,0.
          0030  37 32 34 2C 2C 30 35 30 33 31 36 2C 2C 2C 41 2A  724,,050316,,,A*
          0040  36 33 0D 0A                                      63⸮⸮⸮
```

Figure 4.1: Geolocation data acquired by the GPS unit in spectrum sensing for ROAR architecture in the form of tuple data-type

The tuple shown in Figure 4.1 is generated at the spout shown in Figure 4.2 prior to streaming towards the bolts (also shown in Figure 4.2) for processing. The bolts are responsible for transforming the streams to produce additional streams. The combination of spouts, tuples and bolts form a topology known as Storm topology as shown in the Figure 4.2.

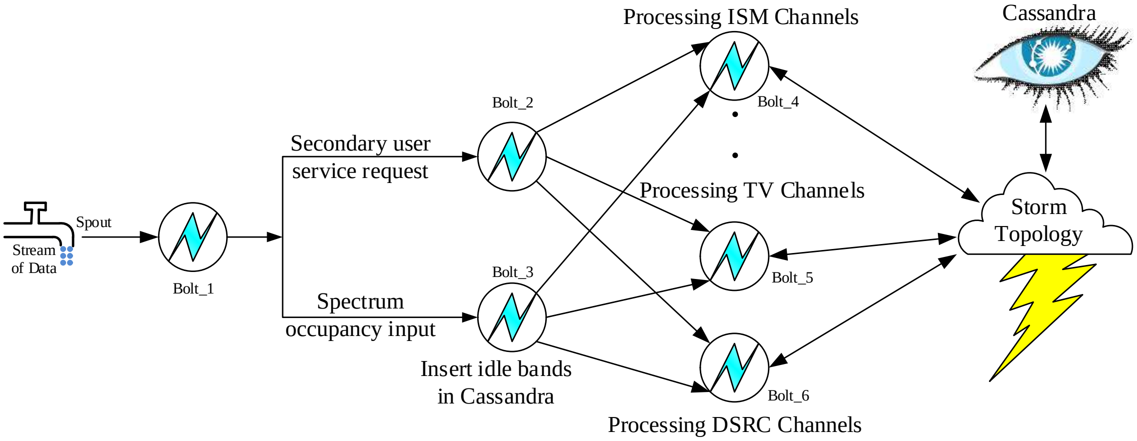


Figure 4.2: Cloud-assisted distributed computing for ROAR architecture using Distributed Remote Procedure Call (DRPC) in Storm

In distributed computing in Storm topology, concept of bolts is introduced that is responsible for handling parallel requests/response [23]. Therefore, there might be multiple bolts handling multiple streams of data from spouts being processed at the same time. In ROAR architecture, the data is generated by the spectrum sensors and SUs at the same time. Therefore, the challenge is to process the data to handle the requests coming from the SUs and the idle channel information coming from the spectrum sensors in real-time.

For opportunistic spectrum access in ROAR architecture, referring to the Figure 4.2, the bolts are responsible for processing the SUs service requests for idle channels and spectrum occupancy information of primary incumbent services. The individual bolts are responsible for parallel processing of information related to any of the ISM channel, TV channel and dedicated-short-range-communication (DSRC) services as shown in the Figure 4.2 put forward by the SU. At the same time, due to the robust nature of bolts in Storm topology, the spectrum sensors could also report idle channel information to the same bolt via the same stream originating at the spout. These information are handled independently in real-time in Storm topology prior to reporting to the Cassandra database for distributed storage.

4.1.2 Cloud-based Distributed Database:

The distributed computing routes the information from the SUs and spectrum sensors for ROAR architecture in parallel processing and stores them to Cassandra database. Cassandra database is responsible for storing spectrum occupancy information of heterogeneous wireless network for various geolocations in cloud-based distributed database platform [23].

It has been mentioned in Chapter 2 that FCC mandates the necessity for SUs to search for opportunity in geolocation database to look for idle spectrums and establish connection only if it does not affect the licensed PUs [21]. A prior knowledge of spectrum collectors for ROAR architecture is known to remotely located cloud-based distributed geolocation database which allows the database system to establish secured connection and check for any intrusion that might compromise the PUs infrastructures.

The cloud-assisted distributed database store idle channel information using database framework invovlving hash table [27]. The dynamic hash table stores database information where the table consists of coordinates (x, y, z) where the reading is taken, radius r till where the service can be administered and T the time to live for any information stored; as shown in Figure 4.3 [23].

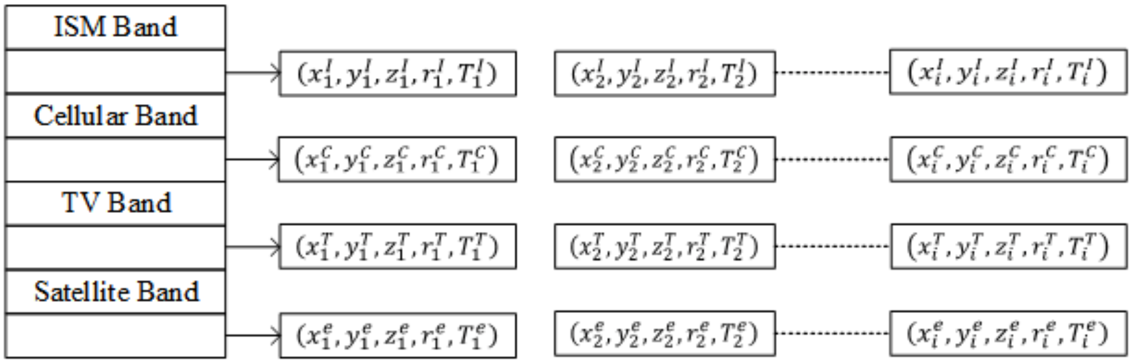

ISM Band
$(x_1^I, y_1^I, z_1^I, r_1^I, T_1^I)$
$(x_2^I, y_2^I, z_2^I, r_2^I, T_2^I)$
$(x_i^I, y_i^I, z_i^I, r_i^I, T_i^I)$
Cellular Band
$(x_1^C, y_1^C, z_1^C, r_1^C, T_1^C)$
$(x_2^C, y_2^C, z_2^C, r_2^C, T_2^C)$
$(x_i^C, y_i^C, z_i^C, r_i^C, T_i^C)$
TV Band
$(x_1^T, y_1^T, z_1^T, r_1^T, T_1^T)$
$(x_2^T, y_2^T, z_2^T, r_2^T, T_2^T)$
$(x_i^T, y_i^T, z_i^T, r_i^T, T_i^T)$
Satellite Band
$(x_1^e, y_1^e, z_1^e, r_1^e, T_1^e)$
$(x_2^e, y_2^e, z_2^e, r_2^e, T_2^e)$
$(x_i^e, y_i^e, z_i^e, r_i^e, T_i^e)$

Figure 4.3: Dynamic hash table for various incumbent primary services sensed and retained for ROAR architecture

The Figure 4.3 shows that for any corresponding bands, such as ISM (denoted by superscript (I)), Cellular (C), TV (T) and Satellite (e), the distributed database structure store all the required information on the hash table in corresponding table. Depending on the time to live (TTL), the information is retained, but as soon as the time-to-live, $T_i^{(I,C,T,e)}$ expires, the information stored in hash table is archived prior to deletion. That way, the information is retained for future reference if need be and as long as it is updated by the spectrum sensors. If the spectrum sensors do not report the availability of the spectrums to the database for long period of time, then the value of $T_i^{(I,C,T,e)}$ becomes zero and the spectrum gets deleted from the hash table indicating the SUs that the spectrum is no longer available.

The block diagram of database reporting from the spectrum sensing for ROAR architecture is as follows:

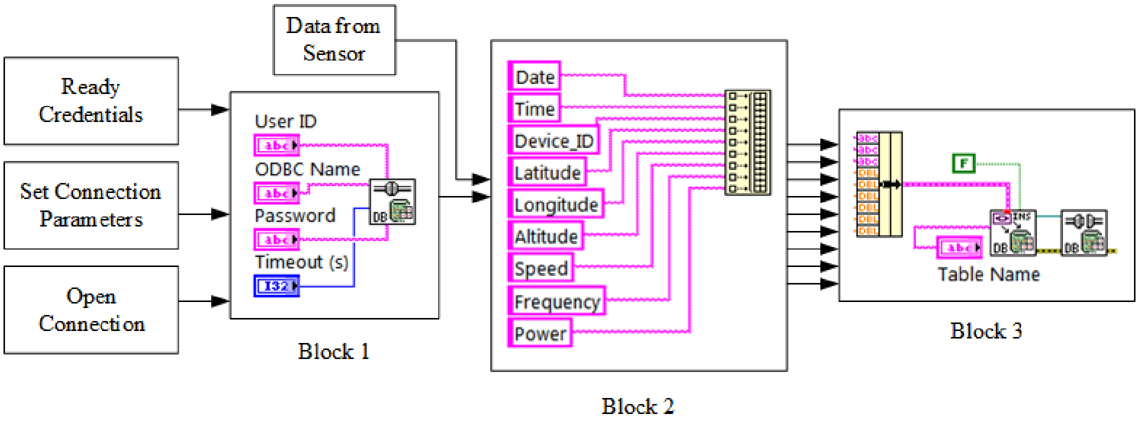


Figure 4.4: Block diagram representation of database reporting from spectrum sensors implemented in LabVIEW program

The abstraction of the complete program implemented in LabVIEW as shown in the Appendix A.3 is shown in the Figure 4.4. Here, the Block 1 shows the process of setting the database parameters by setting the credentials. Once the connectivity to the database is established, the sensor data is passed into the tabular form. Block 2 processes the crude data (tuple) from the spectrum sensor into formatted data and then passes onto Block 3. In Block 3, the formatted data is passed onto the table with specific table name. For this whole process, if connection is established within the timeout value, the database reporting will not terminate. The connection terminates either after successful reporting or if timeout value expires. The idle channel information is then used by SUs to establish connection.

Database operability and functionality is very important to the research. The ROAR architecture testbed completely depends on the geolocation database table for spectrum reporting from sensors' perspective and opportunistic spectrum selection for device-to-device (D2D) or device-to-infrastructure (D2I) communication from SUs' perspective. The detail of cloud-based cognitive radio for opportunistic spectrum access for ROAR D2D and D2I communication is discussed in next Chapter.

4.2 Spectrum Visualization through Heat-map

The idle spectrum information encapsulated with geolocation data is stored in Cassandra database and it can be visualized using Google maps API. The data stored can be plotted as heat-index. The complete process by which the idle spectrum information is processed, stored and visualized is shown in the Figure 4.5.

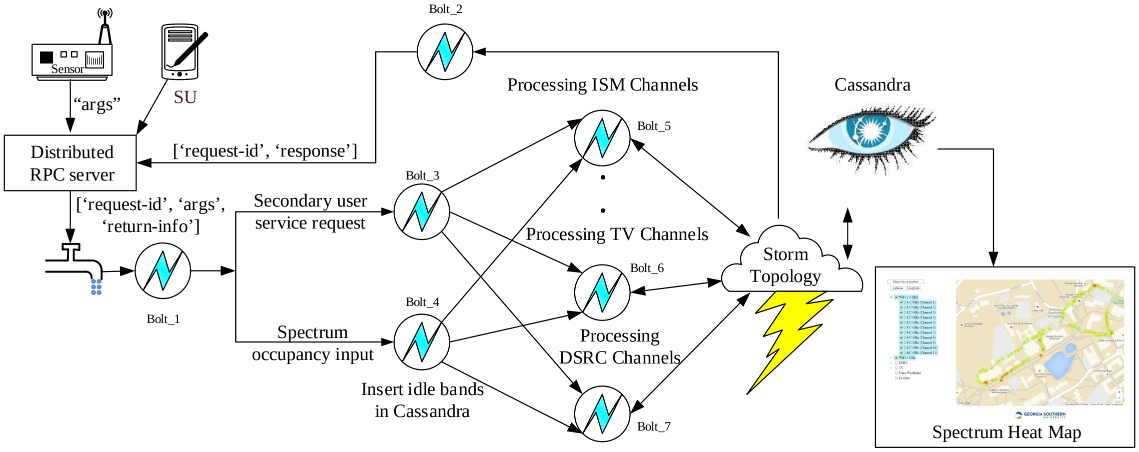
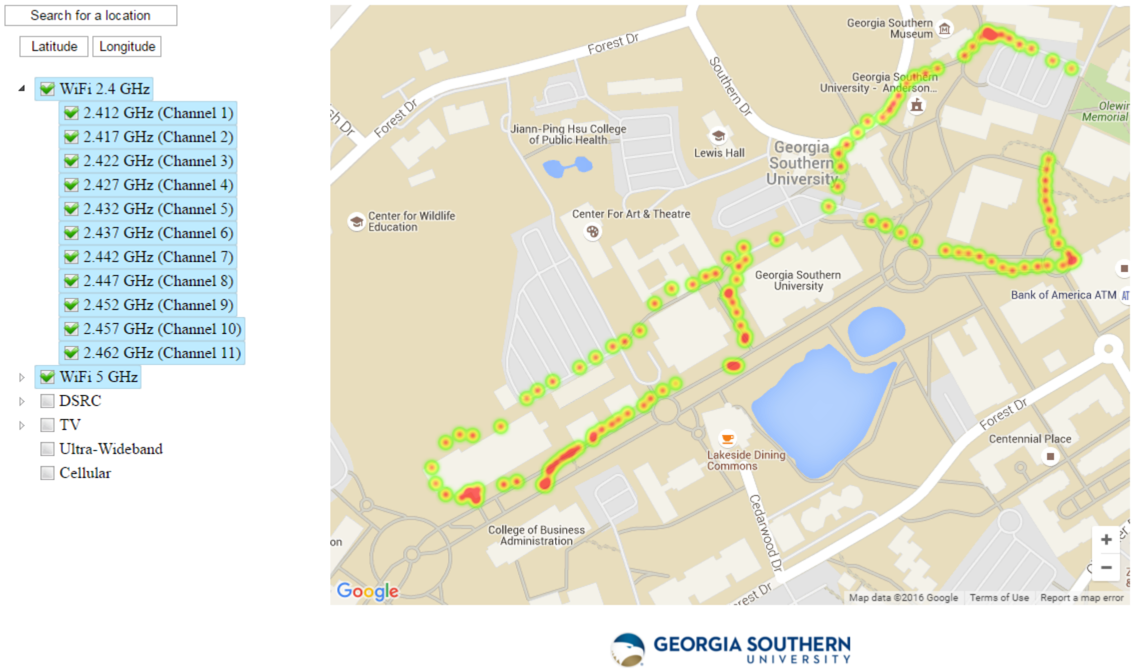


Figure 4.5: Spectrum reporting, processing, storage in Cassandra and heat-map visualization

In the Figure 4.5, the spout originating at the spectrum sensor is processed at DRPC server and based on the arguments, the stream passes onto bolts. The arguments contain request-id and return-info that is used for independent processing at the bolts. The bolts at the partitioning handle spectrum reporting from various sensors sensing incumbent primary services which results in faster parallel processing. The arguments are either processed or accumulated to be stored in Cassandra distributed database and the idle spectrum stored in Cassandra is displayed as heat-map on Google maps displaying the available spectrums at any given geolocation.

The Google map plot shows the heat index for spectrum density for given geolocation. In the Figure 4.6, the dark spots show that the number of idle channels are more in that geolocation. The sparse data shows that there are no idle channel remaining in that area and if SUs want to communicate, they have to either move to a different geolocation or wait till some idle channels are available.


Search for a location

Latitude | Longitude

- ☑ WiFi 2.4 GHz
  - ☑ 2.412 GHz (Channel 1)
  - ☑ 2.417 GHz (Channel 2)
  - ☑ 2.422 GHz (Channel 3)
  - ☑ 2.427 GHz (Channel 4)
  - ☑ 2.432 GHz (Channel 5)
  - ☑ 2.437 GHz (Channel 6)
  - ☑ 2.442 GHz (Channel 7)
  - ☑ 2.447 GHz (Channel 8)
  - ☑ 2.452 GHz (Channel 9)
  - ☑ 2.457 GHz (Channel 10)
  - ☑ 2.462 GHz (Channel 11)
- ☑ WiFi 5 GHz
- □ DSRC
- □ TV
- □ Ultra-Wideband
- □ Cellular

Figure 4.6: Heat-map representation of idle spectrum at respective geolocation based off of Cassandra database

The Google maps API as shown in the Figure 4.6 is programmed using javascript. The website is built on Active Server Pages (ASP) scripting. The Google maps is embedded in the website on top of which a “Firebase (a real-time data reporting from cloud-base database created by Google)” framework is implemented to display the heat-map. Each data on the heat-map represent idle spectrum from the last reading. The checkboxes on the right of heat-map helps visualize the individual spectrum data that might be available at the given geolocation.

The spectrum sensing testbed and database reporting for ROAR architecture is designed, implemented and tested at Georgia Southern University campus where the readings for spectrum availability is taken between the IT building and Russell Union. The heat-map indicates the list of channels (2.4GHz and 5GHz spectrum) that are available at those geolocations where the sensors took idle spectrum readings.

4.3 Chapter Summary

In this chapter, cloud-based distributed computing and database for ROAR architecture is designed, analyzed, implemented and evaluated. The cloud-based distributed computing is used for real-time parallel processing of the arguments, queries and responses from spectrum sensors or SUs for opportunistic access using Storm topology. Similarly, the cloud-based distributed database is used for storage of geolocation spectrum information and the availability of idle channels in Cassandra database framework.

The idle channels from Cassandra database is then used in Google maps API heat-map to visualize the availability of idle spectrums for any given geolocation. The heat-map is created using javascript and Google's "Firebase" framework for Google maps API. The heat-map shows the available channels that are obtained through real-time spectrum sensing at Georgia Southern University for ISM, TV, cellular bands. This information can be used by SUs to communicate with each other without causing harmful interference to PUs. Therefore, by implementing the FCC mandated use of geolocation database in the testbed designed for ROAR architecture, the research is able to implement faster, reliable and efficient database services with cloud-based computing to leverage power, computational capacity and memory off of SUs and onto the remote infrastructure for better performance.

CHAPTER 5

OPPORTUNISTIC SPECTRUM ACCESS IN ROAR

After successfully implementing the cloud-assisted distributed computing and database storage for ROAR architecture, the idle spectrum stored in spectrum repository is used for opportunistic spectrum access by D2D or D2I communication.

5.1 A Concept to Design

In opportunistic spectrum access in ROAR architecture, the SUs use the idle channels opportunistically to communicate with each other. In order to do so, the SUs first send request to get the list of available spectrum from the distributed database. The process is handled by the distributed computing where the Storm topology handles the request in parallel processing. Depending on the location of the SUs placing a request, the cloud-based distributed computing decides the list of idle channels that need to be handed to the SUs based on the haversine distance between the SUs and the location where the idle spectrums are available.

Once the SUs receive the list of idle channels, they implement quorum-based channel selection schemes to find an identical channel that can be used for communication. The reason for using quorum-based common channel selection is to quickly select the common channel and provide secured communication. While the SUs are communicating, the spectrum sensing for ROAR architecture located at the same location as the SUs can sense the use of idle channels and updates to the distributed database via distributed computing that the idle channel is no longer available. Therefore, when second pair of SUs arrive at the same location, they can never access the previous idle channels and are provided new set of idle channels to establish communication. Like this way, the SUs establish communication dynamically without causing harmful interference to other users in the environment. When PUs are administered into the same location, due to their incumbent primary services access, the SUs communication is dropped as to prioritize PUs, and the SUs immediately move to another available channel. The whole process is seamless and does not cause any interference to the PUs.

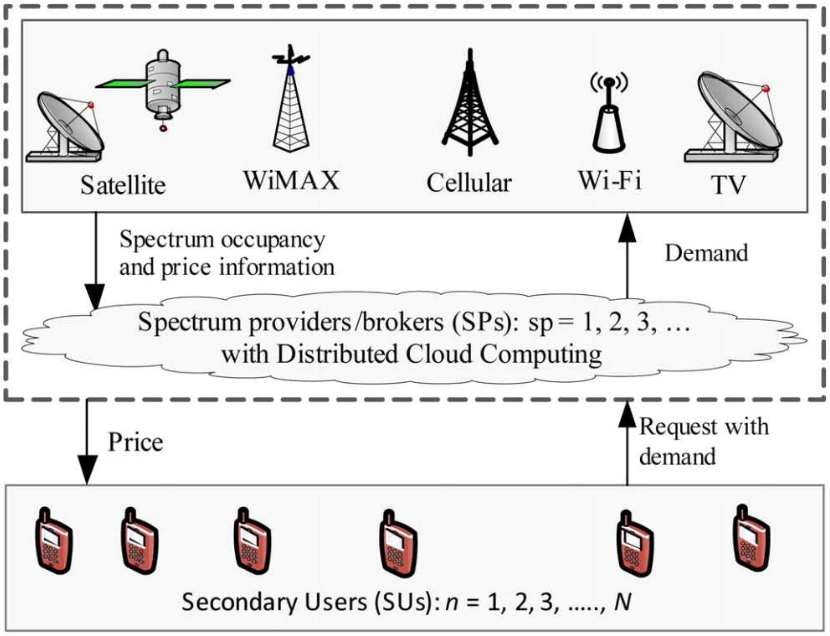

Satellite
WiMAX
Cellular
Wi-Fi
TV
Spectrum occupancy and price information
Demand
Spectrum providers/brokers (SPs): sp = 1, 2, 3, …
with Distributed Cloud Computing
Price
Request with demand
Secondary Users (SUs): n = 1, 2, 3, ..., N

Figure 5.1: SUs dynamically access the idle spectrum of primary services/infrastructure

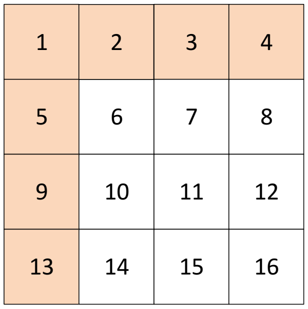
The use of opportunistic spectrum access can be seen in the Figure 5.1 where the SUs can access any spectrum from satellite services or cell phone services dynamically by querying the distributed cloud computing in the absence of PUs [29]. The primary infrastructures using exclusive licensed spectrum are sensed and their spectrum availability information is stored at the distributed cloud database. Upon requests/demands from the SUs, the information stored in geolocation database is extensively used by the distributed cloud computing for retrieving and assigning the available spectrum and services to the SUs [29].

The DSA where the SUs use the opportunity to establish communication is achieved solely by relying on the availability of spectrums.The opportunistic spectrum access in ROAR architecture can only be practical after successful sensing and reporting of available idle channels in the environment to the spectrum repository. Moreover, the SUs equipped with transmitter and receiver must be willing to use opportunistic spectrum access to indulge in this simple and elegant means of communication. Once the SU-transmitter (SU_{Tx}) and SU-receiver (SU_{Rx}) pair acquire the list of available idle channels, they should be able to sense each other to establish communication. There are two methods they both can find common channel. One method is to use sequential sensing where both SU_{Tx} and SU_{Rx} use the idle channel information in sequential manner, one spectrum at a time, until they both find a common channel. The other method is to implement quorum-based rendezvous for common channel selection where SU_{Tx} and SU_{Rx} use quorum scheme for faster, secure and reliable communication.

The main idea in finding relevant channels depend on frequency hopping techniques. In conventional systems, communication channels are established by pre-determined spectrum selection either through fixed or sequential spectrum hops. In the research paper [37], three types of multicast rendezvous algorithms, AMQFH, CMQFH and nested-CMQFH is proposed where these algorithms are designed using uniform k-arbiter and the Chinese remainder theorem for quorum systems. For simplistic approach and for faster USRP response, a different quorum is developed than the one presented in [37]. The quorum being implemented in the ROAR architecture testbed is presented in the conference paper [38].

The sequential search takes longer to sense through all the available idle channels until it finds a common channel. This process results in energy wastage for SU_{Tx} and SU_{Rx} pair. Furthermore, SU_{Tx} slow hops to different channels results in problems for adaptive systems if SU_{Tx} and SU_{Rx} while using sequential hop never meet. Therefore, the sequential channel search is not very intuitive when it comes to frequency hopping schemes. Plus, this problem becomes worse when there are many idle channels in heterogeneous wireless environment. To avoid this problem, quorum-based rendezvous method is used where the SU_{Rx} carries more predictive channel hops and selects the channel once it detects SU_{Tx} transmitting at corresponding channel.

For better understanding, if 2.412GHz is represented as channel 1, 2.417GHz as channel 2, 2.427GHz as channel 3 and so on and so forth upto 5.280GHz as channel 16, then the channel grid can be represented serially as shown in Figure 5.2. In quorum-based common channel selection process, the user lists all the available spectrum acquired from the database in a matrix form. Then it applies selective frequency hopping technique to individually access the elements from the presented grid. Comparison is also made with regards to sequential frequency hops and the results are presented.



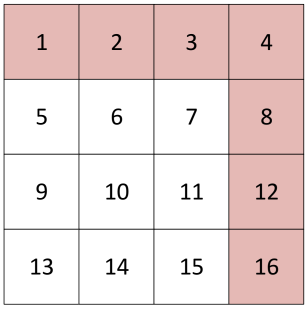
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 5.2: Quorum-based common channel selection scheme for SU_{Tx}

The Figure 5.2 is only for reference and for understanding various quorum-based approach. The ROAR transceiver testbed (SU_{Tx} and SU_{Rx}) can use any number of available spectrum and is not limited to the grid size shown in this thesis. Also, there are multiple quorum techniques and each transceiver could implement different quorum-based common channel selection techniques [37].

In the Figure 5.2, the SU_{Tx} lists all the available frequencies obtained from the database in a matrix form. The SU_{Tx} then begins to transmit by selecting corresponding channel from the grid matrix. Considering the channel listing for the available frequencies to be exactly the same as shown in Figure 5.2, the SU_{Tx} first begins to transmit in channels $\{4,3,2,1,5,9,13\}$. The transmitter stays on that frequency for long enough to give the SU_{Rx} ample opportunity to identify the SU_{Tx}. After certain time has elapsed, the SU_{Tx} will then begin transmitting in $\{8,7,6,10,14\}$ followed by $\{12,11,15\}$. It will repeat the process as long as all the elements in the matrix are completed and the SU_{Tx} has transmitted on the channel $\{16\}$.

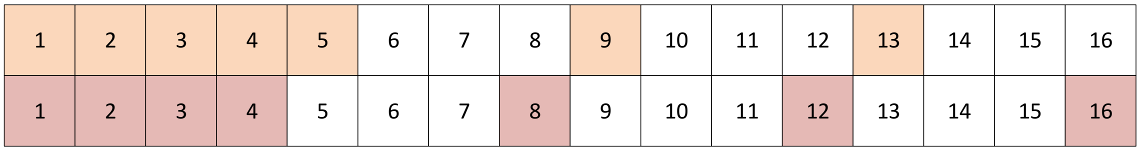
The SU_{Rx} on the other hand might apply different quorum to find the common channel in a ”blind-date” channel selection process where the SU_{Rx} hops to different frequencies hoping to meet the SU_{Tx} on common channel. The term ”blind-date” here refers to the phenomenon where the receiver is unknown of the frequency the SU_{Tx} is transmitting and is hoping to meet it by following certain quorum-based approach. Thus, it is a “blind-date” for the SU_{Rx} who is unaware of the transmitter. Presented in Figure 5.3 is the SU_{Rx} that might apply same quorum as SU_{Tx} or a different quorum scheme to pair-up.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 5.3: Quorum-based common channel selection scheme for SU_{Rx}

In the quorum scheme shown in Figure 5.3, the SU_{Rx} will first begin to receive the spectrum on channel $\{1,2,3,4,8,12,16\}$. The SU_{Rx} tunes to respective spectrum using grid numbering as discussed in the beginning of this section. The chances of meeting the SU_{Tx} by SU_{Rx} on these channels with SU_{Tx}’s own quorum is $4/7$. Similarly, the SU_{Rx} will continue to search for its “blind-date” as long as it doesn’t find it by moving in to channel $\{5,6,7,11,15\}$ as channels $\{3,8\}$ are already used in the first step. This time the chances of meeting SU_{Tx} in this ”blind-date” is $2/5$ which is less than the first chance. As the process continues the chances of meeting the pair gets lower and lower depending on the type of quorum selected by the SU_{Tx} and SU_{Rx} pairs.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 5.4: Quorum-based common channel selection process for SU_{Tx} and SU_{Rx}

For the testbed and prototyping, respective SU_{Tx} and SU_{Rx} use opportunistic spectrum access for ROAR architecture, where the pairs use the same quorum channel hops with random frequency placement. Random frequency placement places the channels in random order and not the ascending order as shown in Figure 5.2 and 5.3. The randomness in the prototype provides relatively higher security towards channel placement and at the same time delivers faster and secure common channel reception for the frequency hopping opportunistic SUs when they successfully communicate without causing any harmful interference to the PUs.

5.2 Device-to-Device (D2D) Communication Implementing Quorum-based Rendezvous

The opportunistic spectrum access is implemented in device-to-device (D2D) communication in ROAR architecture. For ROAR D2D communication architecture, USRP enabled devices are chosen as shown in Figure 5.5. The USRP devices are equipped with GPS units and antennas andare connected to spectrum repository via secondary connection from where they can access the available idle channels. In the Figure 5.5, a pair of SUs (SU_{Tx} and SU_{Rx}) are shown that are both connected to spectrum request systems (the computer). These SUs are identical and share same spectrum range. Their communication protocol is D2D which is based on the sub-system model as shown in the Figure 5.6

From the ROAR D2D sub-system model as shown in Figure 5.6, it is observed that the

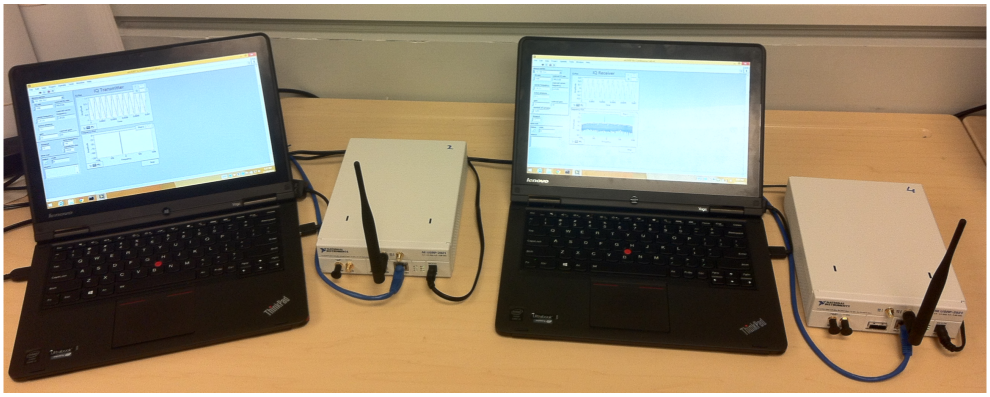


Figure 5.5: ROAR testbed for opportunistic spectrum access in D2D communication

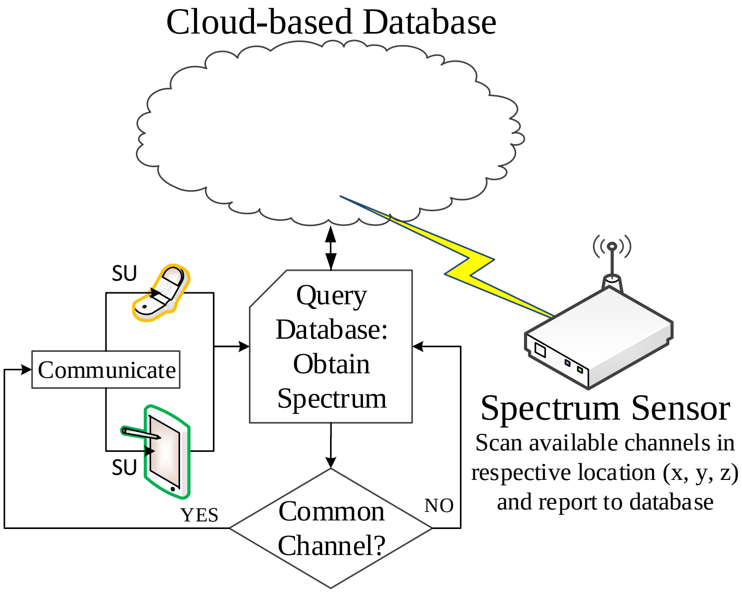

Cloud-based Database
SU
Communicate
SU
Query Database: Obtain Spectrum
YES
Common Channel?
NO
Spectrum Sensor
Scan available channels in respective location (x, y, z) and report to database

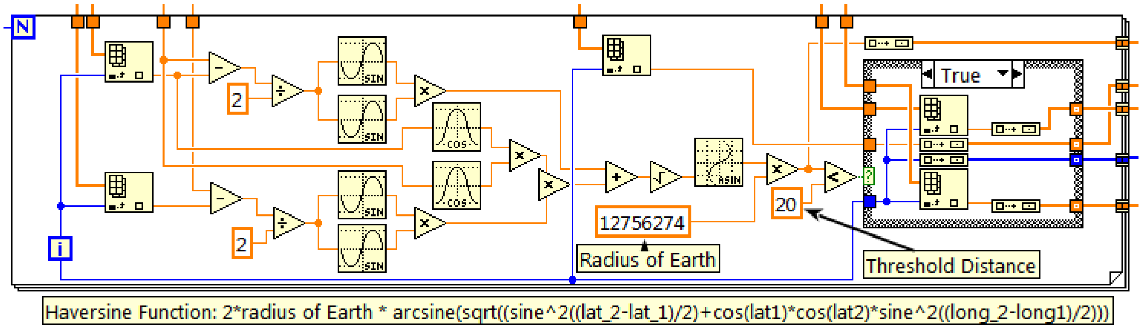
Figure 5.6: Proposed sub-system model of opportunistic spectrum access in D2D ROAR architecture

spectrum sensor senses available channels at any given location (x, y, z) and reports the idle spectrum at that location to the distributed cloud-based database. The SUs try to establish communication based on the request and response received from the spectrum repository. Here the "Query Database" is the spectrum repository that is responsible for handling the spectrum demand and supply to the SUs.

The SUs that are trying to establish opportunistic D2D common channel communication in ROAR architecture are SU_{Tx} and SU_{Rx}. The users rely on the distance between the sensors and themselves for acquiring available spectrum. The threshold distance plays a key role in assigning the users the list of available frequencies within their vicinity. The distance is therefore calculated using haversine distance formula. If SU_{Tx} has coordinates $J_1^s = (x_1^s, y_1^s)$ and the spectrum sensor has stored the available channels based on its GPS location as $L_1^b = (x_1^b, y_1^b)$, then the haversine distance can be calculated as,

$$d(L_1^b, J_1^s) = 2R \arcsin \sqrt{\sin^2 \frac{(x_1^b - x_1^s)}{2} + \cos(x_1^b)\cos(x_1^s)\sin^2 \frac{(y_1^b - y_1^s)}{2}} \qquad (5.1)$$

Here, a two dimensional plane is considered where the haversine distance is calculated assuming that the users are not in different altitudes. The equation (5.1) suffices for such condition and based on the threshold distance, which is the range of SU_{Tx} and SU_{Rx} (transceiver pair), the cloud-based distributed computing located on the distributed cloud-based database decides the available spectrum from the distributed cloud based database and hands it over to the SUs. The LabVIEW function block diagram for haversine distance is custom built for the research and is presented in this thesis as shown in the Figure 5.7.



Haversine Function: 2*radius of Earth * arcsine(sqrt((sine^2((lat_2-lat_1)/2)+cos(lat1)*cos(lat2)*sine^2((long_2-long1)/2)))

Figure 5.7: Custom built haversine distance using function block diagram in LabVIEW

Based on the distance, the distributed cloud computing assigns available frequencies from spectrum repository to both SU_{Tx} and SU_{Rx}. The detail LabVIEW program is shown on Appendix A.5 and the block diagram representation of the SUs transceiver receiving available spectrum from the distributed cloud-based database is shown in the Figure 5.8.

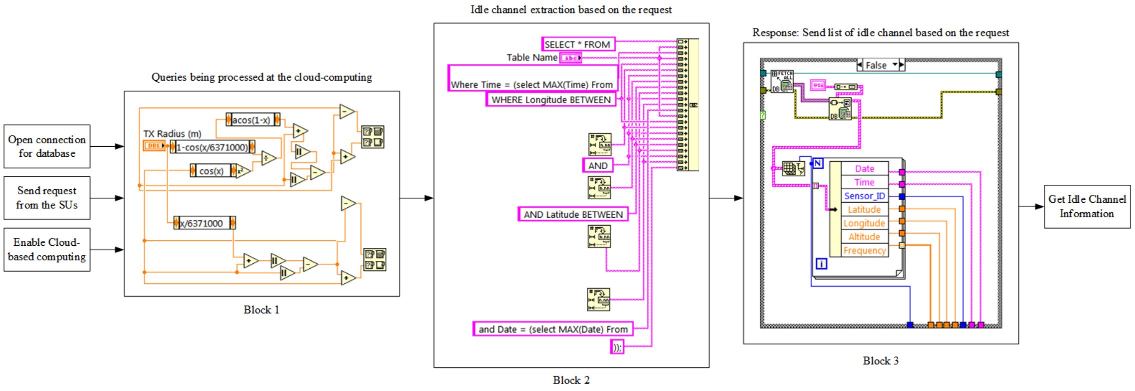

Idle channel extraction based on the request
Queries being processed at the cloud-computing
Response: Send list of idle channel based on the request
Open connection for database
Send request from the SUs
Enable Cloud-based computing
TX Radius (m)
acos(1-x)
1-cos(x/6371000)
cos(x)
x/6371000
Block 1
SELECT * FROM
Table Name
Where Time = (select MAX(Time) From
WHERE Longitude BETWEEN
AND
AND Latitude BETWEEN
and Date = (select MAX(Date) From
));
Block 2
False
Date
Time
Sensor_ID
Latitude
Longitude
Altitude
Frequency
Block 3
Get Idle Channel Information

Figure 5.8: Block diagram representation of opportunistic spectrum access from SUs perspective implemented in LabVIEW

From the Figure 5.8, it can be seen that the SUs send request to get available idle channels. As soon as the request is originated at the SUs' terminal, an instance is created at the cloud-computing end where a connection is set from the SUs and the stream of request begin to flow. In Block 1, the queries are processed at the distributed cloud-assisted computing where a decision is made based on the geolocation of the request. Once the geolocation information is processed and the location of the SUs are determined, the information is passed onto Block 2. From the Block 2, distributed database extracts the information regarding available idle channel at that particular location. Once all the channels are listed, the collected information is passed onto Block 3. The Block 3 is responsible for formatting the idle spectrum information for SUs to be able to read them. Once the idle spectrum information for that given geolocation is formatted as “Data”, “Time”, “Sensor ID”, “Latitude”, “Longitude”, “Altitude” and “Frequency”, the entire information is passed onto the SU that requested for idle channel information. The Figure 5.8 shows the operation similar to the one proposed at the distributed Storm topology responsible for both cloud-assisted computing and Cassandra based database storage framework. This whole process provides SUs the idle channel information. However, the process by which D2D then use the acquired idle channels to communicate is explained in detail using the provided flowchart as shown below:

Flowchart of D2D Opportunistic Spectrum Access:

The flowchart in Figure 5.9 gives the logical overview of ROAR architecture for D2D communication using opportunistic spectrum access. It shows the entire process involved in establishing successful communication in D2D.

In the flowchart, the most important elements of ROAR architecture for D2D communication are the GPS, database, ROAR transceivers (USRP) and cloud-computing. Only after latching to the GPS and with valid GPS data, the system begins to fetch spectrum information. The GPS then checks the time-stamp at the time of read and validates the coordinate data based on NMEA-0183 standard. At the same time, the D2D SUs trying to establish communication send authenticated secured access to the database using their credentials (username, password and distributed cloud server access). The distributed cloud-computing uses haversine distance to compare the distance between the location where the SUs are located and the area of demand. By leveraging the computation to distributed cloud based service, the SUs are saving their computational capacity along with the power required to calculate the distance. Once the distributed cloud-computing lists the available spectrum for given geolocation, it then passes to the SUs for opportunistic access.

The Figure 5.10 shows the list of available frequencies extracted from the database in real-time by the SU_{Tx} prior to communicating with SU_{Rx}.

In Figure 5.10, the received spectrum list of SU_{Tx} is graphed in 3D. The graph shows the available spectrum retrieved at specific time and date. The graph also shows the signal

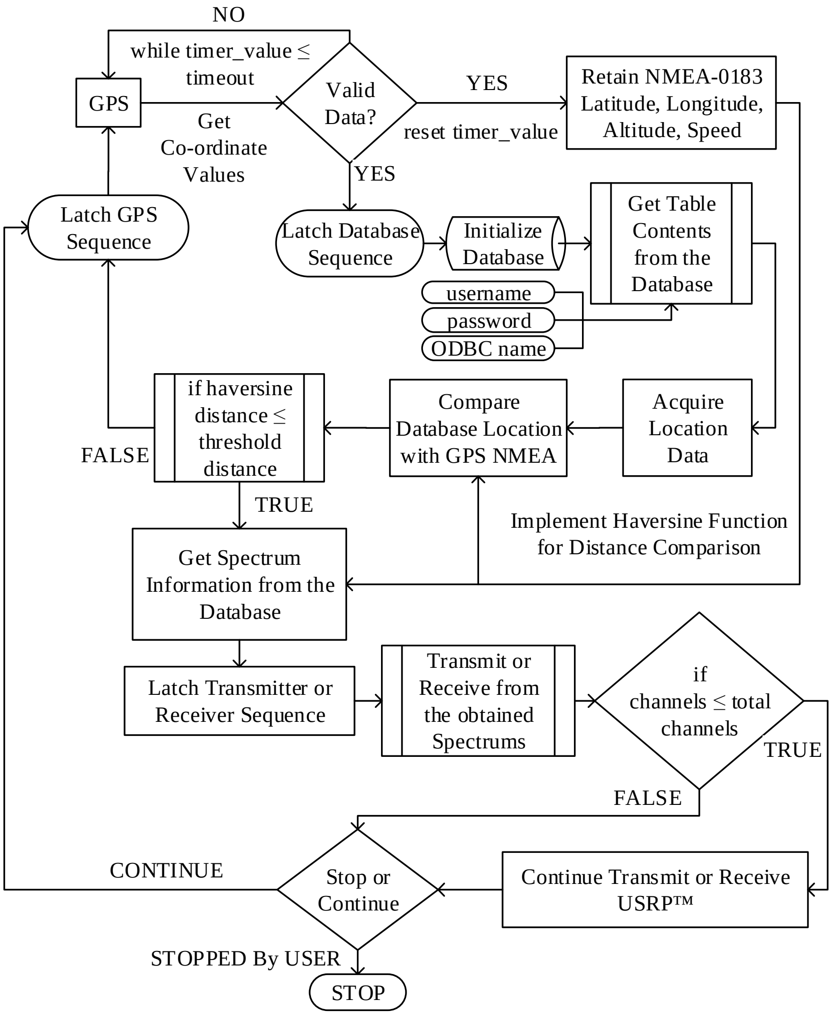

NO
while timer_value ≤ timeout
GPS
Valid Data?
Get Co-ordinate Values
YES
reset timer_value
Retain NMEA-0183 Latitude, Longitude, Altitude, Speed
YES
Latch GPS Sequence
Latch Database Sequence
Initialize Database
Get Table Contents from the Database
username
password
ODBC name
if haversine distance ≤ threshold distance
FALSE
Compare Database Location with GPS NMEA
Acquire Location Data
TRUE
Implement Haversine Function for Distance Comparison
Get Spectrum Information from the Database
Latch Transmitter or Receiver Sequence
Transmit or Receive from the obtained Spectrums
if channels ≤ total channels
TRUE
FALSE
CONTINUE
Stop or Continue
Continue Transmit or Receive USRP™
STOPPED By USER
STOP

Figure 5.9: Opportunistic spectrum access for SUs flowchart diagram

strength at that particular frequency. From the table, the SU_{Tx} tries to transmit at specific frequency by applying quorum-based channel selection. Similarly, SU_{Rx} use the same table from remotely located cloud-based database to look for available spectrum. Once SU_{Rx} acquires all the channels, it also uses quorum-based common channel selection scheme to establish communication.

The detail of quorum-based rendezvous for common channel selection among D2D communication using opportunistic spectrum access in ROAR architecture is discussed

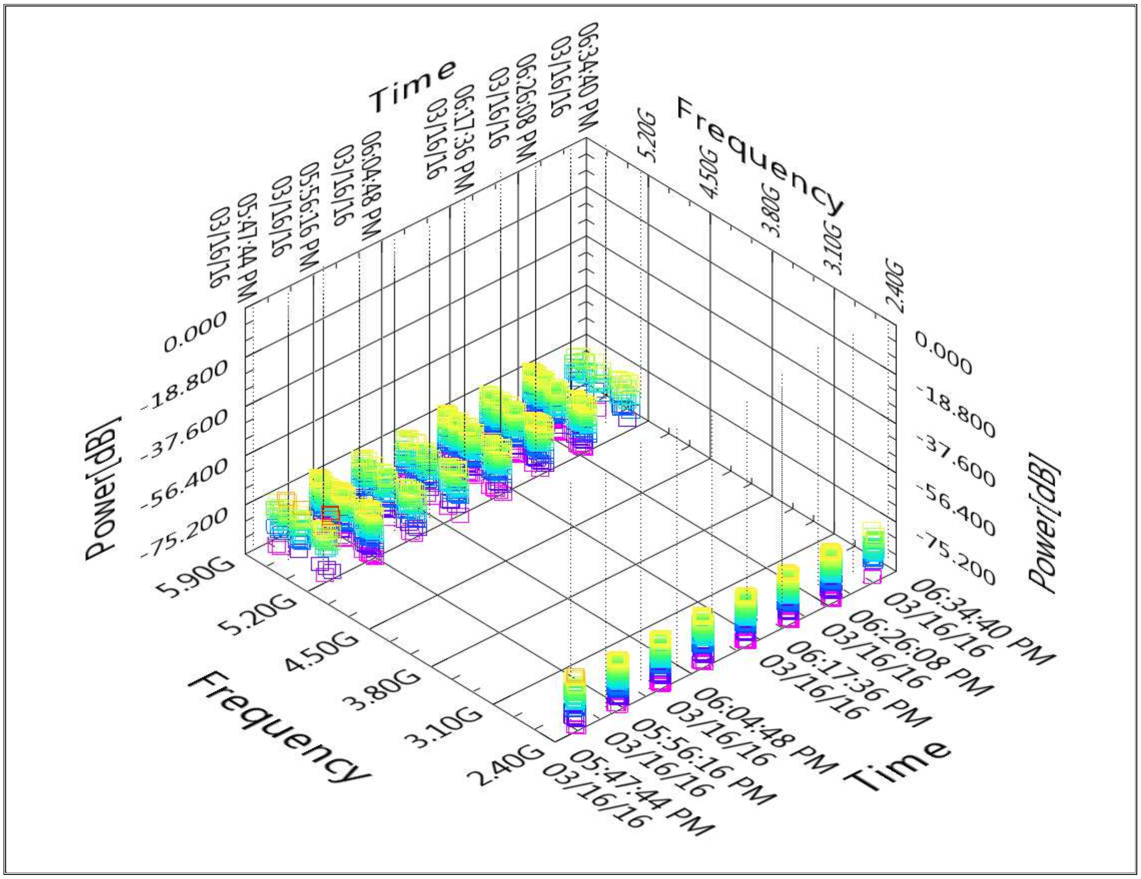
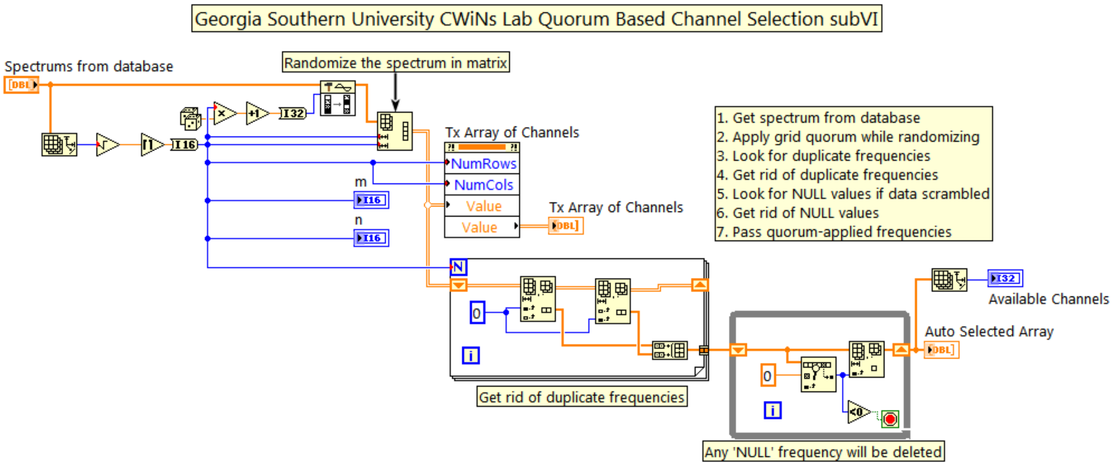


Figure 5.10: List of idle spectrum acquired from cloud-based database by the SU_{Tx} for opportunistic spectrum access represented in 3D graph

below.

Quorum-based Rendezvous for Opportunistic Spectrum Access in D2D communication:

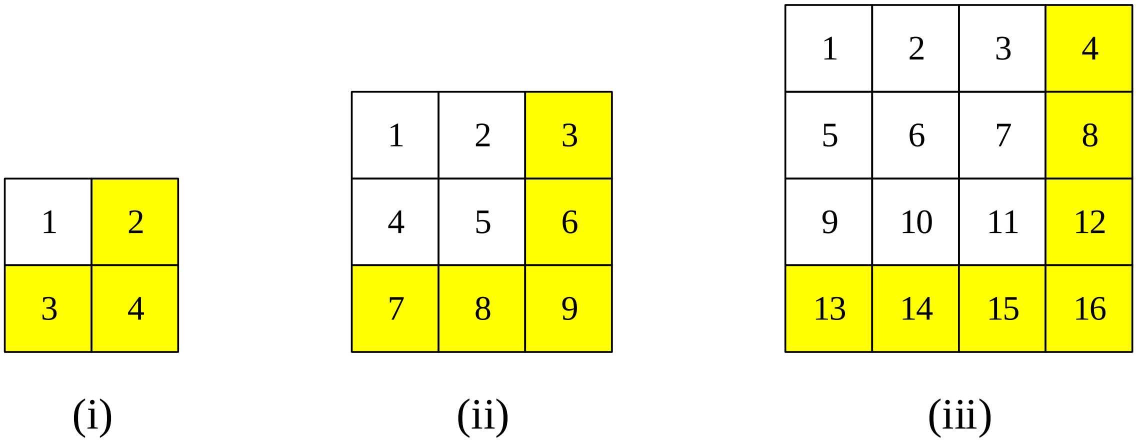
A custom built function block program is created in LabVIEW to serve for quorum-based channel selection as shown in Figure 5.11. A complete system implementation of the quorum algorithm is published in [26], "*A Testbed Using USRP*^{TM} *and LabVIEW*$^{®}$ *for Dynamic Spectrum Access in Cognitive Radio Networks*" on ©IEEE by the author of this thesis implementing a different legal grid method [39] to reduce the search space for idle channel for ROAR D2D opportunistic spectrum access for the SUs.


Georgia Southern University CWiNs Lab Quorum Based Channel Selection subVI
Spectrums from database
DBL
Randomize the spectrum in matrix
I32
I16
Tx Array of Channels
NumRows
NumCols
Value
Value
m
I16
n
I16
Tx Array of Channels
DBL
N
0
i
Get rid of duplicate frequencies
I32
Available Channels
Auto Selected Array
DBL
0
i
Any 'NULL' frequency will be deleted

1. Get spectrum from database
2. Apply grid quorum while randomizing
3. Look for duplicate frequencies
4. Get rid of duplicate frequencies
5. Look for NULL values if data scrambled
6. Get rid of NULL values
7. Pass quorum-applied frequencies

Figure 5.11: Quorum function block programming in LabVIEW

SU gets information about number of idle channels N and creates a random square matrix $m \times m$ where $N = m \times m$. If N is not equal to $m \times m$, SU creates a matrix by repeating the channels from begging to make $N = m \times m$ that can fit in the grid. Similarly, $n \times n$ dimension matrix is generated for the receiver. Note that $m \times m$ can be equal to $n \times n$.



1	2
3	4

(i)

1	2	3
4	5	6
7	8	9

(ii)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(iii)

Figure 5.12: Matrices of idle channels for Legal Grid based channel selection for SUs

For example, to demonstrate the working principal of legal grid method, three matrices are generated in Figure 5.12. In the first 2×2 matrix in Figure 5.12 (i), and the second 3×3

matrix in Figure 5.12 (ii), considering last column and last row, the elements $A = \{2,4,3\}$ and $B = \{3,6,9,8,7\}$ are extracted and used to find common channel. In this case common channel is $'3'$. Once SU transmitter finds this channel, it waits for its receiver to start communication. If there is no common channel found, transmitter looks for another set by considering second last column and second last row of the matrices. This process is continued until it reaches to first column and first row of the channel or until its find idle common channel for communications.

Similarly, while comparing 3×3 matrix in Figure 5.12 (ii) and 4×4 matrix in Figure 5.12 (iii), considering last row and column of elements, it can be seen that the element $'8'$ is common. For second last column and second last row elements in both of these matrices, elements $A = \{3,7,11,15,9,10,12\}$ and $B = \{2,5,8,4,5,6\}$ are available where there is no single common element. In this case, transmitter and receiver may have to get/sense another set of channels or wait for a common channel. While considering second last column and row of matrix in 5.12 (iii) and last column and last row of the matrix in 5.12 (ii), common channels set as $\{3,7,9\}$ are acquired. When transmitter and receiver have similar channel configuration, SU-transmitter could select one of these channels and wait for SU-receiver to negotiate. SU receiver scans only these three channels instead of all channels as in sequential method. Thus, advantage of using quorum-based channel selection is to reduce the search space and find the common channel quickly for communications.

Therefore, quorum-based common channel selection not just helps in faster channel access to preserve resources such as power and computational capacity, but also provides secured connection and hard-to-guess frequency hopping schemes similar to what is discussed in [40]. When SUs in ROAR architecture detect jamming attacks, instead of hopping sequentially to next available channel in sequential manner, they could implement quorum-based channel selection scheme to randomly select available channels and meet in “blind-date” fashion to establish communication. If both SU_{Tx} and SU_{Rx} implement “blind-date" schemes, it gets hard to predict the common channel. Moreover, with higher number of available channels, complexity to guess the channel increases and prevents integrity and reliability of the D2D communication.

The SUs that are initially programmed to communicate using sequential common channel selection are relatively slower. In sequential common channel selection scheme, the SUs search for common channel from the list of acquired spectrum ranging from 2.412GHz and sequentially increments the index till the last spectrum 5.825GHz is found. While sequential channel selection makes sure that all the spectrums are covered, it is a very slow process. Moreover, there are instances where the SU keeps fixed connection and are prone to jammers as the spectrum hopping in sequential common channel selection is easier to predict.

In the Figure 5.13, the channel number index represents corresponding channels in 2.4GHz spectrum. Therefore, channel 8 represents 2.447GHz, 3 represents 2.422GHz, 7 represents 2.442GHz and so on and so forth. From the Figure 5.13 for 100k sampling, the SUs are not able to establish communnication for 4.13 seconds when they are using sequential channel selection. Similarly, communication is possible only after 4.64 seconds for SUs when communication link is established in channel 9 at 2.452GHz for sequential common channel selection [26].

As compared to the sequential hops to find common channels, quorum-based channel selection is faster. For the equal number of sampling, while finding common channel, quorum-based channel selection is quicker. For establishing communication in channel 8 for the first time, the quorum-based common channel gives instantaneous channel selection within 0.53 seconds. Similarly, for channel 9 where the sequential channel selection takes 4.64 seconds, the quorum-based approach takes only 2.12 seconds to establish communication link.

While using quorum-based common channel selection for opportunistic spectrum ac-

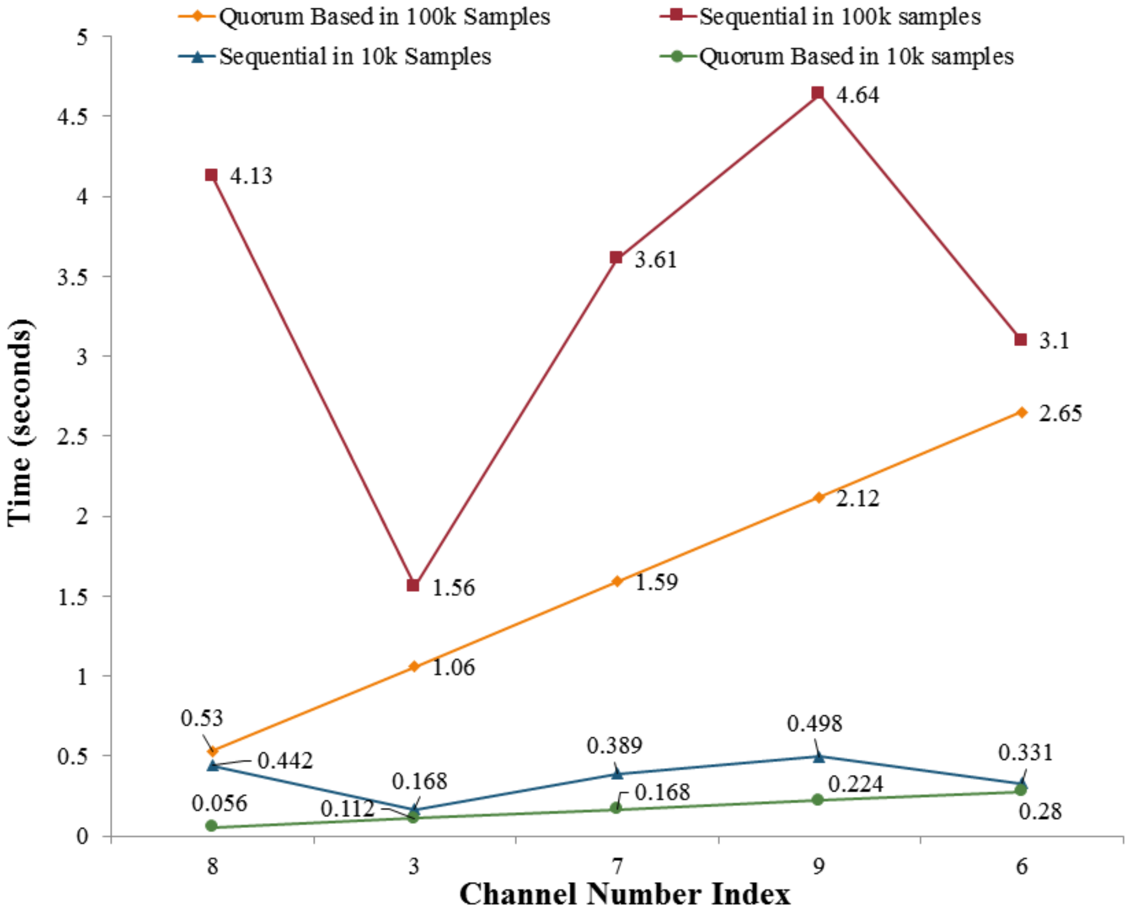


Figure 5.13: Comparison between sequential vs. quorum-based common channel selection in SU at different sampling

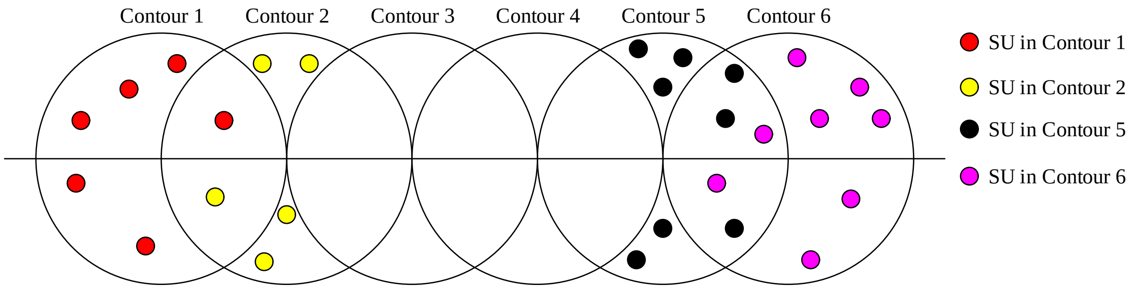
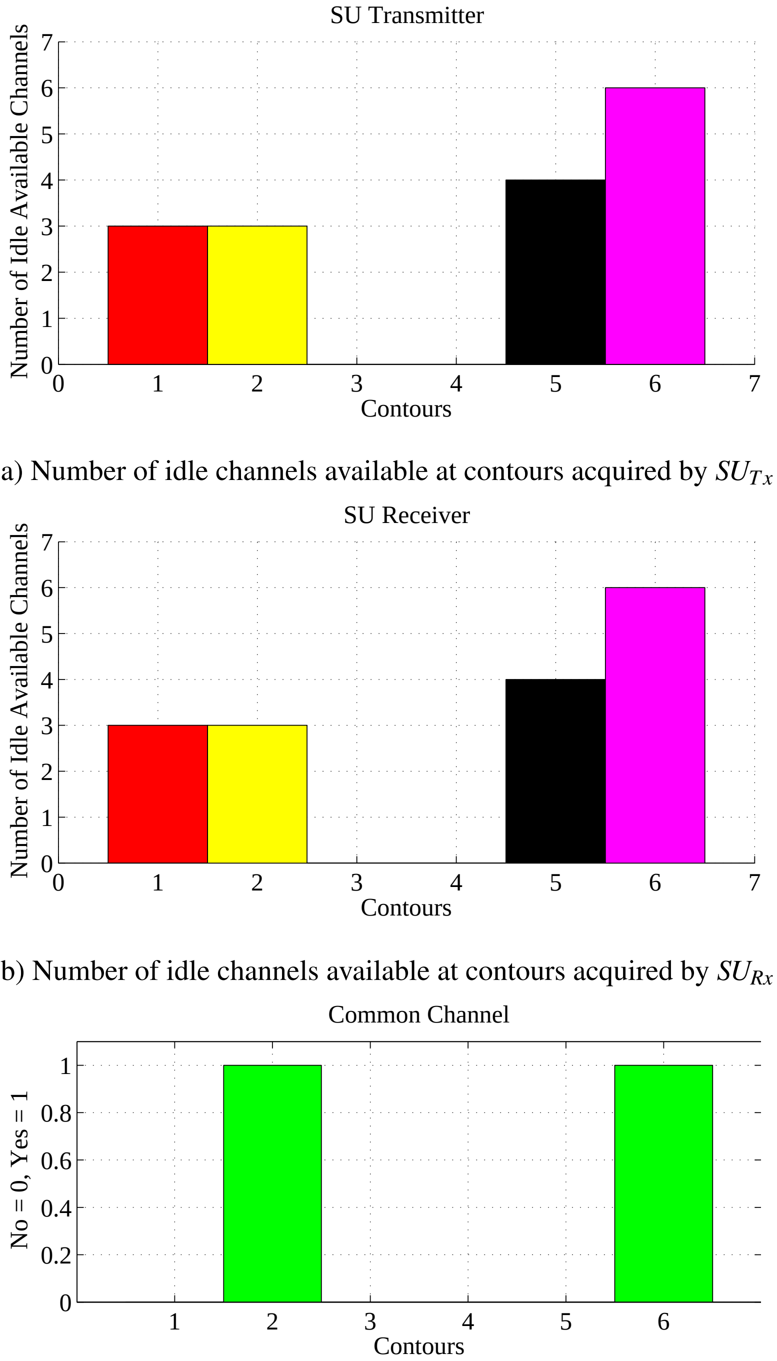


Figure 5.14: Different SUs communicating opportunistically in different contours

cess, the SUs can only communicate if they are able to find common channels. If SUs are in the same contour as shown in the Figure 5.14, they are able to receive the list of available channels for that specific area. Implementing quorum-based common channel selection, they are able to find common channels. Since all channels might not be available to communicate, the SUs resort to using the common channels to communicate. In the figure, it is seen that there are no users in contours 3 and 4.

Based on the contours and the number of idle channels available, the results in Figure 5.15 show that in first contour there are only 3 idle channels. Similarly, there are only 3 idle channels in contour 2. In contour 3 and 4 there are no idle channels available. Based on the information, the SU_{Tx} located at each contour get list of available channels for that contour to communicate. Similarly, the SU_{Rx} located at each contour do the same. Then these SUs rely on quorum-based rendezvous for common channel selection and find corresponding channels to communicate. From the result shown in Figure 5.15(c), it is clear that the SUs located at contour 2 and 6 are only able to establish communication instantly. Here it is assumed that the SUs are located at the fix location and are not moving. If no idle common channels are found, the SUs must either wait long enough to acquire one, or they must move to another location.

Like this way, the quorum-based common channel selection scheme for D2D communication in ROAR architecture is used for opportunistic spectrum access. Since quorum-based channel selection uses randomly selected channels for SU_{Tx} and SU_{Rx} when they hop to various frequencies, jamming attacks are not very effective against quorum-based D2D communication using opportunistic spectrum access. Thus, the transceiver is able to efficiently and effectively establish communication and the opportunistic spectrum access for ROAR architecture testbed is complete.



(a) Number of idle channels available at contours acquired by SU_{Tx}

(b) Number of idle channels available at contours acquired by SU_{Rx}

(c) Common channels found

Figure 5.15: Opportunistic spectrum access of SU_{Tx} and SU_{Rx} as they find common channel to communicate

5.3 Device-to-Infrastructure (D2I) Communication in Opportunistic Spectrum Access

In Device-to-Infrastructure (D2I) communication using opportunistic spectrum access, the SUs utilize similar method as D2D communication but with infrastructures. The infrastructure could be anything from TV stations to cellular towers or satellite stations. These infrastructure provide services that can be used to establish communication with D2I. It is easier to communicate using D2I communication due to predictibility in the nature of communication with the fixed objects like infrastructures than D2D. D2D could be mobile in nature, but generally D2I are stationary. In D2I communication wider geolocation data is available due to distributed private and public cloud that can provide faster, more reliable access to opportunistic spectrums while balancing communication load efficiently among SUs for ROAR architecture [23].

5.4 Chapter Summary

In this chapter, opportunistic spectrum access is discussed from a perspective of concept to designing for ROAR architecture. As a concept, opportunistic spectrum access is studied in detail with reference to DSA as the SUs search for idle spectrum to communicate without causing harmful interference to the PUs. After the SUs are assigned the available spectrums, they resort to using quorum-based common channel selection. As a concept, quorum is very effective in faster and secure common channel selection as it utilizes randomness in channel assignment and selection.

The ideas assimilated in DSA is implemented in D2D communication where ROAR architecture is utilized in USRP testbeds to create a framework for opportunistic spectrum access. In this testbed, the SUs query the distributed cloud-based database via cloud-assisted computing to acquire the list of available idle channels that can be used for communication. Detail testbed implementation is provided in D2D and also the quorum-based common channel selection scheme is implemented by custom building the program in LabVIEW. The flowchart provides the overall logical implementation in the ROAR testbed that uses opportunistic spectrum access.

Upon retrieving the list of available spectrums from the database, the testbed implements the quorum-based common channel selection to establish communication. In quorum-based common channel selection, a comparison is made between quorum and sequential channel selection. It is evident that quorum channel selection scheme is safer, faster and easily implemented in the opportunistic spectrum access for SUs in ROAR architecture. The opportunistic spectrum access implementing quorum-based common channel selection is evaluated for various contours and for different available spectrums.

In D2I communication, opportunistic spectrum access can be implemented more effectively due to stationary nature of the infrastructures. In mobile systems D2D communication, the velocity need to be accounted for opportunistic spectrum access. D2I communication implementing opportunistic spectrum access is similar to D2D communication. Therefore, in this chapter design, analysis, implementation and evaluation of ROAR architecture for opportunistic spectrum access is provided that is valid for dynamic communication between SUs in D2D or D2I communication.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion and Discussions

This thesis introduced the concept of DSA in cognitive radio and presented the ROAR architecture for wideband (50MHz to 6GHz) spectrum sensing and opportunistic accessing. ROAR architecture is discussed and a testbed is created for ROAR architecture using USRPs, distributed cloud-assisted database and real-time opportunistic spectrum access for SUs to prevent causing interference to PUs. The testbed is then implemented, tested and evaluated for feasibility and practicality.

ROAR architecture is perceived with system model and ideas are gathered to prototype the testbed. It is intuitive to utilize opportunistic spectrum access in ROAR architecture to prevent interference to the PUs by allowing the SUs to dynamically access the idle channels. Prior to designing any sub-system for ROAR architecture, conceptual study is done in relevant topic to accumulate enough knowledge. A detail study and experimentation is performed on the USRP to test whether it is fit for ROAR architecture or not. Upon experimentation, it is found that USRP meets all the requirements and expectation as a ROAR architecture testbed and can be used for spectrum sensing, reporting and opportunistic spectrum access. The testbest is then created using GPS-units, SDR spectrum sensors and data reporting utilities. The testbed worked collectively with distributed computing and database to report geolocation and spectrum information to cloud-based database. The database is installed in distributed cloud platform where Storm topology and Cassandra database are implemented for real-time access of the idle channels. The ROAR D2D and D2I transceivers are designed and implemented after successfully evaluating the spectrum sensing and idle channel reporting for the ROAR architecture.

The major challenges faced in this research are rapid quorum implementation for opportunistic spectrum access for the SUs in the ROAR architecture. If the SUs transmitters implementing quorum-based common channel selection are broadcasting in quicker succession using various quorum methods than the receivers, the SUs transmitters and receivers pairs will never get to meet and establish communication. Also, if the SUs transmitters are transmitting in various frequencies quicker than the ROAR spectrum sensors sensing the frequencies, it is harder to detect the transmitters spectrum usage. By the time the ROAR sensors report to the database the list of available idle channels, the SUs transmitters are quick to change the center frequencies which resulted in false spectrum reporting due to the fact that the listed spectrum notified as occupied, is indeed free as soon as the transmitter transmits at a different channel of transmission. This issue is resolved by lowering the transmission cycle of the SUs transmitters and by allowing the receivers to tune to transmitters quickly than the next hopping cycle of the transmitters/receivers pairs.

Similarly, when there are fewer idle channels available to implement quorum from the cloud based database, the SUs transmitter and receiver have hard time establishing communication. In order to establish a common channel, the SUs implement quorum approach. Quorum-based channel selection is faster and reliable, however it is observed that for many idle channels and available channels, quorum-based common channel selection is much more efficient.

Nonetheless, the research is successful and the ROAR architecture prototype is created. The spectrum information are sensed by the spectrum sensors for the ROAR architecture. The GPS units used in spectrum sensors for ROAR architecture are tested to be approximately ±10 meters deviated. The collected geolocation coordinates along with the idle spectrums are reported to the distributed database and are used opportunistically by the ROAR transceivers based on their haversine distance.

The obtained idle spectrums are used in quorum-based channel selection scheme to securely and efficiently find common channel to communicate between D2D or D2I in

ROAR architecture using opportunistic spectrum access.

The ROAR architecture is also responsible for safe-keeping of reported spectrums. The reported idle spectrums to the Cassandra database is used for visualization as a heat-map. The heat-map is created by overlaying the heat index “Firebase” on top of the Google maps API which is built into a website created to display the areas where the idle spectrums are available. The spectrum sensors are distributed throughout the different locations and by the help of cloud computing and spectrum repositories, the idle spectrums that are available at different geolocations are sensed by the spectrum sensors for the ROAR architecture and are displayed on the website.

Like this way, the entire ROAR architecture is built from bottom-up and is observed to be highly stable and reliable. Hence, this thesis presented a methodological approach to designing, analyzing, implementing and evaluating real-time opportunistic spectrum access in cloud-based cognitive radio networks.

6.2 Future Work

The research presented in this thesis can be further expanded by implementing Hidden Markov Model (HMM) to predict the spectrum where there are missing spectrum occupancy data [41]. The research presented in this thesis does not have all the available spectrum information for every geolocation data. The HMM model makes it easy to predict the status of spectrum at any given geolocation based on past spectrum availability. Furthermore, mobility factor can be considered where the users are opportunistically searching for idle channels while moving from one geolocation to another. Security in opportunistic spectrum access in ROAR is another potential area to expand on the research.

REFERENCES

[1] U. Madhow, *Introduction to Communication Systems*. Cambridge University Press, 2014.

[2] C. A. Balanis, “Advanced Engineering Electromagnetics,” *Wiley*, 2012.

[3] J. G. Proakis, M. Salehi, N. Zhou, and X. Li, *Communication Systems Engineering*. Prentice Hall New Jersey, 1994, vol. 94.

[4] FCC, “Federal Communications Commission Spectrum Policy Task Force. Report of the Spectrum Efficiency Working Group,” November 2002.

[5] NTIA, *United States Frequency Allocation Chart*, 2003. [Online]. Available: https://www.ntia.doc.gov/files/ntia/publications/2003-allochrt.pdf

[6] A. Osseiran, V. Braun, T. Hidekazu, P. Marsch, H. Schotten, H. Tullberg, M. A. Uusitalo, and M. Schellman, “The Foundation of the Mobile and Wireless Communications System for 2020 and Beyond: Challenges, Enablers and Technology Solutions,” in *Proceedings of 2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, 2013, pp. 1–5.

[7] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, “Chicago spectrum occupancy measurements & analysis and a long-term studies proposal,” in *Proceedings of the first ACM international workshop on Technology and policy for accessing spectrum*, 2006, p. 1.

[8] J. Mitola III and G. Q. Maguire Jr, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[9] J. Mitola III, “Cognitive radio for flexible mobile multimedia communications,” in *Proceedings of 1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC’99)*, 1999, pp. 3–10.

[10] J. Mitola, “Cognitive radio: An integrated agent architecture for software defined radio,” 2000.

[11] J. Mitola III, “Cognitive radio architecture evolution,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 626–641, 2009.

[12] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[13] B. Sadhu and R. Harjani, *Cognitive radio receiver front-ends: RF/analog circuit techniques*. Springer Science & Business Media, 2013, vol. 115.

[14] A. Ginsberg, J. D. Poston, and W. D. Horne, “Toward a cognitive radio architecture: integrating knowledge representation with software defined radio technologies,” in *Proceedings of 2006. MILCOM IEEE Military Communications Conference*, 2006, pp. 1–7.

[15] P. Sutton, J. Lotze, H. Lahlou, S. A. Fahmy, K. Nolan, B. Ozgul, T. W. Rondeau, J. Noguera, and L. E. Doyle, “Iris: an architecture for cognitive radio networking testbeds,” *IEEE Communications Magazine*, vol. 48, no. 9, pp. 114–122, 2010.

[16] M. Mohorcic, M. Smolnikar, and T. Javornik, “Wireless Sensor Network Based Infrastructure for Experimentally Driven Research,” in *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013)*, 2013, pp. 1–5.

[17] A. Puschmann and M. A. Kalil, *Cognitive Radio Experimentation World*, 2009. [Online]. Available: http://www.crew-project.eu/portal/w-ilabt

[18] T. R. Newman, A. He, J. Gaeddert, B. Hilburn, T. Bose, and J. H. Reed, “Virginia Tech Cognitive Radio Network Testbed and Open Source Cognitive Radio Framework,” in *Proceedings of 2009 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops (TridentCom)*, 2009, pp. 1–3.

[19] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, “Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols,” in *2005 IEEE Wireless Communications and Networking Conference*, vol. 3, 2005, pp. 1664–1669.

[20] FCC, “Federal Communications Commission Understanding the FCC Regulations for Low-Power, Non-Licensed Transmitters,” October 1993.

[21] ——, “Federal Communications Commission Second Memorandum Opinion and Order, FCC ET Docket No 10-174,” September 2010.

[22] ——, “Federal Communications Commission Second Report and Order and Memorandum Opinion and Order, FCC ET Docket No 08-260,” November 2008.

[23] D. B. Rawat, S. Shetty, and K. Raza, “Geolocation-aware resource management in cloud computing-based cognitive radio networks,” *International Journal of cloud computing*, vol. 3, no. 3, pp. 267–287, 2014.

[24] ——, “Game theoretic dynamic spectrum access in cloud-based cognitive radio networks,” in *Proceedings of 2014 IEEE International Conference on Cloud Engineering (IC2E)*, 2014, pp. 586–591.

[25] D. B. Rawat, S. Reddy, N. Sharma, B. B. Bista, and S. Shetty, “Cloud-assisted GPS-driven dynamic spectrum access in cognitive radio vehicular networks for transportation cyber physical systems,” in *Proceedings of 2015 IEEE Wireless Communications and Networking Conference (WCNC)*, 2015, pp. 1942–1947.

[26] N. Sharma, D. B. Rawat, B. B. Bista, and S. Shetty, “A Testbed Using USRP(TM) and LabView(R) for Dynamic Spectrum Access in Cognitive Radio Networks,” in *Proceedings of 2015 IEEE 29th International Conference on Advanced Information Networking and Applications (AINA)*. ©IEEE, 2015, pp. 735–740.

[27] D. B. Rawat, “ROAR: An Architecture for Real-time Opportunistic Spectrum Access in Cloud-assisted Cognitive Radio Networks,” in *Proceedings of 13th IEEE Annual Consumer Communications & Networking Conference (CCNC) 9-12 January 2016, Las Vegas, NV, USA*, 2016.

[28] FCC, “Federal Communications Commission Integrated Spectrum Auction System,” April 2005.

[29] D. B. Rawat, S. Shetty, and C. Xin, “Stackelberg-Game-Based Dynamic Spectrum Access in Heterogeneous Wireless Systems,” *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–11, 2014.

[30] National instruments universal software radio peripheral. [Online]. Available: http://www.ni.com/sdr/usrp/

[31] H. Harada, H. Murakami, K. Ishizu, S. Filin, Y. Saito, H. N. Tran, G. Miyamoto, M. Hasegawa, Y. Murata, and S. Kato, “A software defined cognitive radio system: cognitive wireless cloud,” in *Proceedings of 2007 IEEE Global Telecommunications Conference (GLOBECOM’07)*, 2007, pp. 294–299.

[32] ”What is IQ data?”. [Online]. Available: http://www.ni.com/tutorial/4805/en/

[33] D. B. Rawat and G. Yan, “Signal processing techniques for spectrum sensing in cognitive radio systems: Challenges and perspectives,” in *Proceedings of 2009 IEEE First Asian Himalayas International Conference on Internet (AH-ICI 2009.)*, 2009, pp. 1–5.

[34] D. Cabric, A. Tkachenko, and R. W. Brodersen, “Experimental study of spectrum sensing based on energy detection and network cooperation,” in *Proceedings of ACM first international workshop on Technology and policy for accessing spectrum*, 2006, p. 12.

[35] A. Nafkha, M. Naoues, K. Cichon, and A. Kliks, “Experimental spectrum sensing measurements using usrp software radio platform and gnu-radio,” in *Proceedings of 2014 9th IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, 2014, pp. 429–434.

[36] ”Mean Sea Level, GPS, and the Geoid”. [Online]. Available: http://www.esri.com/news/arcuser/0703/geoid1of3.html

[37] M. J. Abdel Rahman, H. Rahbari, and M. Krunz, “Adaptive frequency hopping algorithms for multicast rendezvous in DSA networks,” in *Proceedings of 2012 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, 2012, pp. 517–528.

[38] S. Reddy, R. Grodi, N. Sharma, and D. B. Rawat, “Rendezvous methods for opportunistic spectrum access in cognitive wireless networks: Analysis and implementation,” in *Proceedings of IEEE SoutheastCon 2015*, 2015, pp. 1–2.

[39] S. Romaszko and P. Mähönen, “Quorum-based channel allocation with asymmetric channel view in cognitive radio networks,” in *Proceedings of the 6th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, 2011, pp. 67–74.

[40] M. J. Abdel-Rahman and M. Krunz, “Game-theoretic quorum-based frequency hopping for anti-jamming rendezvous in DSA networks,” in *Proceedings of 2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, 2014, pp. 248–258.

[41] I. A. Akbar and W. H. Tranter, “Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case,” in *Proceedings of 2007 IEEE SoutheastCon*, 2007, pp. 196–201.

Appendix A

APPENDIX

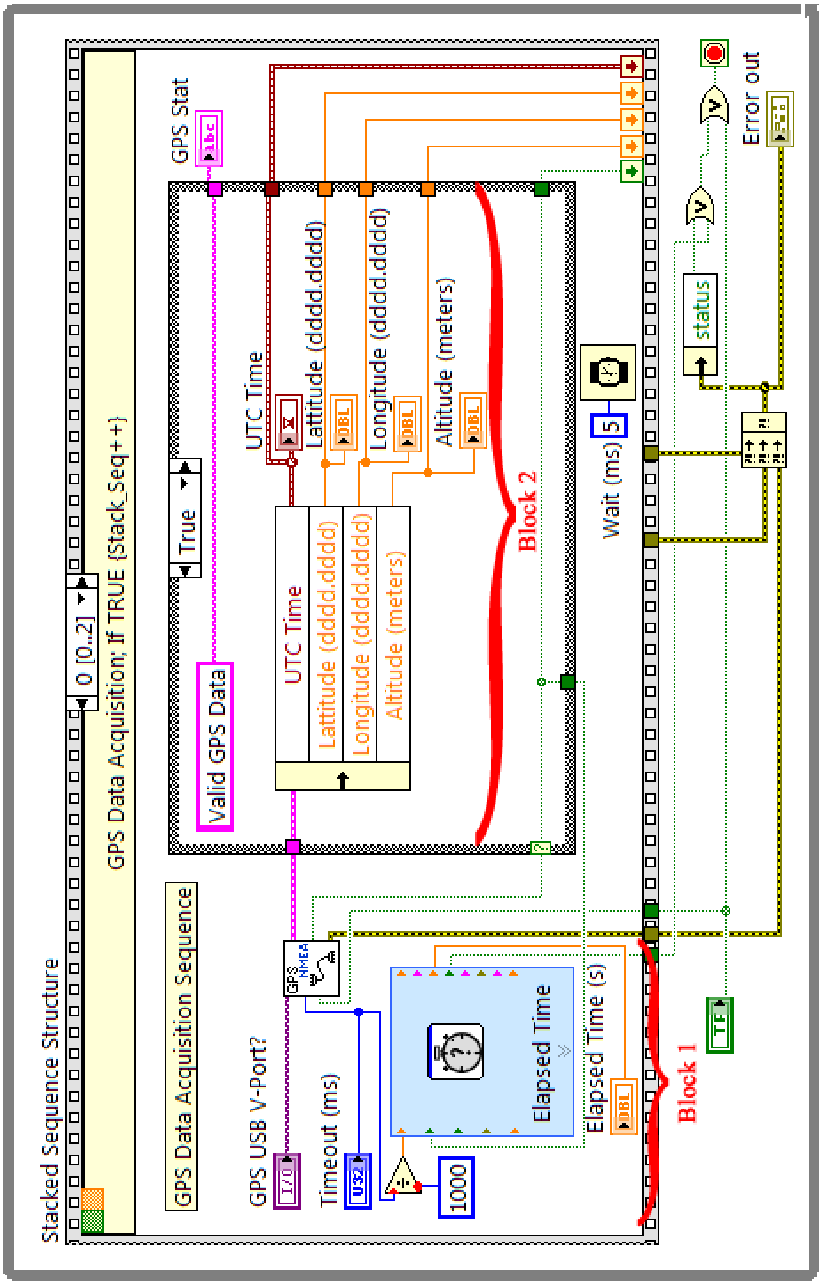


Figure A.1: Function block diagram of GPS location acquisition in LabVIEW

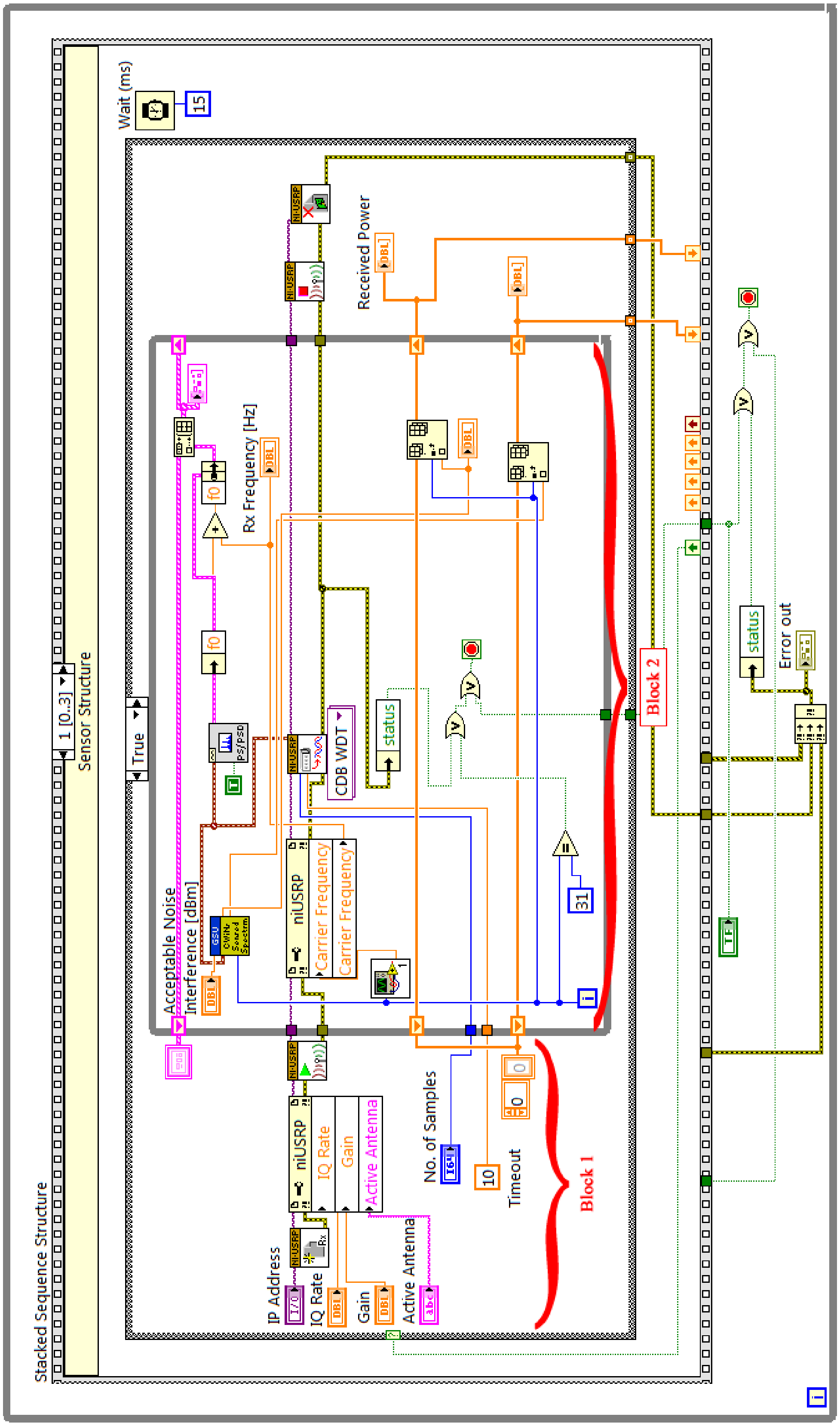


Figure A.2: Function block diagram of sensor spectrum acquisition in LabVIEW

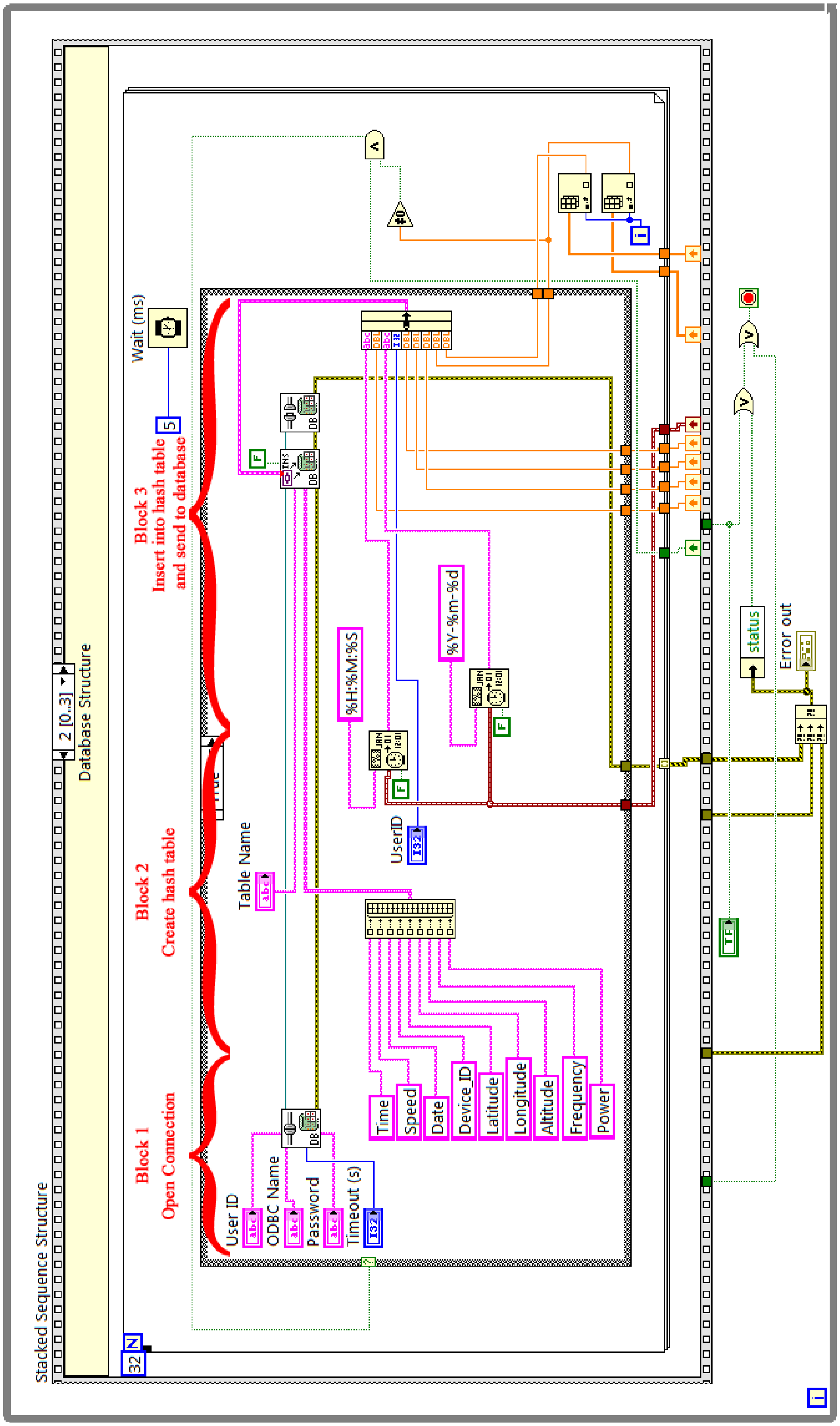
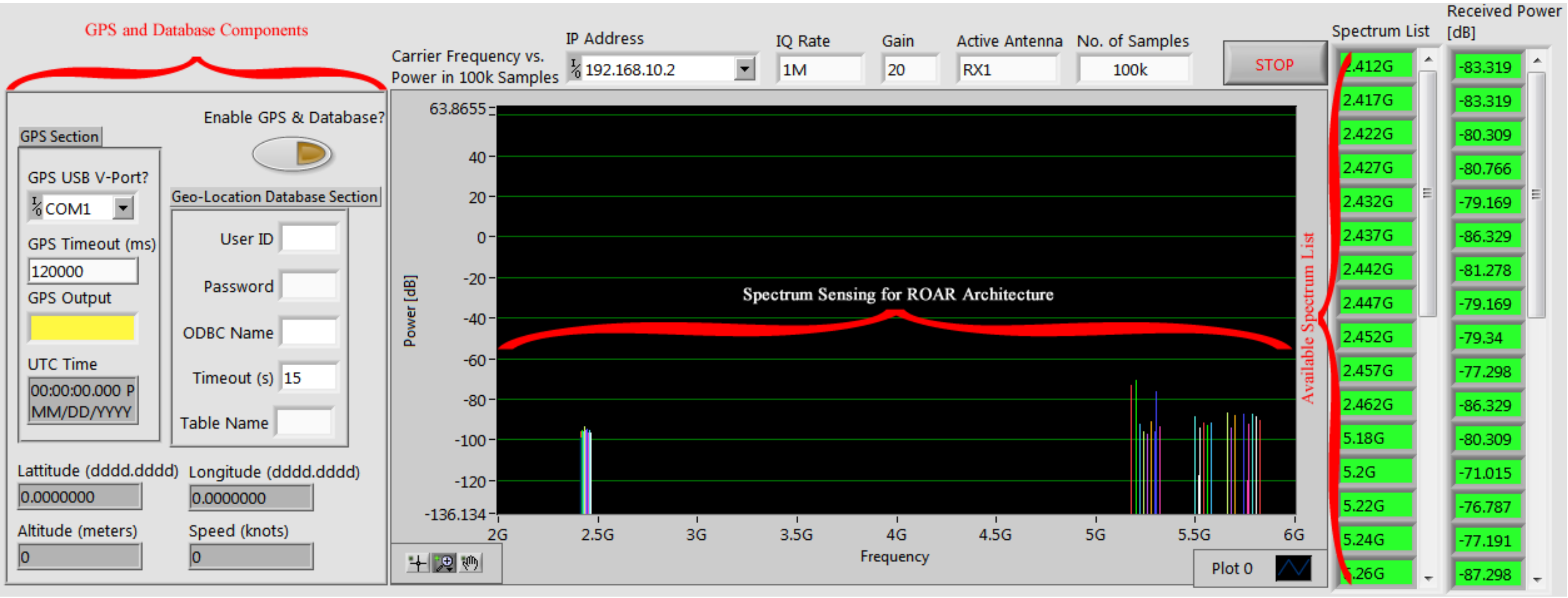


Figure A.3: Function block diagram of database reporting in LabVIEW



Enable GPS & Database?

GPS Section

GPS USB V-Port? COM1

GPS Timeout (ms) 120000

GPS Output

UTC Time 00:00:00.000 P MM/DD/YYYY

Geo-Location Database Section

User ID

Password

ODBC Name

Timeout (s) 15

Table Name

Lattitude (dddd.dddd) 0.0000000

Longitude (dddd.dddd) 0.0000000

Altitude (meters) 0

Speed (knots) 0

Carrier Frequency vs. Power in 100k Samples

IP Address 192.168.10.2

IQ Rate 1M

Gain 20

Active Antenna RX1

No. of Samples 100k

Spectrum List	Received Power [dB]
2.412G	-83.319
2.417G	-83.319
2.422G	-80.309
2.427G	-80.766
2.432G	-79.169
2.437G	-86.329
2.442G	-81.278
2.447G	-79.169
2.452G	-79.34
2.457G	-77.298
2.462G	-86.329
5.18G	-80.309
5.2G	-71.015
5.22G	-76.787
5.24G	-77.191
5.26G	-87.298

Figure A.4: Complete sensor interface in LabVIEW

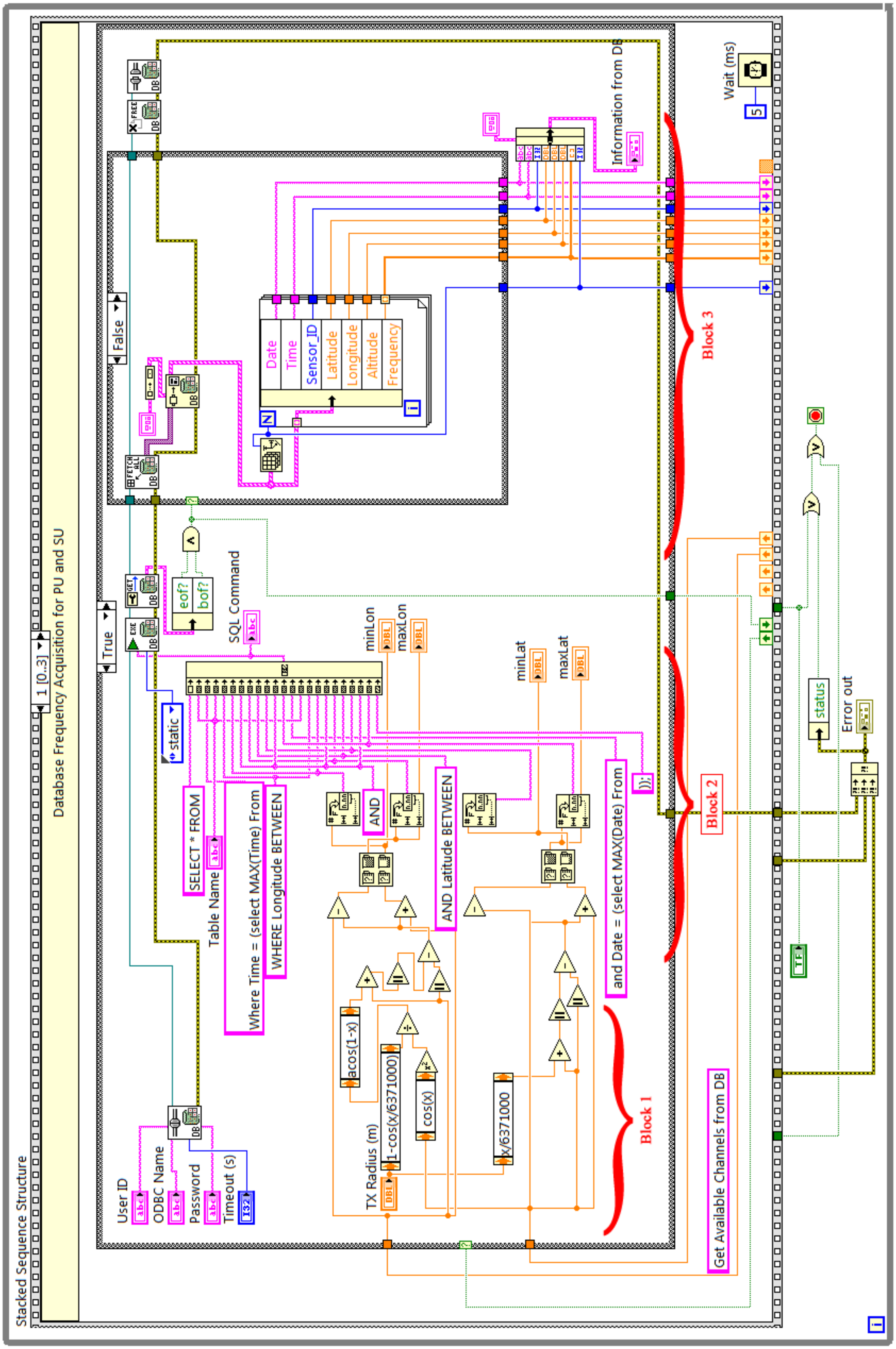
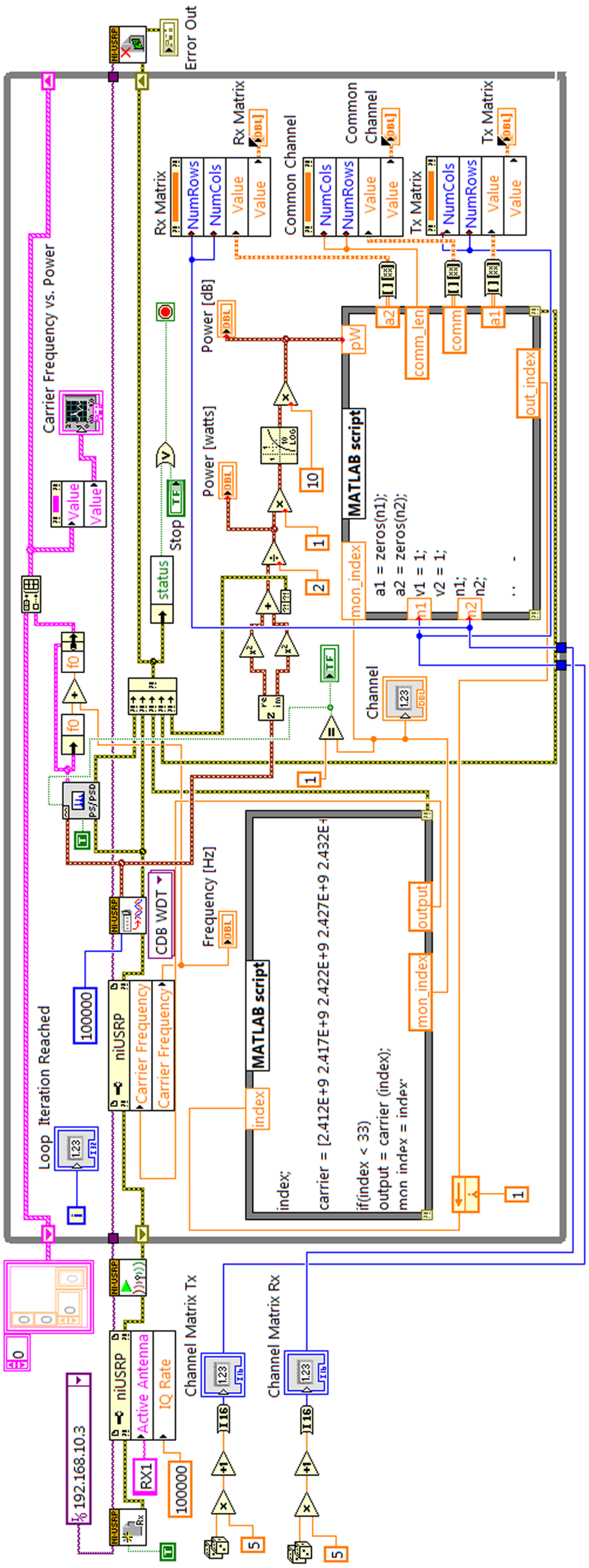


Figure A.5: Function block diagram of transceiver in LabVIEW



```
index;

carrier = [2.412E+9 2.417E+9 2.422E+9 2.427E+9 2.432E+

if(index < 33)
output = carrier (index);
mon index = index
```

```
a1 = zeros(n1);
a2 = zeros(n2);
v1 = 1;
v2 = 1;
n1;
n2;
```

Figure A.6: Function block diagram of quorum implementation with MATLAB in LabVIEW