Georgia Southern University
Digital Commons@Georgia Southern

Electronic Theses and Dissertations | Graduate Studies, Jack N. Averitt College of

Summer 2017

Secure Cloud Controlled Software Defined Radio Network For Bandwidth Allocation

Isaac J. Cushman

Follow this and additional works at: https://digitalcommons.georgiasouthern.edu/etd

Part of the Electrical and Electronics Commons, and the Systems and Communications Commons

Recommended Citation
Cushman, Isaac J., "Secure Cloud Controlled Software Defined Radio Network For Bandwidth Allocation" (2017). *Electronic Theses and Dissertations*. 1644.
https://digitalcommons.georgiasouthern.edu/etd/1644

This thesis (open access) is brought to you for free and open access by the Graduate Studies, Jack N. Averitt College of at Digital Commons@Georgia Southern. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.

SECURE CLOUD CONTROLLED SOFTWARE DEFINED RADIO NETWORK FOR BANDWIDTH ALLOCATION

by

ISAAC J. CUSHMAN

(Under the Direction of Rami J. Haddad)

ABSTRACT

The purpose of this research is to investigate the impact of mobility of wireless devices for opportunistic spectrum access and communications using National Instrument Universal Software Radio Peripherals devices. The overall system utilizes software defined radio networks for frequency allocation, cloud connectivity to maintain up-to-date information, and moving target defense as a security mechanism. Each USRP device sends its geolocation to query the spectrum database for idle channels. The cloud cluster was designed for complex data storage and allocation using a smart load balancer to offer ultra-security to users. This project also explores the advantages of data protection and security through moving target defense. To achieve this, the system would use an array of antennas to split the data into different parts and transmit them across separate antennas. This research provides the design to each of the mentioned projects for the implementation of a fully developed system.

INDEX WORDS: Wireless networks, Software defined radio networks, Cloud computing, Network security

SECURE CLOUD CONTROLLED SOFTWARE DEFINED RADIO NETWORK FOR BANDWIDTH ALLOCATION

by

ISAAC J. CUSHMAN

B.S., Georgia Southern University, 2016

M.S., Georgia Southern University, 2017

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

MASTER OF SCIENCE

STATESBORO, GEORGIA

©2017

ISAAC J. CUSHMAN

All Rights Reserved

SECURE CLOUD CONTROLLED SOFTWARE DEFINED RADIO NETWORK FOR BANDWIDTH ALLOCATION

by

ISAAC J. CUSHMAN

Major Professor: Rami J. Haddad

Committee: Lei Chen

Sungkyun Lim

Electronic Version Approved:

July 2017

DEDICATION

To my mother, your demonstration of perseverance and hard work has been the guiding hand of my ambition to achieve all of my dreams. I will be eternally grateful for your endless support in all of my passions and adventures, from playing hockey to playing the banjo, you have always been by my side. Most importantly, you have taught me the most important aspect about life is simply to be happy with what I have. To my father, your guidance and knowledge paved the way for my intrigue and wonder in the scientific world, without which I would not have found this path. The memory of holding the flashlight while you explained what you were fixing, so I could learn how to do it myself will always remain with me. You truly could build and fix anything and remain an inspiration in my life. Thank you both for all your support along the way.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere and heartfelt gratitude to the guidance given by Dr. Rami Haddad and Dr. Lei Chen. I would also like to thank the members of the ONSmart Lab at Georgia Southern and Mr. Md Baitul Al Sadi for helping in research and design of this project.

TABLE OF CONTENTS

Page

DEDICATION 2

ACKNOWLEDGMENTS 3

LIST OF TABLES 7

LIST OF FIGURES 8

CHAPTER

1 Introduction 11

2 Literature Review 17

2.1 Wireless Spectrum Sensing and Sharing 17

2.2 List of Frequencies 18

2.3 Global Positioning System for Geolocation 18

2.4 Idle Spectrum Database 20

2.5 Inclusion of GPS and Database in the System 20

2.6 Adaptive Threshold Based Joint Energy and Bandwidth Detection Approach 21

3 Bandwidth Allocation using Software Defined Radio Networks 24

3.1 Experimental Setup 24

3.2 LabVIEW and GNU Radio Companion 26

3.3 RF Spectrum Sensing 27

3.4 Detection Scan for B200 GNU Radio 32

3.5 Multiple Frequency Scan 35
3.6 Experimental Results for Threshold Limiting 39
3.7 Adaptive Threshold Evaluation 42
3.8 Geolocation Mapping for Frequencies 47
4 Private Cloud Controller 51
4.1 Multimedia Access in Cloud Computing 54
4.2 Heterogeneity in Cloud Networks 55
4.3 Confidentiality, Integrity and Accessibility 56
4.4 Hybrid Cloud 57
4.5 Private Cloud Infrastructure 59
4.6 Cloud Virtualization Techniques 60
4.7 OpenStack Cloud Design 61
4.7.1 Troubleshooting OpenStack Cloud Design 63
4.8 Resource Management, Allocation and Provisioning 64
4.9 Dynamic Resource Allocation 65
4.10 Smart Load Balancer 66
4.11 Proposed Methodology 69
4.12 Chapter Results 71
5 Adding a Layer of Security in SDRN: Moving Target Defense 74
5.1 Moving Target Defense Configurations 74
5.2 The Attack Process on a Network 76
5.3 Operational Costs 78

5.4 Obfuscation of the Attack Surface 79

5.4.1 Encrypted Key Exchange 81

5.5 Cloud Controller Characteristics 82

5.6 Packet Fragmentation with Frequency Hopping 84

5.7 3x3 MIMO Connection 84

5.8 Chapter Results 86

5.8.1 Experimental Design Specification 86

5.8.2 Frequency Hopping WITHOUT Packet Fragmentation 87

5.8.3 Frequency Hopping WITH Packet Fragmentation 88

6 Conclusion and Future Work 90

6.1 Bandwidth Allocation using Software Defined Radio Networks Conclusion 90

6.2 Private Cloud Cluster Controller Conclusion 91

6.3 Adding a Layer of Security in SDRN: Moving Target Defense Conclusion 92

References 94

LIST OF TABLES

Table Page

4.1 Common OpenStack Services and their corresponding Related Project Name 62

5.1 Empirical Probability of Successful Eavesdropping using Various System Parameters 88

5.2 Empirical Probability of Successful Eavesdropping with/without Fragmentation for 25 Channel System 89

LIST OF FIGURES

Figure		Page
2.1	Scenario which may Result False Available Channels Outside the Circle	20
3.1	Typical RF Spectrum Sensor Unit	25
3.2	Typical experimental setup with USRPs representing a transmitter-receiver Pair Spectrum bands (50 MHz to 5.9 GHz)	25
3.3	The GRC interface is used by connecting the blocks into their respected terminals	26
3.4	Inside the Building	28
3.5	Outside the Building	28
3.6	Spectrum Scan 2.4 GHz: No Channels Active	29
3.7	Spectrum Scan 2.4 GHz: One Channel Active	29
3.8	Spectrum Scan: Two Channels Active	30
3.9	Spectrum Scan 5 GHz: No Channels Active	31
3.10	Spectrum Scan 5 GHz: One Channel Active	31
3.11	Spectrum Scan 5 GHz: Two Channels Active	31
3.12	Experimental scenario with fixed primary location of spectrum sensor with direction of movement	32
3.13	USRP B200 is set to receive signals at 91.7 MHz while corresponding transmitter is turned off	33

3.14 USRP B200 is set to receive signals at 91.7 MHz while corresponding transmitter is turned on 34

3.15 USRP B200 is set to receive signals at 2.412 GHz while corresponding transmitter is turned off 34

3.16 USRP B200 is set to receive signals at 2.412 GHz while corresponding transmitter is turned on 35

3.17 USRP B200 is set to receive signals at 5.3 GHz while corresponding transmitter is turned off 36

3.18 USRP B200 is set to receive signals at 5.3 GHz while corresponding transmitter is turned on 36

3.19 USRP B200 scanning the 2.4 GHz range 10 times 37

3.20 USRP B200 scanning the 5 GHz range 10 times 38

3.21 USRP B200 scanning the DSRC range 10 times 38

3.22 3 USRP B200 scanning results for FM bands 10 times 39

3.23 USRP B200 spectrum sensing the 2.4 GHz range 10 times 40

3.24 USRP B200 spectrum sensing the 5 GHz ISM bands 10 times 40

3.25 Probability of detecting signal vs distance of spectrum sensor from transmitter (meters) where spectrum sensor moved away from transmitter 41

3.26 Probability of misdetection for different threshold values 42

3.27 Probability of misdetection and false alarm vs. the threshold 43

3.28 Sensing period vs. the speed of the sensor with lower and upper limit of time period for given transmission ranges 43

3.29 USRP B200 spectrum sensing the 2.4 GHz range with a threshold of -80 dB 44

3.30 USRP B200 spectrum sensing the 2.4 GHz range with a threshold of -60 dB 45

3.31 The number of occurrences for both misdetections and false alarms in the 24 GHz range 45

3.32 USRP B200 spectrum sensing the 5 GHz range with a threshold of -80 dB 46

3.33 USRP B200 spectrum sensing the 5 GHz range with a threshold of -60 dB 46

3.34 The number of occurrences for both misdetections and false alarms in the 5 GHz range 47

3.35 Experimental set up for Walking Speed Analysis 48

3.36 Results from Walking Experiment 49

3.37 Results from Vehicular Speed Experiment 50

4.1 Third Party Auditor Topology 57

4.2 Cloud in Cloud Model 58

4.3 Cloud Network Using OpenStack based Topology 60

4.4 Proposed Infrastructure includes a Smart Load Balancer and Bandwidth Shaper (SLBBS) and Georgia Southern Secure Private Cloud 70

4.5 Hardware Network Model 72

4.6 Front View of Hardware used for Mobile Cloud Network 73

5.1 Moving Target Defense System with Attacker 75

5.2 Flow Diagram for Transmission of Packet 85

5.3 Structure of the 3x3 MIMO USRP network with Cloud Database Controller 85

5.4 3x3 MIMO USRP Experimental Setup 86

CHAPTER 1

INTRODUCTION

The exponential growth in the availability of lightweight hand-held devices to access wireless networks is one of the primary contributors to the increase in wireless traffic, which leads to severe spectrum shortages for wireless service providers. Opportunistic spectrum access by unlicensed secondary users (SUs) is regarded as one of the emerging techniques to utilize scarce spectrum efficiently (Rawat *et al.*, 2015c). For opportunistic spectrum access, secondary users can sense idle channels and use those idle channels opportunistically. However, there are uncertainties in channel sensing by secondary users as wireless channel suffer by shadowing, multipath, reflection, etc. Recently, spectrum database based opportunistic spectrum access is regarded as an emerging solution where unlicensed secondary users search idle bands by sending their request to the database of idle channels. Note that in database based opportunistic spectrum access secondary users sense the channels and report their sensing results to the database. Furthermore, when they want to access idle channels, they query the database for the currently idle channels. While sensing channel by spectrum sensors, they use signal detection techniques to find idle channels. When energy detection is used to identify primary user/signal in a given channel using fixed threshold, signal energy level could be higher than the threshold but that signal spike could be because of noise. Furthermore, when fixed threshold is used, the signal in low signal to noise ratio region could not be detected or in case of high noise cases may lead to false alarm. There are adaptive threshold based spectrum sensing approaches (Rawat & Yan, 2009a), (Yucek

& Arslan, 2009) where the primary objective of adaptive threshold is to just detect the signal peak by lower or increasing the threshold but not based on the width of the signal energy spectra.

Communication can now be done wirelessly from anywhere and anytime. With advances in robotics and artificial intelligence, engineers are tasked to seamlessly connect all forms of machinery into networks (Rawat *et al.*, 2015d), (Akyildiz *et al.*, 2006), (Rawat *et al.*, 2015c). There are already over 6 billion wireless subscriptions and it is increasing exponentially. This exponential growth results in shortage of wireless spectrum (Akyildiz *et al.*, 2006), (Rawat *et al.*, 2015c), (Mauri *et al.*, 2014). On one hand, all frequency bands are already allocated to wireless service providers for exclusive use for long time and vast geographic area. On the other hand, almost all frequency bands are underutilized or idle most of the time (Akyildiz *et al.*, 2006), (Rawat *et al.*, 2015c), (Haykin, 2005). Thus, static frequency allocation policy has resulted in spectrum scarcity. Dynamic Spectrum Access (DSA) provides a solution to this problem by taking advantage of the ability to be able to sense whether a frequency is busy or idle and accessing the idle frequencies without creating any interference to primary users. To replicate this strategy, it is imperative to obtain a method of identifying idle frequencies. RF spectrum sensing is a method used to find idle channels using different methods (Rawat *et al.*, 2015c), (Haykin, 2005), (Rawat & Yan, 2011), (Rawat & Yan, 2009b). For threshold based energy detection method compares the received signal energy level with the pre-specified threshold and makes the decision whether the channel is idle or not. However, with this method, for low signal strength and high threshold, channel will be detected as idle, while in fact it is not. Thus

adaptive threshold methods can be used as a remedy of this problem so that all channels can be detected correctly. Furthermore, energy detection method may detect channel as active when noise spike is present which is a false alarm.

The performance study of opportunistic spectrum access often overlooks the impact of mobility of unlicensed SU mobility. Many of them assume SUs stationary or with low mobility (Yan *et al.*, 2010), (Rawat *et al.*, 2015b). First, a geolocation database of idle spectrum stored in the cloud is created (Rawat *et al.*, 2015c), (Rawat *et al.*, 2015a), (Rawat *et al.*, 2016). The sensors scan channels within the geographic location that they are in and then report to the database all channels within the spectrum that are not currently being used periodically using energy detection technique (Rawat *et al.*, 2015c), (Yucek & Arslan, 2009), (Sharma & Rawat, 2015). For opportunistic spectrum access, each USRP device sends its geolocation to query the spectrum database for idle channels using dedicated links and gets a list of idle channels. Once SUs get lists of channels, they find common communication channels using quorum based rendezvous approach (Bian *et al.*, 2009). SUs also query the database periodically from their current locations to get updated list of channels if they are within the acceptable range, and establish the link every time for opportunistic communications.

The inclusion of a cloud storage system for dynamic spectrum access, allows for the geolocational data to be stored in large quantities. It also offers the ability to bridge together different sets of localization data that is taken from the sensor network. This enables a user to travel from one city to another while maintaining connection to their current location's data set. In this part of the project, a complete cloud computing cluster is initialized to

handle this data among other sets of data as well. The key concern is the ability to securely store this data while also maintaining the ability to handle large numbers of requests. The key concern with the cloud network is the design and implementation of a load balancer that will effectively handle all the request coming from the sensor network.

Cloud computing is a rapidly growing resource for large data and has spark interest for innovations in the design and security of this new type of big data architecture. One of the major concerns with cloud data storage is that the owners of the information give up the direct control of their information; this leads to significant rise in the need for technology innovations in cyber security. Multimedia applications such as medical images and videos, secure video conferencing, and video streaming consume massive amounts of data requiring significant numbers of servers to provide services for large populations that use them. It becomes increasingly important that the sensitive data offered through cloud resources is to be kept confidential to the respective users without violating confidentiality, integrity and availability of the data. A device which was designed with the sole purpose of making mobile audio phone calls is now the leading basis for functionality in the social world. The types of applications widely vary from audio and video calls, internet browsing, healthcare applications, to mobile games with online connectivity, among many others. These applications have expanded the original idea of what a mobile device could be, however, there have been constant drawbacks to these devices, namely short battery life and limited available storage memory. Other issues that are a current concern with mobile devices with cloud computing is the higher data consumption when on mobile network data. Integrating in a mobile cloud system to allocate and store these applications will allow

for the mobile devices to conserve battery and memory by avoiding large computational processes.

The last major concern that is taken within this project is the addition of layers to security. A software defined radio network has several vulnerable areas to be attacked. The key areas that are considered for protection in this project are the database, transmitters and the receivers. The system to protect the transmitters and receivers uses the concept of moving target defense (MTD). MTD increases the difficulty for an attacker to breach into a network by constantly changing security parameters. The software defined radio network (SDRN) will help the network in MTD to protect sensitive data, so varying the system parameters, causing the attacker to lose their trace. Wireless networks have the advantage of having a very wide attack surface, due to the dynamically changing size of the network. In mission critical systems, loss or theft of data can become a serious problem to the users involved. It is possible to increase the level of security offered by the network through confusing the attacker. The attacker will generally follow an attack process which starts with eavesdropping the network to discover the system configuration. Then, once the attacker becomes part of the network, it will begin sniffing packets for either theft or destructive purposes.

In chapter 2, a literature review of bandwidth allocation is presented. In chapter 3, an evaluation of an adaptive threshold based RF spectrum sensing approach using USRP Software Defined Radio (SDR) for real-time opportunistic spectrum access in cloud based cognitive radio networks (aka ROAR) architecture. The performance of the proposed approach based on probability of misdetection and false alarms is determined. The proposed

approach is particularized to a scenario with energy based detection or bandwidth based detection. The proposed approach is validated through numerical results obtained from both experiments. In chapter 4, an in depth design and implementation for a private cloud controller is presented with a proposed methodology and system design. In chapter 5 a level of security is considered in the form of moving target defense. More specifically utilizing a pseudo-random frequency hopper, controlled by the cloud cluster controller. Chapter 6 concludes the work with overall system design remarks and possible future work that can be added to the system.

CHAPTER 2

LITERATURE REVIEW

2.1 Wireless Spectrum Sensing and Sharing

Spectrum sensing is the process of determining whether a wireless channel is active or idle. It is mainly implemented in dynamic spectrum access in cognitive radio networks which is a potential solution for spectrum scarcity created by static allocation of RF spectrum for exclusive use. The goal of the experiment is to use Universal Software Radio Peripheral, USRP, series 292x and B200 to make the most effective use of allocated frequency spectrum. An adaptive threshold based method is used for spectral width and energy detection to find active channels which should be avoided by the cognitive radio network users. Energy detection based approach may result in high false alarm since it does not consider the width of the signal spectra. Thus to avoid this, both width of the signal spectra and energy level to detect the signal in a given band. Once a cognitive radio identifies the active channels, it avoids those active channels if it chooses to use them for opportunistic communications and/or reports the idle channels to a database for other unlicensed secondary users. This chapter lays out the design and procedures of the Software Defined Radio Network (SDRN) with a wireless sensor network to determine the available spectrum for secondary users.

2.2 List of Frequencies

The Federal Communication Commission (FCC) is the US government agency that regulates all wireless spectrum and regulation associated to them (FCC, n.d.). The USRP B200 GNU Radio can scan frequency bands or channels specified by the FCC. Thus, the channels defined/provided by the IARU (International Amateur Radio Union) region I band plan, which encompasses the Amateur labels provided by the FCC were collected and stored into a file so that B200 GNU radio can read the list and scan the channels. Then, all channels from 50MHz to 6 GHz such as FM bands, ISM 2.4 GHz bands, 5 GHz bands, TV bands, 5.9 GHz DSRC bands, etc. are collected. Note that the USRP B200 GNU radio can scan frequencies from 50 MHz to 6 GHz.

2.3 Global Positioning System for Geolocation

This system model requires the location of the spectrum sensor which becomes the location of the idle channel if it scans the channel. For geolocation, an external Global Positioning Systems (GPS) module, i.e. U-blox 4 GPS module was used. This GPS block provides a simple, manageable method to record the latitude, longitude, altitude, and speed. The sensors scan channels and report idle channels with their geolocation to spectrum database of idle channels periodically using dedicated link such as cellular link. Then, SUs who are seeking opportunistic spectrum access send their geolocation to the cloud controller to find idle channels available for their location periodically. The period is determined based on time lapsed after previous query if they are stationary, or a time estimated based on their

speed if they are moving. If two devices, one receiving and the other transmitting, are within a common channel area and within range of each other, they communicate as long as each stay within the communication ranges.

The Haversine function is used for determining the distance between two geographical locations (location of idle channels and location of the SUs). The Haversine function for longitude and latitude values is given as (Veness, 2002)

$$d = 2 * R \arcsin[\sqrt{\sin(\frac{\phi_2 - \phi_1}{2} + \cos(\phi_1)\cos(\phi_2)\sin^2(\frac{\lambda_2 - \lambda_1}{2}} \tag{2.1}$$

where d is the distance between the two points, R is the radius of the sphere, ϕ_1, ϕ_2: latitude of an idle spectrum and latitude of a SU. λ_1, λ_2: longitude of an idle spectrum and longitude of a SU. The Haversine function was implemented in LabVIEW. This function is used in twice: first when determining the distance the current location is to the database location, this determines whether the user is within acceptable range to receive the list of channels. The second is when creating a timer to check for available channels based on the speed of the SU, the LabVIEW uses the Haversine function to determine the distance of the current location compared to the previous location. It is noted that, in the event that a SU request for active channels at the endpoint of an idle contour, it would then carry those idle channels into an area where they might be restricted to SUs. At high speed and short distances this is a very serious problem to be considered as illustrated in Figure 2.1.

This problem can be reduced by frequently querying the database using dedicated radio

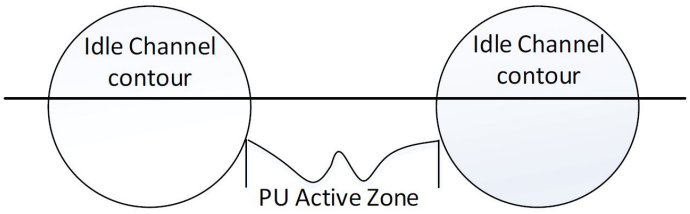

Figure 2.1: Scenario which may Result False Available Channels Outside the Circle

and link. The optimal query interval is determined by

$$t = min(\frac{r - d}{v}, t_p) \tag{2.2}$$

where r the radius of idle contour and v is the relative velocity of the mobile device and t_p is the default query interval which can be adapted based on its history.

2.4 Idle Spectrum Database

The data that is acquired using RF spectrum sensor can be stored for other secondary users. It is important to be able to share data that has been recorded. The most convenient method of sharing information between devices is creating and updating a database. A database allows users to store, access, and update information. The information that is to be sent to the database include the date, time, latitude, longitude, altitude, speed, frequency and signals energy level.

2.5 Inclusion of GPS and Database in the System

The code was built after the inclusion of the GPS and the database aspects. To provide assurance that the program will not timeout due the crowding of information between scans,

global variables and others measures were put in place.

1. Initialize USRP B200 GNU Radio
2. Read GPS Location
3. Read Wireless Channel/Frequency List from Data File
4. Scan a Given Frequency
5. Display the Spectrum of the Signal for Occupancy Information
6. Report Channel Occupancy Information to the Database

2.6 Adaptive Threshold Based Joint Energy and Bandwidth Detection Approach

The received signal at the sensor could be just noise or noise plus the signal from primary users which is represented using two hypotheses as:

$$r(i) = ni : \qquad H_0 \tag{2.3}$$

$$r(i) = s + ni : \qquad H_1 \tag{2.4}$$

where, *r(i)* is the received signal, *ni* is the zero-mean additive white Gaussian noise, and *s* is the primary signal. Then, the energy of the received signal can be computed as

$$r_E = \frac{1}{N} \sum_{i=1}^{N} [r(i)]^2 \tag{2.5}$$

where N is sensing duration. Then, energy based detection can be used to detect idle channels as

$$r_E > \lambda_E = 0 \qquad if \qquad H_0 \tag{2.6}$$

$$r_E > \lambda_E = 1 \qquad if \qquad H_1 \tag{2.7}$$

where r_E is the received energy amplitude and λ_E is the predefined energy threshold. It is noted that the energy based detection may result in false alarm even if there is just a spike because of noise. Thus, the width of the signal spectrum should be considered while detecting a signal by adapting the threshold. The adaptive threshold valued for the energy can be obtained by using the mean (μ_i) and variance (σ_i) for a signal's FFT as

$$\lambda_E = \mu_i + \alpha\sigma_i \tag{2.8}$$

and the threshold for bandwidth detection can be expressed as

$$\lambda_B = .5 * B \tag{2.9}$$

where B is the bandwidth of the signal. Then, the signal detection using energy and bandwidth can be expressed as:

$$r_E > \lambda_E \qquad AND \qquad s_b > \lambda_B = 0 \qquad if \qquad H_0 \tag{2.10}$$

$$r_E > \lambda_E \qquad AND \qquad s_b > \lambda_B = 1 \qquad if \qquad H_1 \tag{2.11}$$

where *sb* received is the signal bandwidth of the received signal.

An important aspect to the accuracy of a sensor is knowing the probability at what ranges and speeds of a sensor that it will trigger a misdetection or a false alarm. A

misdetection is the result of a sensor not detecting a known active channel to be within the threshold power; a false alarm is the result of the sensor detecting a signal within a threshold amount which is known to be idle. The probabilities of these two events occur based on the energy sensed can be described as follows:

$$P_f = P(r_E > \lambda_E \quad AND \quad s_b > \lambda_B) \quad |H0 \tag{2.12}$$

$$P_m = P(r_E < \lambda_E \quad AND \quad s_b < \lambda_B) \quad |H1 \tag{2.13}$$

where P_f is the probability of a false alarm and P_m the probability of a misdetection.

CHAPTER 3

BANDWIDTH ALLOCATION USING SOFTWARE DEFINED RADIO NETWORKS

This chapter is used to go into further detail about the dynamic spectrum access and bandwidth allocation portion of the project. In this chapter is the proposed methodology and experimental setup followed by the results obtained (Cushman *et al.*, 2016a), (Cushman *et al.*, 2016b).

3.1 Experimental Setup

For experiments, LabVIEW was used to program NI USRP devices (USRP 2920, 2921 and 2932). These USRP devices cover wide bands (50 MHz to 6 GHz) by connecting them together through a MIMO cable. An external GPS unit was used with USRP 2920 and 2921, however the USRP 2932 comes with an internal GPS. The GPS unit helps collect the geolocation of the spectrum sensor (that is the location of idle channels), speed, time stamp, etc. A typical RF spectrum sensor diagram is shown in Figure 3.1

Figure 3.2 shows the experimental setup with one pair of transmitter and receiver, and one RF spectrum sensor using USRPs and LabVIEW. The transmitter-receiver pair were pretended to be the primary users where they were communicating in different channels (FM bands, ISM bands, etc.) using quorum based approach (Sharma *et al.*, 2015). RF spectrum sensor was scanning the channels and reporting to the database periodically. Note that, as mentioned, primary network infrastructures are used to get information where spectrum sensors are not available to sense the channels.

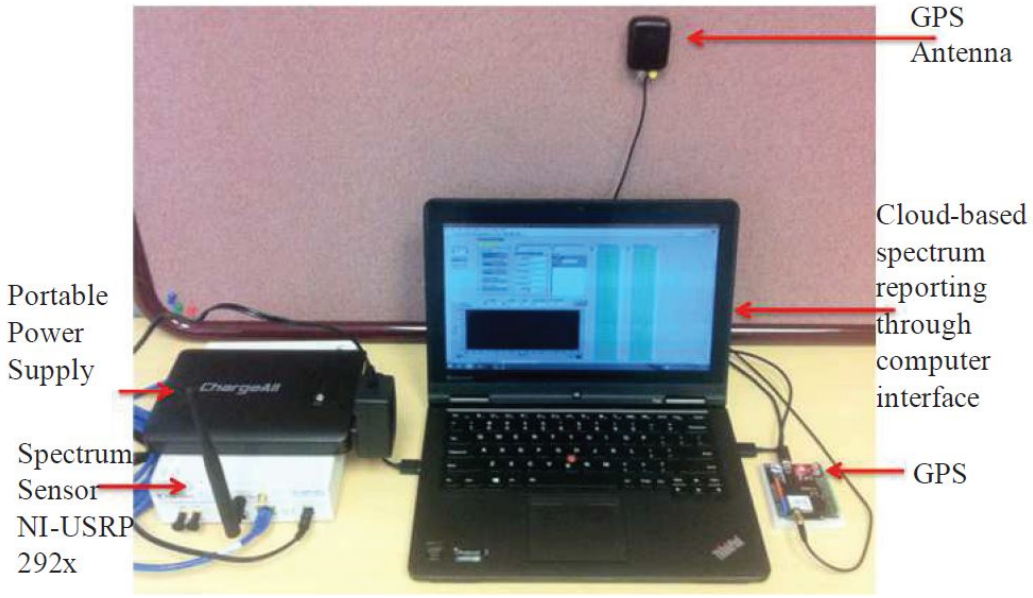

Figure 3.1: Typical RF Spectrum Sensor Unit

Figure 3.2: Typical experimental setup with USRPs representing a transmitter- receiver Pair

Spectrum bands (50 MHz to 5.9 GHz)

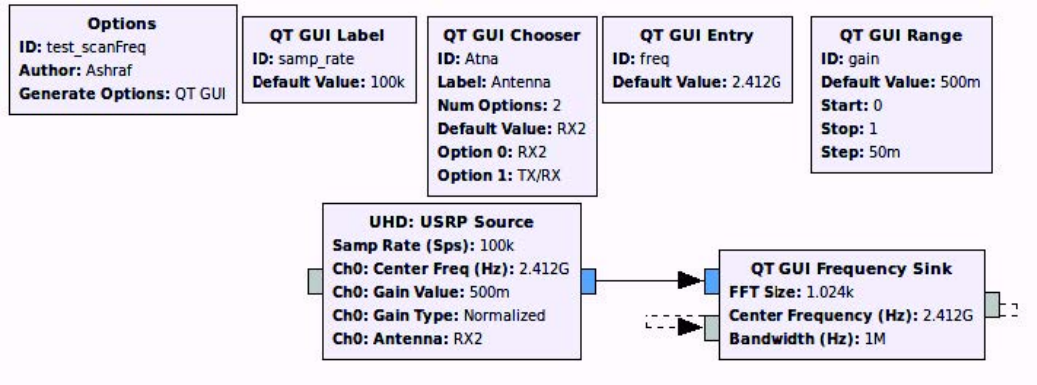

Figure 3.3: The GRC interface is used by connecting the blocks into their respected terminals

Once the geolocation (longitude, latitude, and altitude) of idle spectrum is available in the database, heat map of idle spectrum is generated to visualize the availability of idle channels

3.2 LabVIEW and GNU Radio Companion

Experimental data was taken by designing programs that would be able to interface with National Instruments and Ettus Research devices. GNU Radio Companion (GRC) is a software design program in which functions and methods are represented by blocks, as shown in Figure 3.3, and they connect together to create a program. It was intended to be used in this research but circumstances restricted its effectiveness in the task to scan multiple frequencies. GRC provided an easy to use graphical interface to see the activity of the desired frequency.

To scan channels in the range 50 MHz to 6 GHz, two antennas embedded in B200 GNU radio are used; VERT2450, with dual band 2.4-2.5 GHz and 4.9-5.9 GHz and a VERT400

tri-band antenna.

1. Initialize B200 GNU Radio.

2. Read wireless channel/frequency list from data file.

3. Scan a given frequency

4. Display the spectrum of the signal for occupancy information.

The code is made up of 2 functions. The first is the initialize program for the USRP using B200 GNU Radio, and the last is the main function to scan the channels.

3.3 RF Spectrum Sensing

To demonstrate the spectrum sensing, the first band was the FM Radio frequencies inside the building and outside the building. The scanned results are plotted in Figure 3.4 and Figure 3.5. The spectrum sensor caught frequency 91.9 MHz with high energy inside the building, however outside scan showed a few other FM channels. As expected, with threshold of -105dBm, there were only 3 FM channels active inside the building and over 6 FM channels outside the building as shown in Figure 3.4 and Figure 3.5.

Next, a controlled experimental setup for 2.4 GHz ISM bands (i.e., 2.412 GHz to 2.46 GHz) using USRP devices was prepared. The experiment was designed in three scenarios. Scenario 1: no transmitters transmitting in any channels in 2.4 GHz bands, Scenario 2: one transmitter transmitting in 2.437 GHz (Channel 6) and Scenario 3: two transmitters

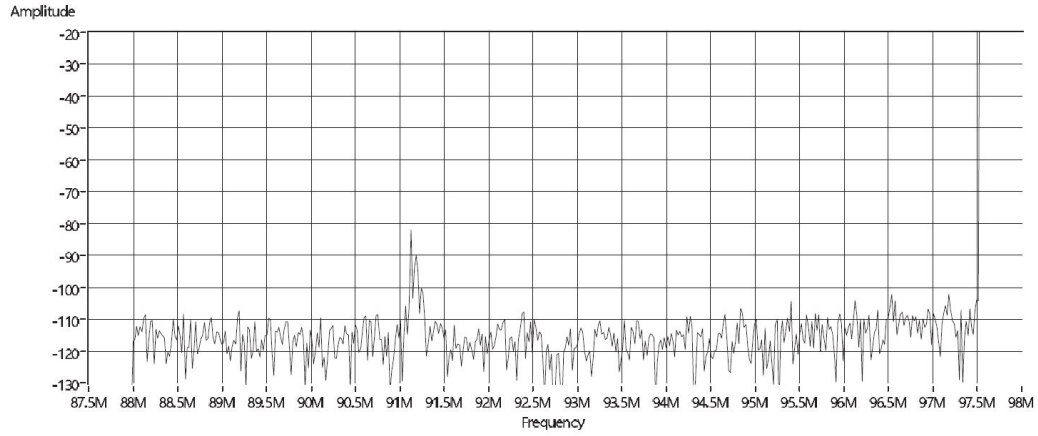

Figure 3.4: Inside the Building

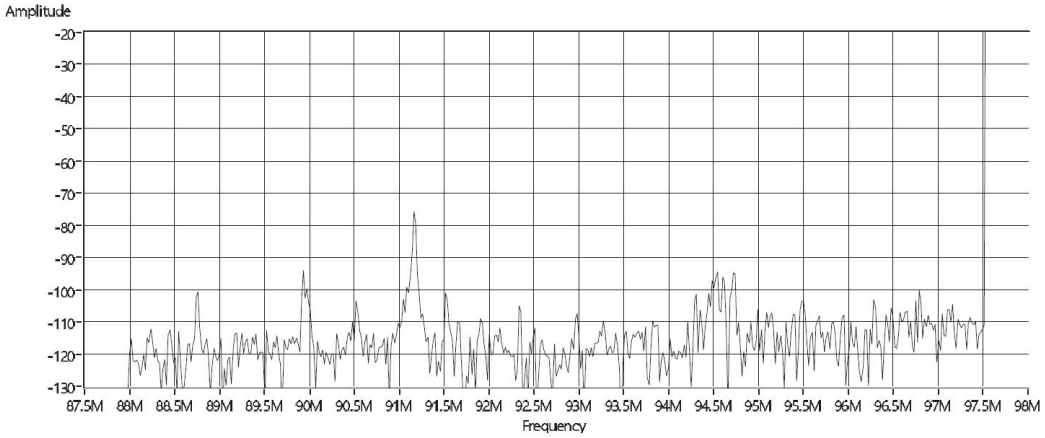

Figure 3.5: Outside the Building

transmitting in two different channels in 2.417 GHz (Channel 2) and 2.457 GHz (Channel 10).

In Figure 3.6, the scanned results of Scenario I for ISM bands and observed a flat spectrum with a couple of spikes with very small bandwidth are shown. When the energy based detection with threshold is -105 dBm, there would be false alarm. In Figure 3.7, the plotted scanned results of Scenario 2 for ISM bands and observed that the 2.437 GHz (Channel 6) is active. Similarly, for Scenario 3 for ISM bands, the scanned results shown

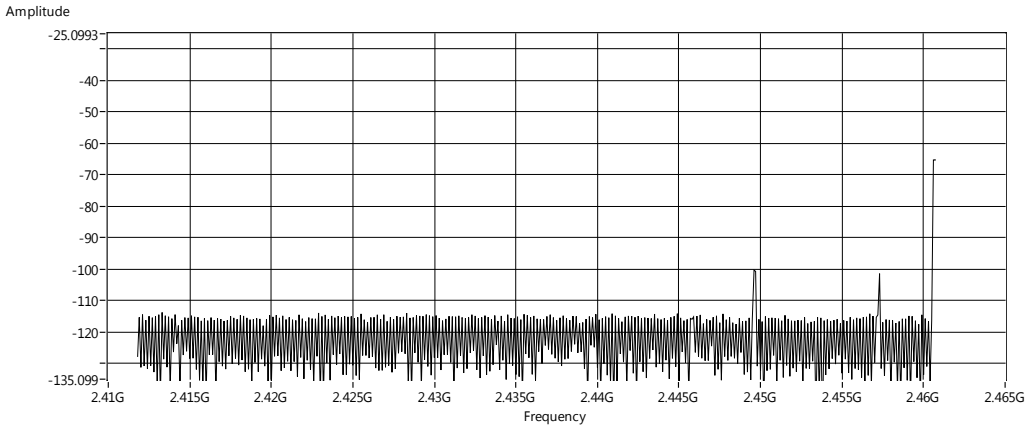

Figure 3.6: Spectrum Scan 2.4 GHz: No Channels Active

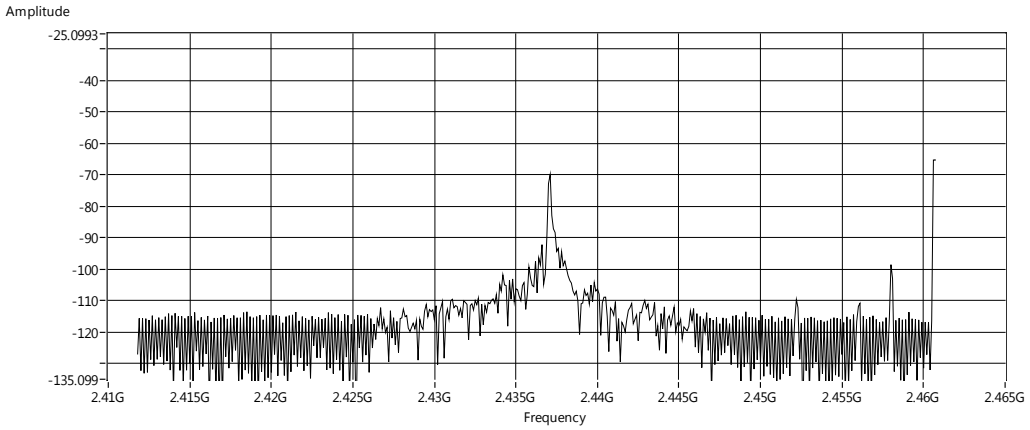

Figure 3.7: Spectrum Scan 2.4 GHz: One Channel Active

in Figure 3.8 show two channels 2.417 GHz (Channel 2) and 2.457 GHz (Channel 10) that were active as expected. Note that the spectrum sensor does not report the active channels to the database, however it reports the idle channels with its geolocation, time stamp, speed, etc. Last, was the experiment to scan 5 GHz ISM bands.

Experiments were then designed and implemented where no channel active, one channel active and two channels active in 5 GHz bands, then scanned the 5 GHz and plotted the respective scanned results as shown in Figures 3.9, 3.10 and 3.11 respectively. Note

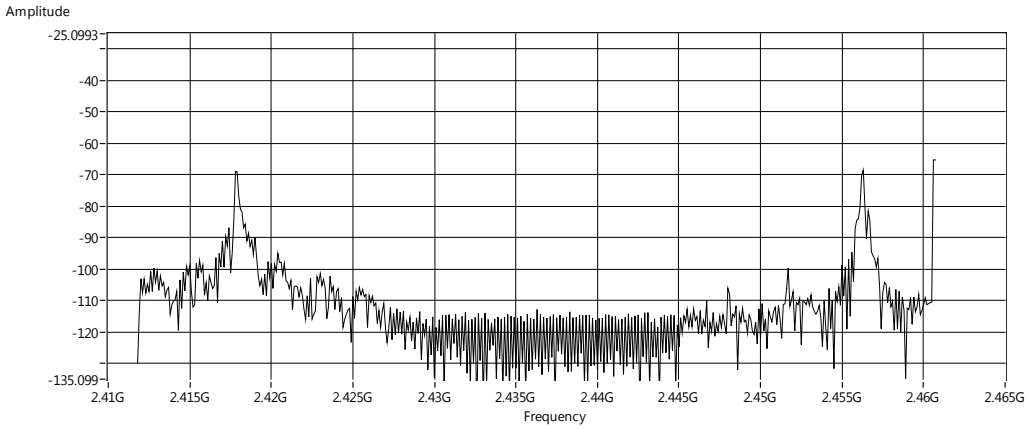

Figure 3.8: Spectrum Scan: Two Channels Active

that in Figure 3.7 there were very small spikes at around 5.2 GHz which were not active channels/signal along with one true signal at 5.5 GHz. However, signal energy detection approach treats the channels with higher energy than threshold (say 90 dBm) as active which is false alarm in this case. However using adaptive threshold and band-width detection, this error was eliminated resulting in desired results (i.e., channel at 5.5 GHZ was active and none others in this case). Similarly in two channels active case in Figure 3.11, there were only two channels were active, however there are two more channels had spikes with small bandwidth.

To see the impact of mobility of RF spectrum sensor, transmission range of primary user and sensing range of the RF spectrum sensor, an experiment was designed where a fixed USRP device had the RF spectrum sensor and the user walked away and took a measurement scan for power every 50 meters as shown in Figure 3.12.

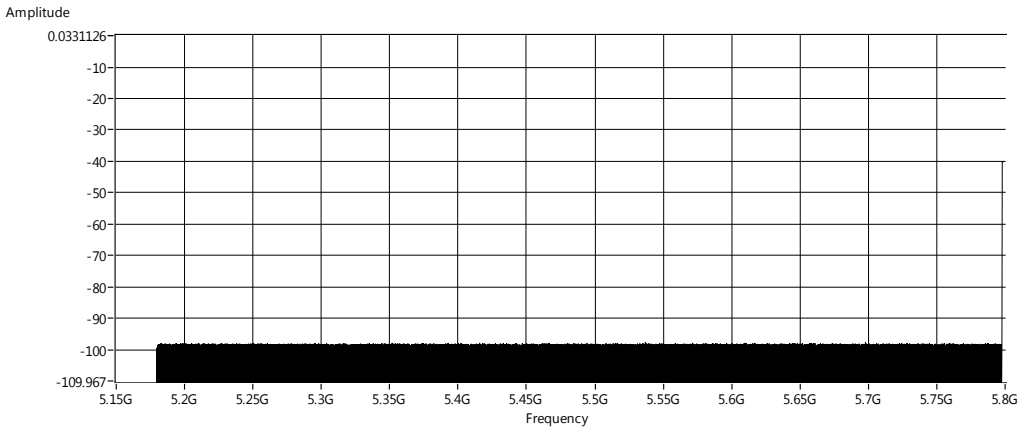

Figure 3.9: Spectrum Scan 5 GHz: No Channels Active

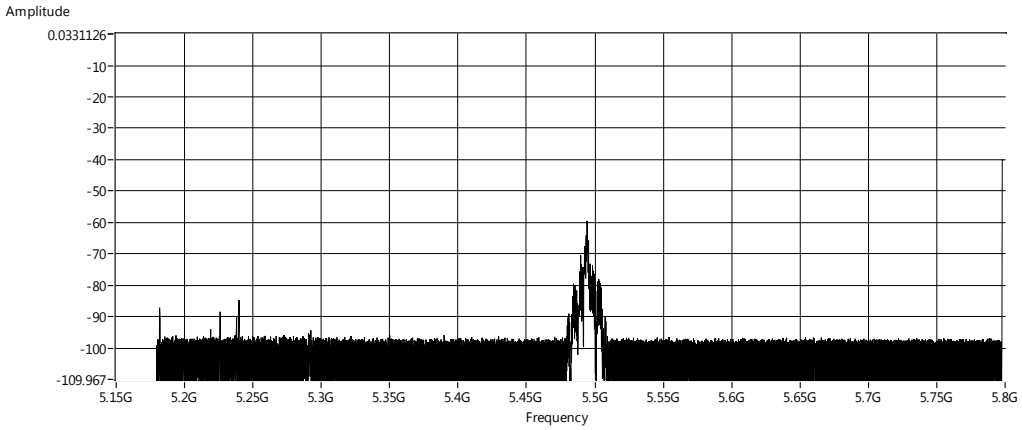

Figure 3.10: Spectrum Scan 5 GHz: One Channel Active

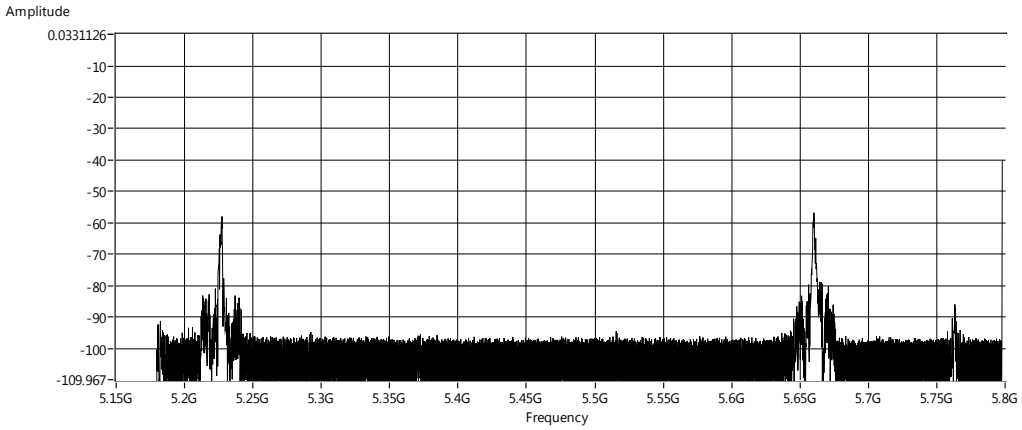

Figure 3.11: Spectrum Scan 5 GHz: Two Channels Active

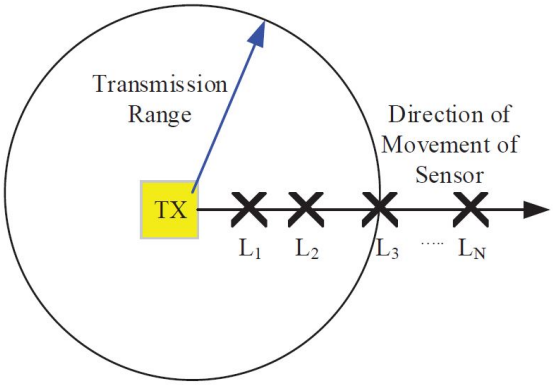

Figure 3.12: Experimental scenario with fixed primary location of spectrum sensor with direction of movement

3.4 Detection Scan for B200 GNU Radio

In this part of the experiment, similar experiments were conducted as in 3.3, except done with a USRP B200. This is done to showcase to variety of software defined radios that can use this system. The first scenario, the GRC frequency graph was generated with no signals transmitted in a list of given channels. In the second scenario, the same graph was generated, while a signal is being transmitted. These scenarios were tested in the FM bands, 2.4 GHz bands and 5 GHz bands. The USRP B200 was configured as a transmitter while a NI USRP 292x devices were configured as a transmitter. A VERT 2450 antenna was used for the 2.4 GHz and 5 GHz channels and a VERT 400 antenna was used for the FM channels. In the first trial, the detection for the FM channels was tested. The USRP B200 was set to receive a signal at 91.7 MHz while the transmitter is off, and the scanned results are plotted on Figure 3.13. It is apparent that the receiver cannot see any distinguishable signals while the transmitter is off, observed by the absence of high power readings, in the

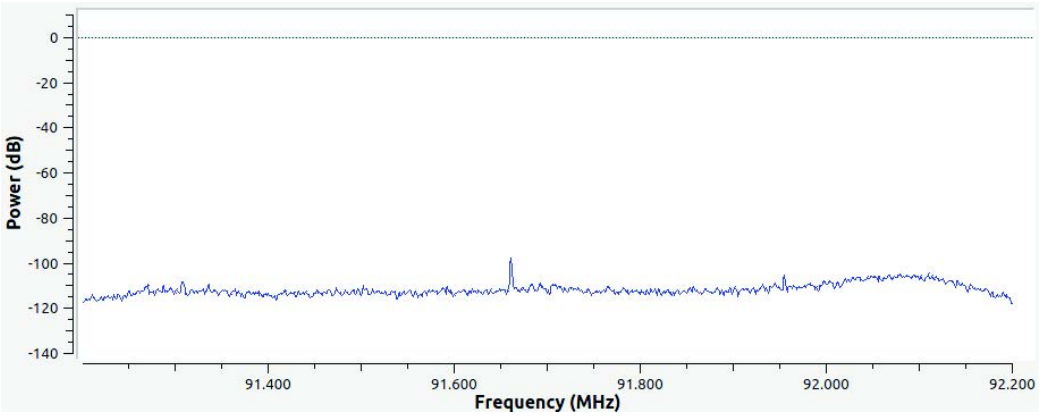

Figure 3.13: USRP B200 is set to receive signals at 91.7 MHz while corresponding transmitter is turned off

first scenario. The readings that are displayed, are the unwanted noise that were discussed in the previous sections which do not have bandwidth greater than the threshold λ_B. It is clear that the USRP B200 was able to detect the signal when the transmitter was turned on as shown in Figure 3.13 where both $r_E > \lambda_E$ AND $bs > \lambda_B$ were satisfied.

By comparing the received signal spectra in Figure 3.13 and Figure 3.14, it can be easily be determined whether the signal can be determined to be active channel and an idle channel. The transmitted signal appears exactly where it is being transmitted on the receiving side and its energy and bandwidth are distinct compared to the unwanted signals.

In the second trial, the receiver and corresponding transmitter are set to frequency 2.412 GHz which is located in the 2.4 GHz ISM bands. This was repeated in two more scenarios in which the transmitter is on or off. As expected, the unwanted signals appear when the USRP B200 scans for the frequency as evident in Figure 3.16. This figure displays the peak that correlates to the transmit frequency. This trend is also demonstrated when the

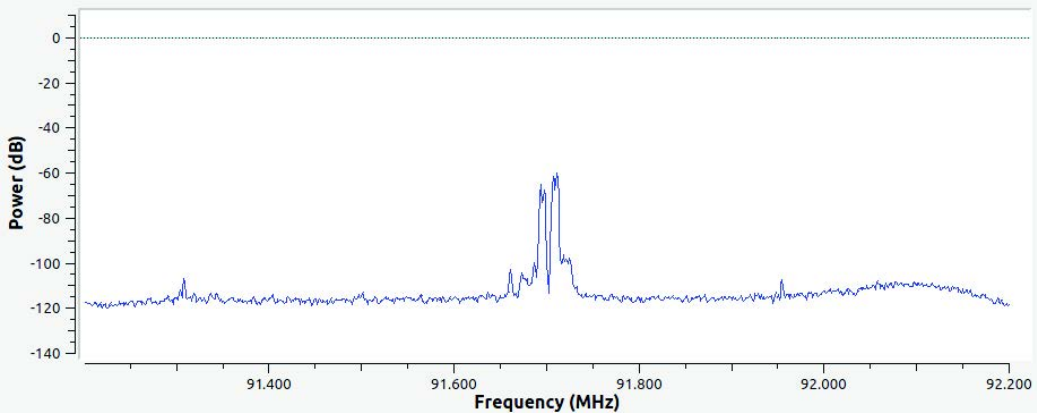

Figure 3.14: USRP B200 is set to receive signals at 91.7 MHz while corresponding transmitter is turned on

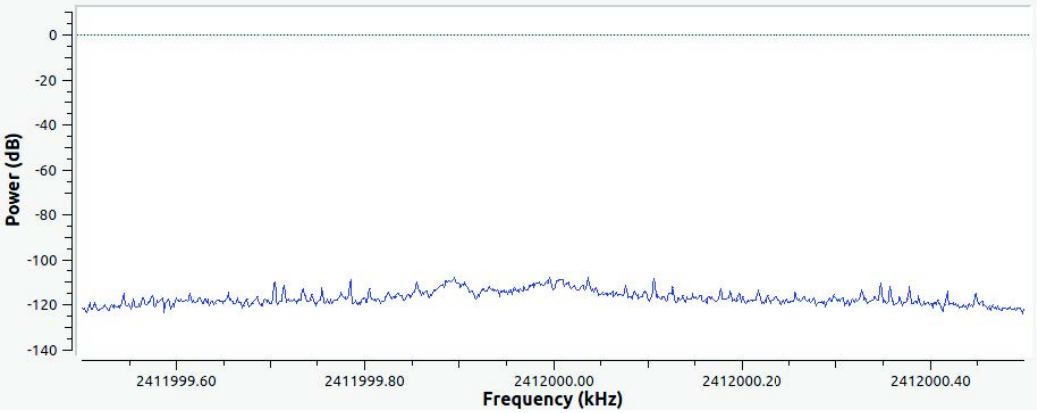

Figure 3.15: USRP B200 is set to receive signals at 2.412 GHz while corresponding transmitter is turned off

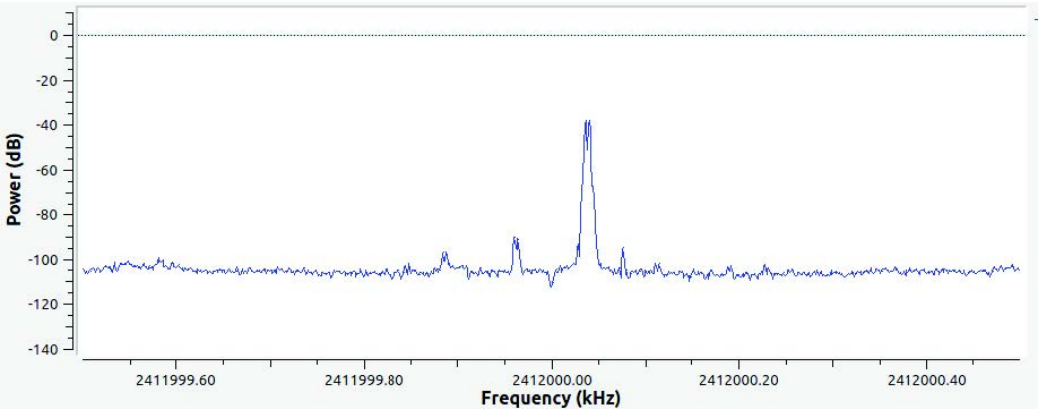

Figure 3.16: USRP B200 is set to receive signals at 2.412 GHz while corresponding transmitter is turned on

USRP B200 scans in the 5 GHz range, shown in Figures 3.17 and 3.18.

By observing the past several figures, it is easy to differentiate between idle and active channels by comparing the signal energy and bandwidth against corresponding threshold values. For example, if the threshold value is set to -60 dB then the active and idle scenarios can be distinguished. The same can be stated with a threshold value of -80dB. It can be concluded that, for these results, any threshold value within the range from -60 dB to - 80dB is sufficient as appropriate markers to determine a channel status as active or idle.

3.5 Multiple Frequency Scan

It is now evident that the USRP B200 GNU radio can detect frequencies in the FM, 2.4 GHz, 5 GHz and DSRC range. In the next experiments, the USRP B200 is tasked to scan the frequencies in the aforementioned ranges. It is expected to have considerable peaks in the readings when signal is present or high noise is present. The energy of received signal

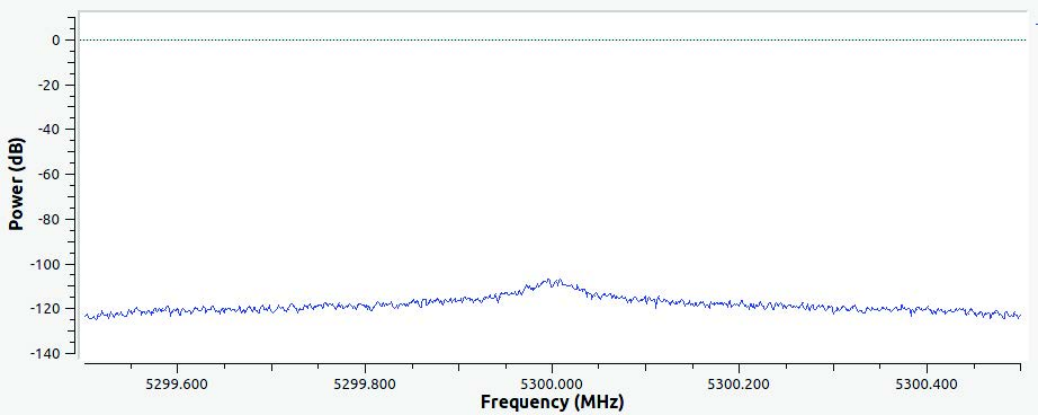

Figure 3.17: USRP B200 is set to receive signals at 5.3 GHz while corresponding transmitter is turned off

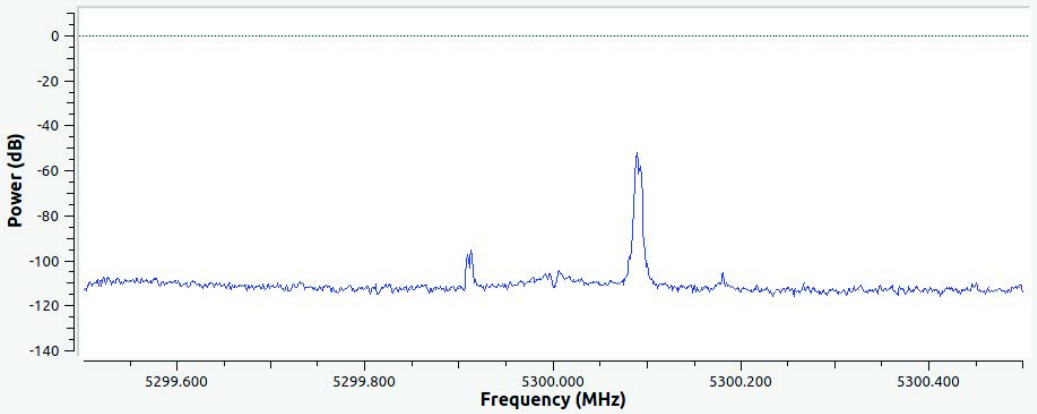

Figure 3.18: USRP B200 is set to receive signals at 5.3 GHz while corresponding transmitter is turned on

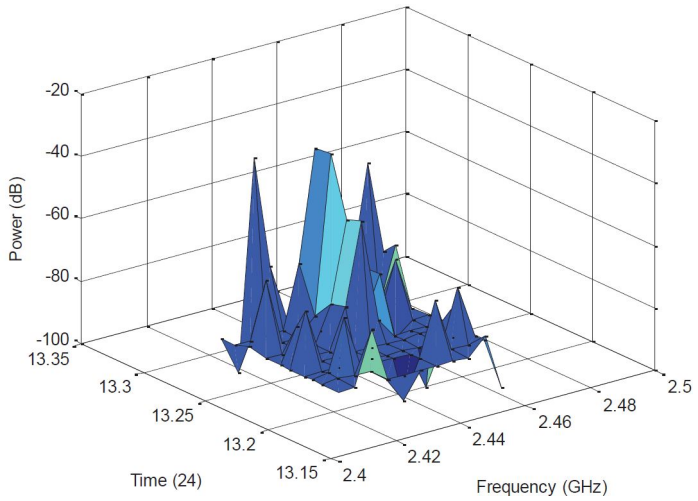

Figure 3.19: USRP B200 scanning the 2.4 GHz range 10 times

by the B200 in 2.4 GHz ISM bands, 5 GHz ISM bands and 5.9 GHz DSRC bands is plotted in Figures 3.19, 3.20 and 3.21 respectively. The DSRC channels show the least amount of activities as there were no transmitters transmitting any signals. Then, the FM bands are scanned and plotted in Figure 3.22.

In the next experiment a system model is created in which four transmitters send out a signal at different times. The test is conducted in the 2.4 and 5 GHz range because those are the ranges where the results were more apparent. The frequencies that are transmitted in the 2.4 GHz range were 2.452, 2.422, 2.437, and 2.417 GHz. The 5 GHz frequencies were 5.3, 5.785, 5.54, and 5.24 GHz. These were the channels that were chosen to be transmitted because they were not in close proximity of each other. The experiment was conducted in a controlled environment where all the transmitters were set within three meters of the receiver and set them at 10 gain. The transmitters were turned on in every combination possible except for them being turned off. Then, 10 trials of scanning each range were

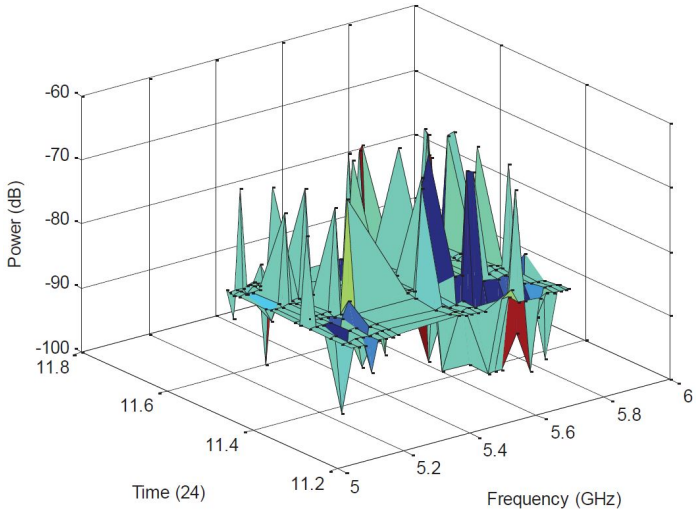

Figure 3.20: USRP B200 scanning the 5 GHz range 10 times

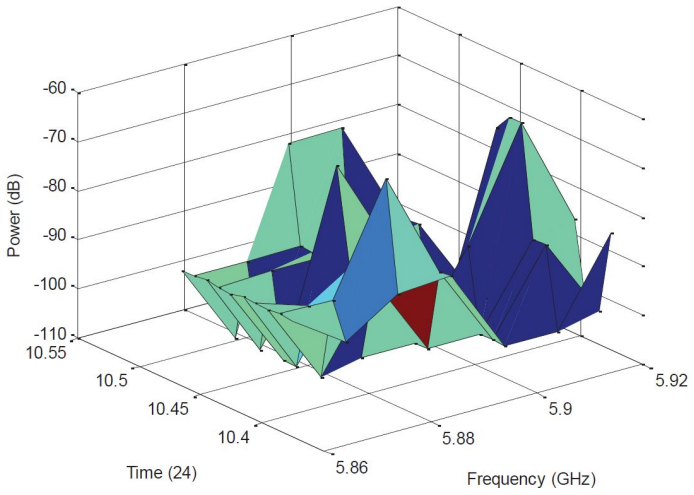

Figure 3.21: USRP B200 scanning the DSRC range 10 times

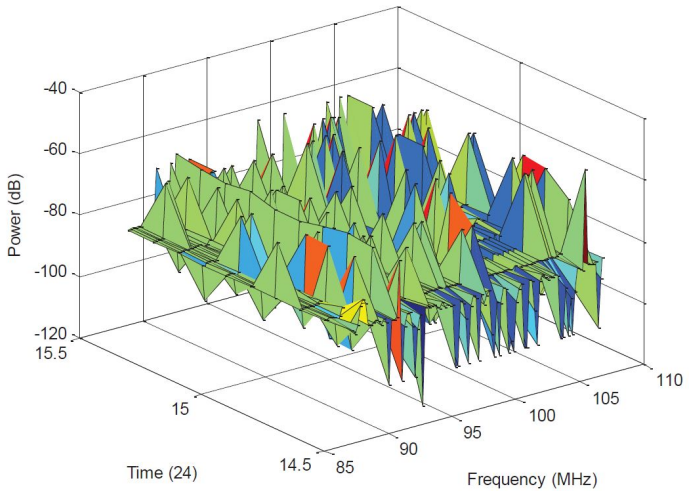

Figure 3.22: 3 USRP B200 scanning results for FM bands 10 times

done. The results obtained from the scans confirm the responsiveness of the USRP B200. Keeping a database of all recorded frequncy entries allows for three dimensional plots of the current high and low peaks of the observed spectrum. Figures 3.23 and 3.24 are good examples of this monitoring system. The displayed peaks from one to another suggest that one standard threshold limit would cause either misdetections or false alarms when applied. This is why an adaptive threshold must be used. Sensing performance as well as the results from detection scan will provide insight on the optimal threshold value to uphold.

3.6 Experimental Results for Threshold Limiting

As expected, the probability of detecting the signal decreased when sensor move away from the primary transmitter as shown in Figure 3.25. From this figure, it was observed that when the sensor was 50 meters away, the power was within threshold 100 percent of the time. As the device got further away, for instance at 600 meters, it still managed to get a

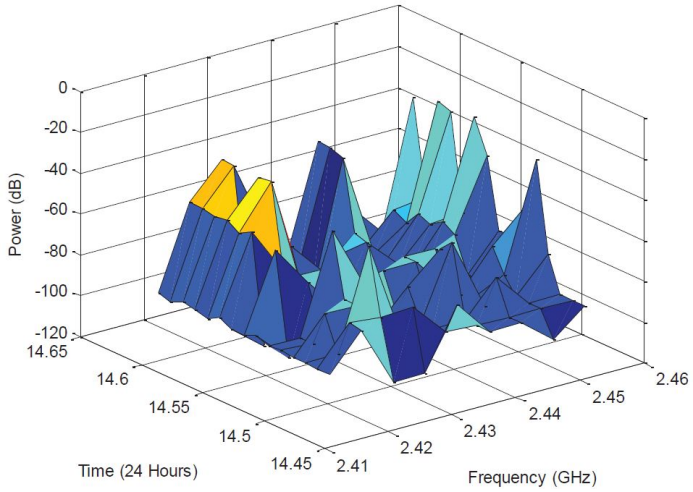

Figure 3.23: USRP B200 spectrum sensing the 2.4 GHz range 10 times

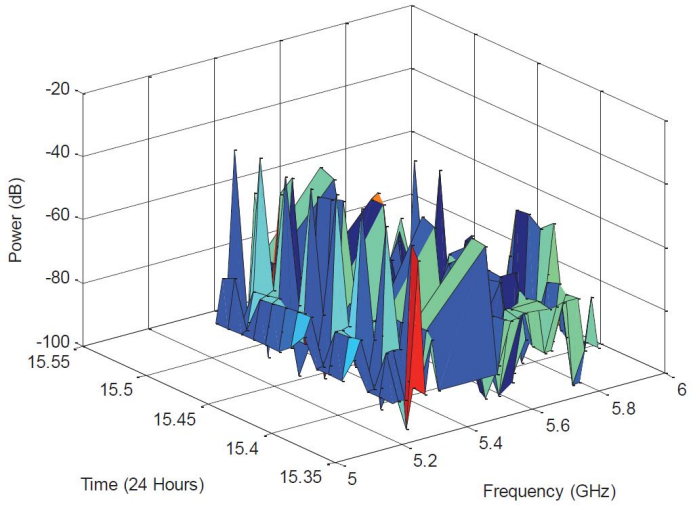

Figure 3.24: USRP B200 spectrum sensing the 5 GHz ISM bands 10 times

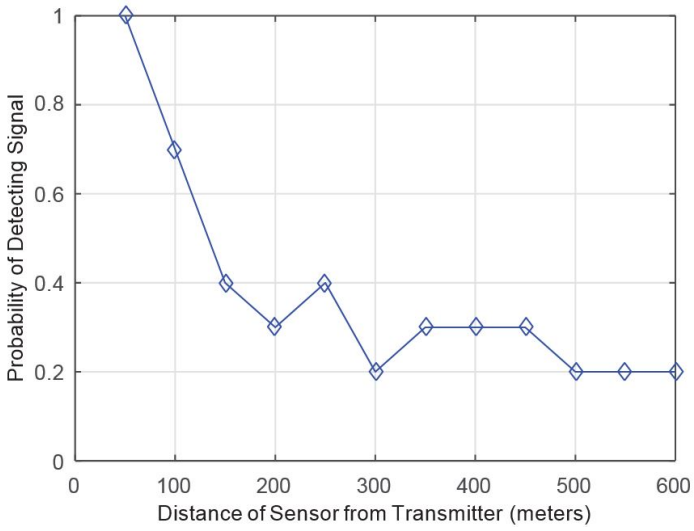

Figure 3.25: Probability of detecting signal vs distance of spectrum sensor from transmitter (meters) where spectrum sensor moved away from transmitter

20 percent chance to be within the threshold. Also in instances like at 300 meters away, due to problems with localization objects, i.e. trees, people, and buildings, it caused a lower probability. Then, the probability of misdetection for different threshold values was plotted in Figure 3.26. As threshold value decreased, the misdetection values decreased. Furthermore, misdetection decreased with increased gain. It is worth noting that the choice of the proper threshold is also important for identifying the idle channels.

The false alarm and misdetection probability was plotted in Figure 3.27. It was observed that the probability of false alarm decreased with increased threshold value or vice versa and misdetection probability increased with increased threshold value. Furthermore, it can be noted that the change in threshold value does not result in change in false alarm whereas there is significant change in misdetection probability as shown in Figure 3.27. The perfect threshold value considering trade-off would be the value at intersection of two

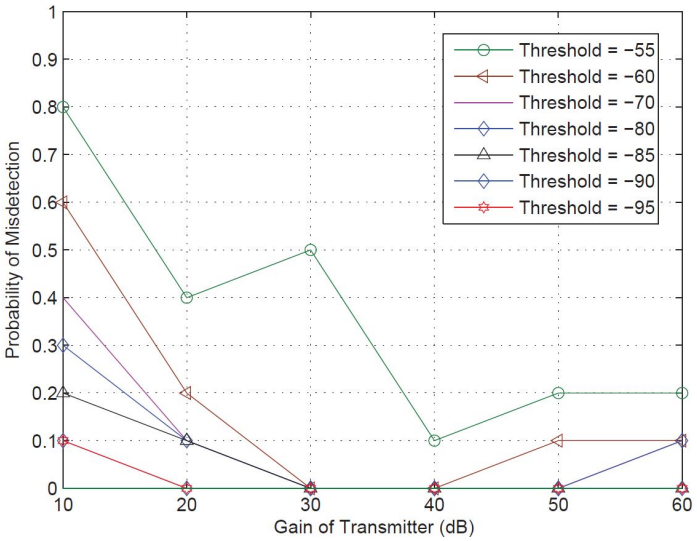

Figure 3.26: Probability of misdetection for different threshold values

probability values.

Last the sensing period vs the speed of a sensor for a given transmission range of primary users, as shown in Figure 3.28 was plotted. It can be observed that higher the speed of the sensor, the lower the time available to sense demanding faster scanning period.

3.7 Adaptive Threshold Evaluation

The results taken by spectrum sensor and analyzed it with various threshold values (Younis *et al.*, 2016). From Figure 3.29, it was concluded that the threshold value must be within the range of -60 and -80 dB based on the results of the detection scan. The factors that must be considered for the performance of the USRP B200 are the probability of miss detection and the probability of false alarms. If the threshold value is too low, e.g. -90dB, then it will increase the chances of false alarms. In contrast, if the threshold is set too high e.g., -50dB,

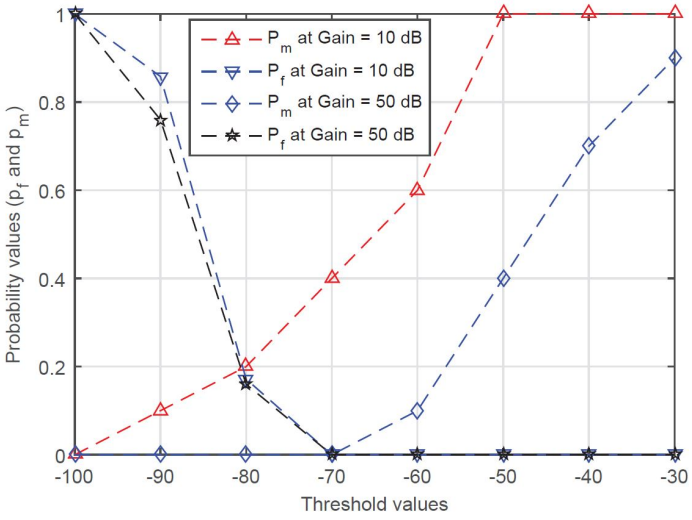

Figure 3.27: Probability of misdetection and false alarm vs. the threshold

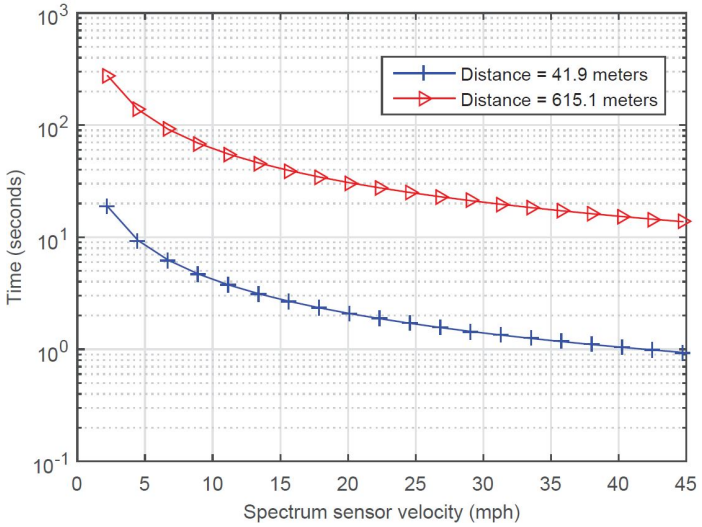

Figure 3.28: Sensing period vs. the speed of the sensor with lower and upper limit of time period for given transmission ranges

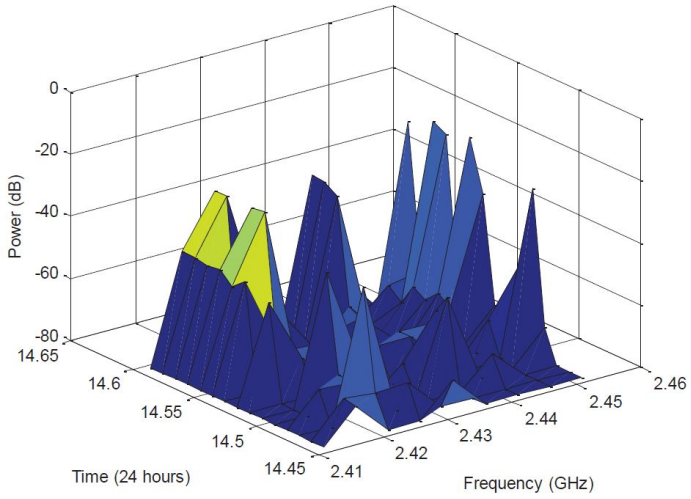

Figure 3.29: USRP B200 spectrum sensing the 2.4 GHz range with a threshold of -80 dB

then far more misdetections will occur. The purpose of this section is to further examine the data from the previous experiments and define the model threshold value.

It can hard to differentiate between the false alarms and the busy frequency. Figure 3.30, in contrast, shows virtually no false alarms, but the peaks are scarce prompting the assumption that there are many miss detections. The principle discussed before is now verified by these results, and better illustrated by Figure 3.31. The threshold being set too low will cause more false alarms and it being set too high will cause more miss detections. The same observations were made while viewing the 5 GHz range, Figure 3.32, Figure 3.33. From these two figures, a graph of the number of misdetections and false alarms is created and displayed in Figure 3.34.

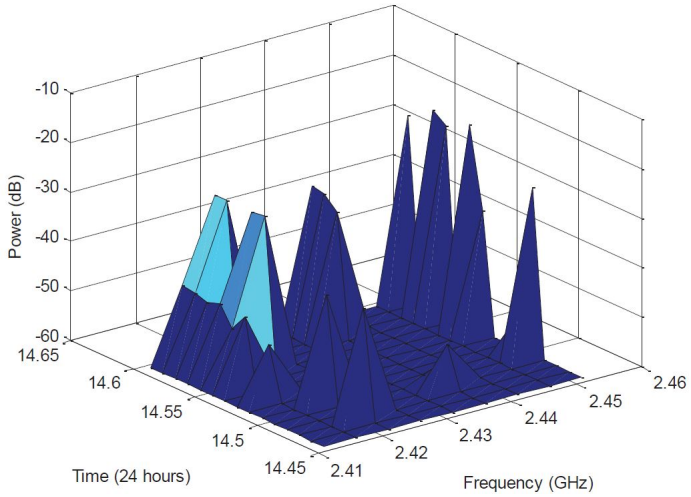

Figure 3.30: USRP B200 spectrum sensing the 2.4 GHz range with a threshold of -60 dB

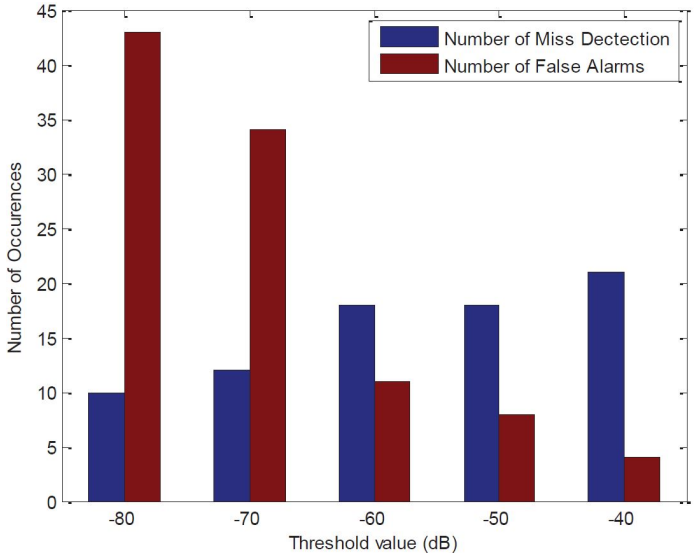

Figure 3.31: The number of occurrences for both misdetections and false alarms in the 24 GHz range

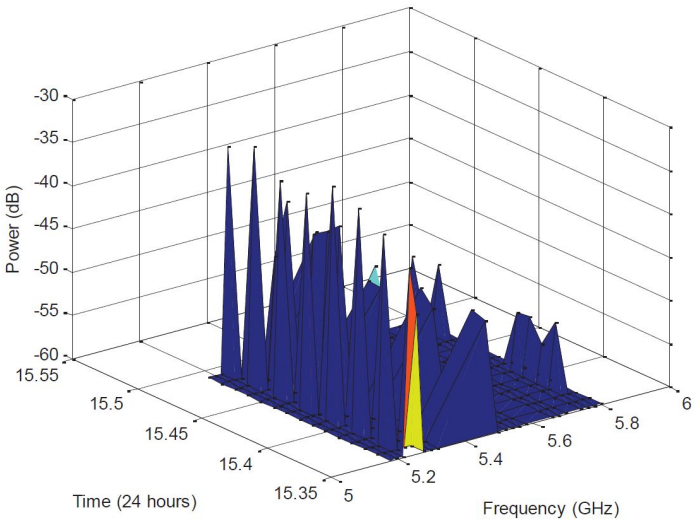

Figure 3.32: USRP B200 spectrum sensing the 5 GHz range with a threshold of -80 dB

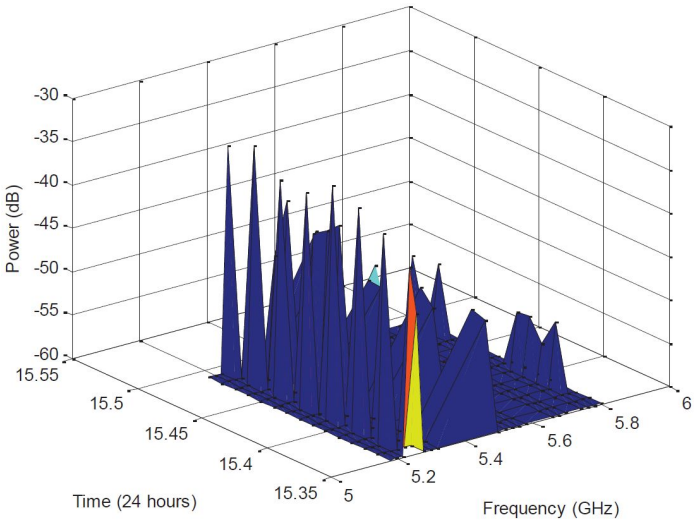

Figure 3.33: USRP B200 spectrum sensing the 5 GHz range with a threshold of -60 dB

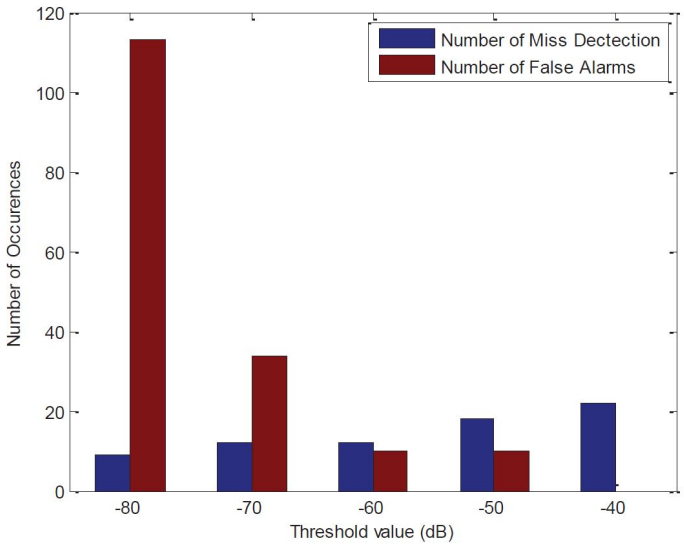

Figure 3.34: The number of occurrences for both misdetections and false alarms in the 5 GHz range

3.8 Geolocation Mapping for Frequencies

In this section, experiments were done using the system to communicate between two devices while moving through designated zones. The first experiment was done at walking speed. This experiment is set up where the secondary user transmitter and secondary user receiver were carried with typical walking speed (2 mph to 3 mph). The distance between them was 5 meter where SU transmitter was following SU receiver in a path as shown in Figure 3.35. This figure shows the two locations of idle Wi-Fi channels for the experiment that are marked as diamonds. Without loss of generality, it is considered that the radius of idle channel circle as 20 meters. The red dots are the locations of SU-transmitter and SU-receiver when they queried the spectrum database. Ideally SUs should get channels only when they are within these circles and none when they are outside the circles.

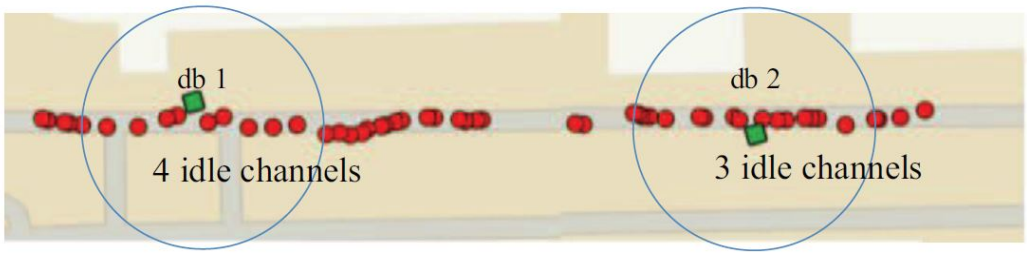

Figure 3.35: Experimental set up for Walking Speed Analysis

In the first communication experiment, secondary user transmitter and secondary user receiver separated by 10 meter started walking from left side of Figure 3.35 towards the first circle. A constant speed between secondary user transmitter and secondary user receiver so that the separation distance remains the same throughout the experiment. Secondary Users were constantly querying the database using their dedicated links to find idle channels. When they found idle channels they used the quorum based rendezvous to find common channel. When the SU transmitter finds list of channel it chose one channel using quorum based rendezvous and was broadcasting a Hello! message. This process was repeated throughout the experiment, as long as it was able to obtain channels from spectrum database for its location. Similarly, SU receiver was doing the same thing that SU transmitter was doing to get a list of channels. Once SU receiver found a list of channels, it used the quorum based rendezvous to find communication channel. It can be observed that the receiver received the Hello! message only when SUs were within the communication range of each other and within the idle channel locations.

The number of idle channel vs. the time instances (steps) that SU receiver received in the first plot and SU transmitter received is plotted in Figure 3.36. As receiver leading

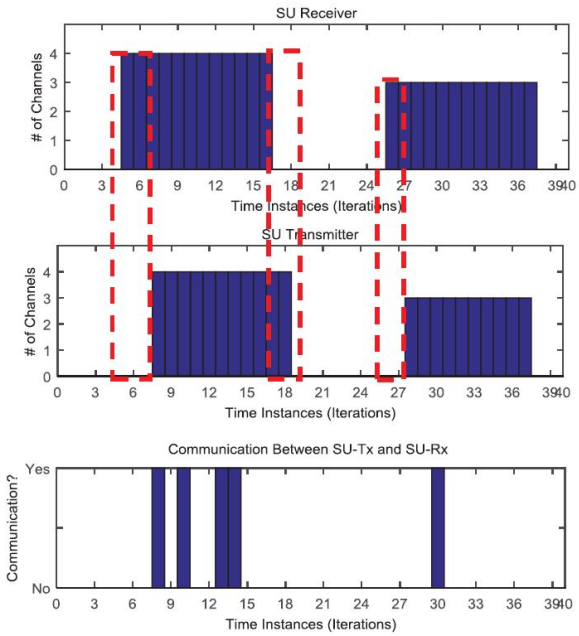

Figure 3.36: Results from Walking Experiment

the transmitter by 5 meters, SU receiver was getting idle channels (4 idle channels when it was in the first circle and 3 channels when it was in the second circle) little earlier than the SU transmitter as shown in Figure 3.36. Once the SU transmitter was also inside the circle of idle channels, it started getting all idle channels (4 channels in the first circle and 3 channels in the second circle). Once both transmitter and receiver got idle channels, they used quorum based rendezvous method to find a common channel to communicate. Last plot in Figure 3.36 shows SU transmitter and SU receiver were able to communicate when they were within the range of each other and within the idle locations. Furthermore, when SUs were moving away from the center of the circle of idle channels, SU receiver was getting no channel earlier than the SU-transmitter since SU receiver left the circle earlier than SU-transmitter as shown in Figure 3.36.

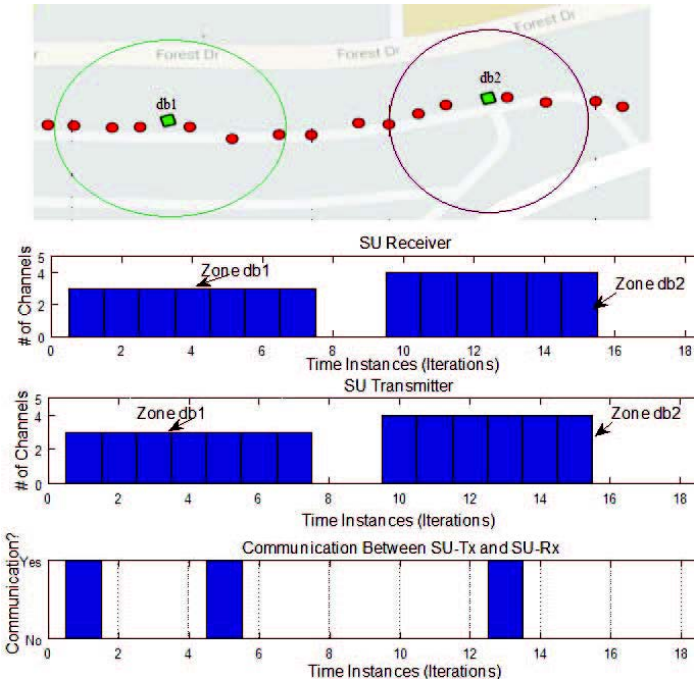

Figure 3.37: Results from Vehicular Speed Experiment

Experiment with Vehicular Speed: This experiment tested the impact of velocity on opportunistic spectrum access and communications. The experiment was set up in parking lot at GSU where both devices were placed inside the car. The lists of channels obtained by SU transmitter and SU receiver are plotted in Figure 3.37. It can be observed that both transmitter and receiver were able get the same list of channels at the same time as they were querying frequently than the previous case. Using those channels SUs were able to establish the link and communicate as shown in the third plot in Figure 3.37. Note that the list of channels depends on when it was queried. For instance there are channel for SUs outside the boundary which is caused by query interval as shown in Figure 3.37. Multiple trials were conducted each with similar results.

CHAPTER 4

PRIVATE CLOUD CONTROLLER

Cloud network capabilities find strength in the ability to operate as a service. Cloud as a service allows users to use its resources dynamically where user demands grow or shrink on-the-fly depending on their operating environment and user demands. Multimedia applications, such as video conferencing, require time sensitive processing in order to maintain a high quality-of-service between any participating parties; in other cases, such as recorded video streaming, we may consider the information to be non-time sensitive due to the fact that the video can be stopped and played multiple times. Cloud based networks operate with the ability to connect to and program virtual operating systems through the means of the Internet. In some applications this can be Virtual Private Networks (VPN) where a user can remotely connect to and operate their personal computer through a cloud based network (CBN). Another application of CBN is the ability to share and write on documents that others are also connected to at the same time. The information stored in a publicly or privately owned storage may see problems with accessibility or security when many users are able to connect to it. This ideology is supported through the concept of Cloud as a Service (CaaS). This term has three major concepts, Software as a Service, Infrastructure as a Service and Platform as a Service, or SaaS, IaaS and PaaS, respectively. SaaS is able to provide users with application based computing without the need of storing the application on the physical hard disk of their machine (Kulkarni *et al.*, 2012); IaaS can provide the user with hardware, software and storage through the Internet (Dawoud *et al.*, 2010), and PaaS

delivers operating system and application development tools over the Internet (Krebs *et al.*, 2014). The strength that comes with CaaS is that a combination of each of these concepts can be purchased by the user and provides the ability to run their entire company with little need of large on-site data centers. However in this scenario privacy and security become a very essential asset. There are many different options when creating a CBN, and one of the most widely used is OpenStack. OpenStack (OpenStack.org, 2016) has a large community that is constantly using, critiquing and updating how the system operates for many different tasks. Further uses and development of OpenStack is explored later in this paper. Various proposed solutions to the secure, high traffic demands of cloud computing have already been implemented by companies such as Amazon, who has employed several techniques in cloud service, such as elastic load balancer (Amazon, 2016), to maintain availability to their large servers. Microsoft has also created their own cloud service, Microsoft Azure, which provides several different services to manage big data and company portfolios managed through their service (Microsoft, 2016). Other popular techniques have been applied to allow for portions of a public cloud server to be rented out for private use; however this raises several concerns, such as the loss of availability if the public cloud is hit with a denial of service attack, and the private sector would also be inaccessible. In this paper we explore several techniques that have been used to provide better quality and security to cloud servers in multimedia access and also relate to how a smart load balancer could be employed to make current methods better. In this research we propose a hybrid cloud where the Smart Load Balancer and Bandwidth Shaper (SLBBS) selects the best suited cloud (private or public) based on sensitiveness and delay requirements of the request.

By offering a cloud network, the service provider can extend to their user resources on demand through service packages. At the same time, the concept of Mobile Cloud Computing (MCC) is also evolving. MCC has the potentiality to overcome the constraints of the performance of mobile entities, such as computational power, storage, bandwidth, heterogeneity and scalability (Chalaemwongwan & Kurutach, 2016). The recent mobile standard Long-Term Evolution (LTE) is supporting the cloud augmentation as new generation mobile applications are needed to overcome the limitations of computation (Chalaemwongwan & Kurutach, 2016). Next generation application data are no longer static as there is a lot more diversity in mobile applications (Selvi *et al.*, 2014). To handle such dynamic data, dynamic resource management can be used by dynamic resource allocation technique in a virtual cloud system (Selvi *et al.*, 2014). This concept allows for users to avoid having to purchase large packages that may include many other pieces of software or too much processing power for the required use. The driving force behind this is known as as-a-service, where software, platforms or infrastructures are offered to the user virtually. A new business owner will be able to maintain their entire business operation on a single machine without needing the complete knowledge of how to configure and operate their operating systems and servers as all the backend processes and procedures will take place on the cloud server side. Resource allocation and data management within mobile clouds have a variety of challenges that have previously been researched, most critically of which are: heterogeneity of data, availability to the network, offloading, and security and privacy (Hu *et al.*, 2016).

The inclusion of a cloud storage system for dynamic spectrum access, allows for the geolocational data to be stored in large quantities. It also offers the ability to bridge together

different sets of localization data obtained from the sensor network. This enables a user to travel from one city to another while maintaining connection with their current location's data set. In this part of the project, a cloud computing cluster is designed and implemented in hardware using three servers. The system is designed to handle complex data storage and access in the form of a smart load balancer.

4.1 Multimedia Access in Cloud Computing

The design of an algorithm that can handle multimedia data, data storage and access becomes a trying task. The first design concept that needs to be addressed is how to conform several data types and device communication protocols into a uniform protocol. Next to address is how to securely store and distribute this data to the intended users upon request. Multimedia data takes the form of many different types, whether it be photographs, videos, or sound clips. Along with them are several types of devices with varying security and communication protocols in which they connect to the Internet. This issues cause a concern for data analysts and developers with security in mind. It is important to build a cloud structure that can handle various data types and has the ability to serve the many types of devices connected to the network. The research in (Chen *et al.*, 2011) proposes one such method to handling heterogeneity in networks with the IP Multimedia Subsystem (IMS) framework. The IMS framework uses the three concepts of as-a-service mentioned earlier in order to build a mechanism that is capable to maintain high quality of service (QoS) manage computing services and user preferences and allows for users to access specific

applications in the cloud with IaaS, PaaS and SaaS, respectively.

4.2 Heterogeneity in Cloud Networks

As addressed previously, a major challenge in the storage of a cloud network that provides user media access is the ability to adapt to the specific users media request, communication protocols and the actual system requirement that particular type of media requires. It would also require the overall cloud network to pull data where the network may exist virtually in very different locations, increasing the cost of transmission of data. It is possible in a cloud network to provide methods for specific types of data to either be placed in or converted to appropriate data type tables in order for the computing system to categorize a user request to maximize efficiency. This ideology is explored in (Korotich & Samaan, 2011) with the use of applying a virtual service model (VSM) hierarchy. In this method they design the system to contain a root layer, containing all possible data that could be requested through the cloud. Then a new layer is introduced for each general type of media. High resolution requests from users will take a higher precedence in the hierarchy compared to low resolution media types. Also mentioned in the work is the problem with redundancy in the layers because each new layer is constructed on the basis of the root; it does however bring to surface a possible way to distribute data for storage inside a cloud.

4.3 Confidentiality, Integrity and Accessibility

Security schemes in data storage offer certain levels of confidentiality, integrity and accessibility to a network system; for example a specific scheme may require the user and provider to share a service level agreement (SLA) which will continually check if what is being stored is agreed upon by both parties. In cloud computing, this system is more complex and has many areas where potential malicious users would be able to steal, change or destroy valuable data. In this section, the framework proposed in (Huang *et al.*, 2011) is explored in order to generate better data integrity and confidentiality for our multimedia cloud network. For data integrity, the proposed method uses two concepts: a Third Party Auditor (TPA) and Proofs of Retrievability (PoR). TPA is a mechanism used to gain trust between the service provider and the user in the network. This mechanism is built by monitoring the data stored in the cloud and its interaction with the cloud provider. A homomorphic authenticator is used to audit the data sent by the data owner and generate a corresponding result. A potential draw back in this system is the possibility of revealing the data owners identity if a malicious user was to sniff the data audit. This could be fixed, however, by using data masking techniques and encryption schemes. Figure 4.1 below demonstrates the typical design of a TPA system.

For obvious reasons it is critically essential to keep multimedia data, either being streamed from a video conference or stored from medical procedures, confidential to only the users with the proper authentication. Several proposed works mentioned in (Huang *et al.*, 2011) cover different schemes to provide confidentiality in cloud networks. One

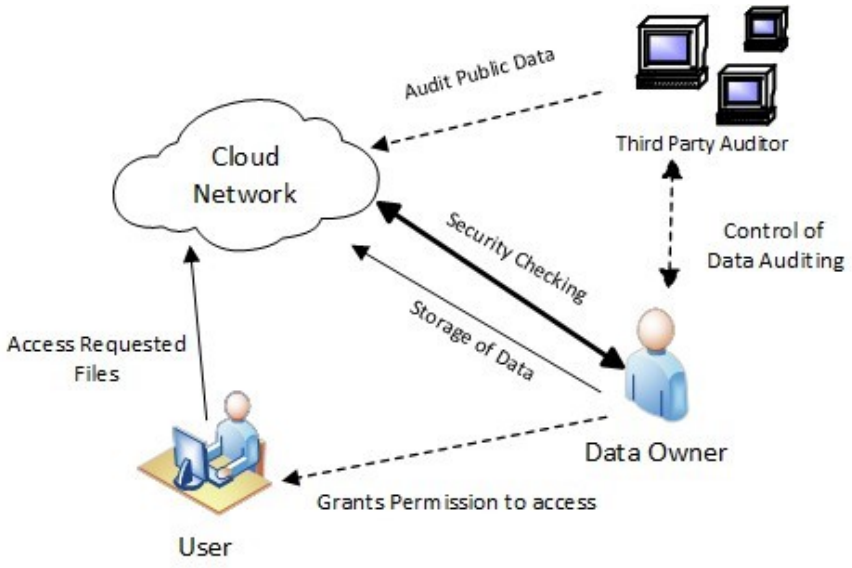

Figure 4.1: Third Party Auditor Topology

method is the use of cryptographic algorithms placed on the data blocks with the key given to the data owners. This ensures that the data stored in the cloud can only be accessed by data owner. Another method is the use of secure provenance model, recording the ownership and the process history to increase the trust of the data owner to the network. Additionally, the use of a fully homomorphic encryption (FHE), was proposed by Craig Gentry. This encryption technique allows circuit evaluation over encrypted data without being able to decrypt it, allowing for better confidentiality. However it may restrict the distribution of data for the data owner trying to access from multiple locations.

4.4 Hybrid Cloud

The idea of renting out small sectors of a cloud to paying subscribers is a viable concept to cloud service providers. This allows them to allocate portions of their network that may

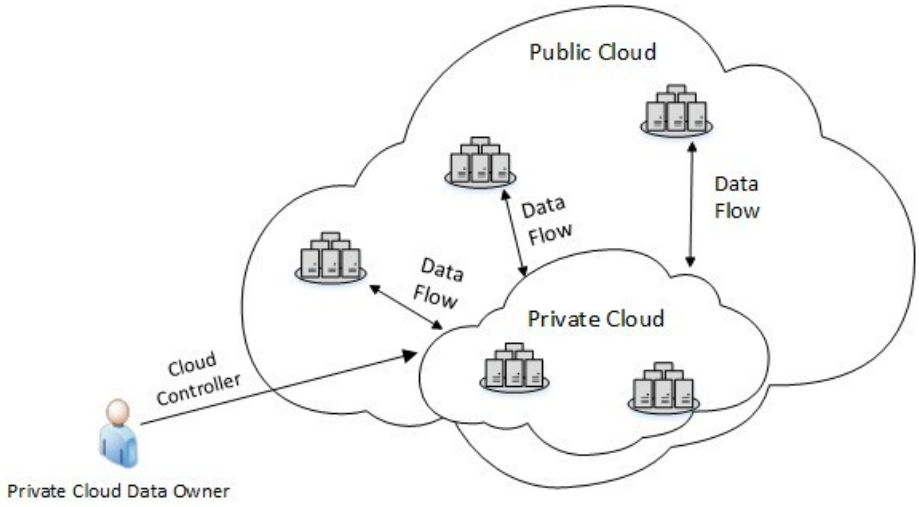

Figure 4.2: Cloud in Cloud Model

not have been fully utilized. Having a public and private cloud exist inside the same overall structure provides significant increases to usability. However it comes with new types of security risks to consider. The research work in (Zhang *et al.*, 2013) considers the effects of placing a cloud inside the cloud, otherwise known as a hybrid cloud. Demonstrated in Figure 4.2, this model allows for a small subsection of the public cloud to be exclusively owned by an administrator while still keeping data links to other parts of the public cloud. The private cloud is able to act as both its own structure and still remains connected to either the entire public cloud or via certain specific data links, depending on the need and configuration of the private cloud. In this model, it is possible to reduce the cost of communication between public and private than traditional sense of hybrid cloud where the two clouds act as separate entities.

The cloud-inside-cloud configuration brings around a new ideology on how to manage private sectors in cloud based networks. However it does not consider the effects on heavy

multimedia traffic access that would occur when many private networks call on a public cloud at the same time. In this scenario, innovating onto the cloud in cloud structure, by applying methods found in private cloud frameworks to achieve the goal of creating a load balancer that will distribute data quickly among large cloud networks.

4.5 Private Cloud Infrastructure

One of the main challenges when managing a cloud network for users to store and access data is the ability to maintain confidentiality, availability and integrity of the system. One of the key problems comes from the need of a uniform security intrusion and detection method to be employed over the cloud. A cloud network could not allow for individual users to access and change security parameters simply because otherwise the availability to sectors of the network would break. The research in (Krautheim, 2009) establishes several concepts of private cloud security. First they introduce a private virtual infrastructure (PVI), where the data owner and cloud operator are in common terms of security protocol while the virtual data center stays in direct control of the data owner. In this scenario, it is obvious that role based interactions will control the structure of the cloud where both the operator and the client would need to establish service level agreements before establishing a secure connection. The concept of Trusted Platform Module (TPM) is also introduced in their research. This module stores cryptographic keys in the platform configuration registers (PCRs) and establish the access of the clients to their configured platform. Based on this architecture, a certain level of trust is formed in the cloud network as only specific users

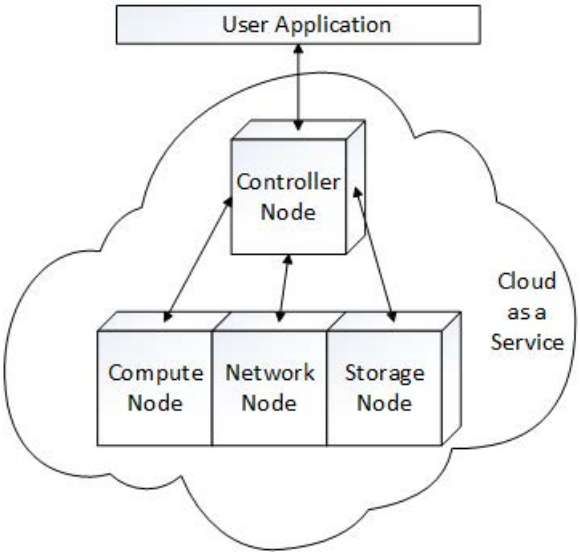

Figure 4.3: Cloud Network Using OpenStack based Topology

will be able to access specific sections of the network. This concept builds a two layer architecture for private cloud security, the IaaS fabric layer and the PVI layer, establishing important rule based operations for both vendor and data owner.

4.6 Cloud Virtualization Techniques

OpenStack provides an IaaS for users to develop cloud networks. It uses several components in order to design their cloud computing architecture, consisting of the essential blocks: the cloud controller, compute node, network node and optional storage node (Docs.OpenStack.org, 2016). Figure 4.3 demonstrates the typical architecture of an OpenStack private cloud service (Docs.OpenStack.org, 2016).

The controller runs the virtual machine Identity and Image services, management portion of compute node and the dashboard. The dashboard is a web-based interface

that users can use to interact with OpenStack services, such as launching an instance and assigning IP addresses. It also serves as a means for the data owner to interact with their data, through queries, entry tracking and utilization of their cloud server. The controller node is also capable to operate as a storage block and cloud operator, generally used to access the network and compute node in order to run the cloud server. The compute node is responsible for the hypervisor that operates tenant virtual machines or instances, connects network plug-ins and firewall services. It can also contain a third network interface in the storage to improve system performance. The network node runs the networking plug-in and several agents to provide switching, routing, Network Address Translation (NAT) and Dynamic Host Control Protocol (DHCP). OpenStack requires several pieces of software in the design to mainatian authenticitiy to the users and for storages of data. These services each have a project name inside OpenStack, which may cause confusion to users configuring the network for the first time. To avoid such confusion, the core services and their counterparts are listed in pairs (in the same row) in Table 4.1. This information and the discussions in the following section are useful in helping us determine the appropriate structure, model, system, operating system, software and hardware for building our own private cloud in this research.

4.7 OpenStack Cloud Design

In many academic cloud deployments, open source allows the deployment of cloud networks without the need of expensive licenses. This project uses OpenStack to build and test the

Core Services	***Project Name***
Dashboard	Horizon
Compute	Neutron
Object Storage	Swift
Block Storage	Cinder
Identity	Keystone
Image Service	Glance

Table 4.1: Common OpenStack Services and their corresponding Related Project Name

cloud network. OpenStack is open source and was selected in this project thanks to its large community of developers in both industrial and academic cloud deployments. A key strength that comes with developing a computing network using OpenStack is that the cloud models can have a variety of configurations to serve a task with excellent flexibility. Some example uses are public cloud, high throughput computing, web hosting, and video processing and content delivery, etc. The architecture built for this project requires three components: the controller, compute and network nodes. The controller node is responsible for running the virtual machine Identity and Image services, management portion of compute node and the dashboard. The compute node is responsible for running the hypervisor that operates tenant virtual machines or instances, and connects network plug-ins and firewall services. Lastly, the network node is responsible for providing switching, routing, NAT and DHCP (Docs.OpenStack.org, 2016).

4.7.1 Troubleshooting OpenStack Cloud Design

OpenStack is opensource software, meaning it is free to use and gives the user developer abilities. Due to this, there is the potential for errors to occur during the setup phase. The recommended method, at this time, from OpenStack is to use their Autopilot, however it requires several extra machines to act as outside controllers in the installation process. When working with Autopilot, problems can occur when the user is connected to a network where they are not the network administrator, such as a university. Autopilot requires the ability to bootstrap the machines and dynamically address IP addresses to it. This causes issues to the network, because it appears as if a computer in the network is trying to issue IP addresses using the gateway address. When not going the Autopilot method, it is possible to manually install and configure each piece of the OpenStack cloud. This method requires much more time as each ".conf" file of all the software must be reconfigured with the correct IP, database password, and user authorization. There are several documents from OpenStack that serve as a guide to this process, however the manual method requires a higher knowledge in network addressing and administration, however it can be accomplished. The final method that was explored is to use the "git clone" command. This method grabs all the required software and a user generated file for host IP address and database password, and installs everything to the machine. It is by far the easiest method, however finding the script files to make adjustments can cause the whole system to stop working. It also became a challenge to add more software later, due to the fact that stopping the services would often cause one or more other services to start working improperly.

4.8 Resource Management, Allocation and Provisioning

Resource allocation within mobile cloud computing networks has been presented in several different ways, typically generating a cost function per the efficiency of the required request. The work presented in (Su *et al.*, 2016) presents an adaptation where the overall cloud network is not localized and requires mobile social users, cloud brokers and a mobile cloud. When a request from a social user is presented to the broker, a cost for the resources is determined and the request is sent to the cloud. When the cloud broker negotiates higher or lower costs, the mobile user would then make the decision to connect. Their work presents a game theoretic method of resource allocation for better energy efficiency. Another proposal made by (Wang *et al.*, 2016) aims to reduce the overloading on the cloud by optimizing user traffic through segmenting the data. In this manner, incoming tasks can be organized in a more dynamic order to appear as if there is less traffic coming in. Where solutions developed have aimed to solve specific issues, mobile cloud lacks a common framework that will dynamically determine the needs of the system based on the user requests.

To determine the ability of a load balancer to efficiently handle these problems, a measure of QoS is conducted. QoS can be considered as several different measures dependent on the system that is being observed. In mobile cloud computing, the important factors of QoS are the ability to remain connected to the network and the overall throughput of the data. Network connectivity and reliability among mobile carriers has significantly increased, however there are still areas where dead zones exist. Lack of availability in a system where major computation and storage for a mobile phone takes place becomes a

major concern. Mobile cloud is a technology supporting online dynamic resource allocation enabled services. Dynamic Load Balancing (BLD) mechanism can be used to distribute the resources by maintaining scalable workload among every node in the network. Features like resource optimization, diminishing of response time and down time, maximizing the throughput, avoiding of overload can be obtained by Dynamic Load Balancing techniques (Liu *et al.*, 2013).

4.9 Dynamic Resource Allocation

In resource allocation, one of the challenging parts is to categorize the mobile resources per its priority factor. The priority factor can be assigned per its requirement, time sensitivity and the size or space of the data. For example, if there is an application in the mobile device that is dealt with real time voice data or some real-time gaming data, undoubtedly the data of this certain application is highly time sensitive. Similarly, sometime certain applications are required to access and process the data immediately or depending on the time sensitivity of the data, for example, video broadcasting and streaming. To explain further, a variety of data that is stored on a mobile device does not require continuous synchronization or need to be processed immediately when created but they just need to store in mobile cloud storage. Subsequently, this kind of data can be considered as less prioritized data. Some applications, such as HD video capturing, may generate large amount of data, hence they may consume large amount of storage in mobile devices and therefore may affect the overall performance of these devices. In this case, data from the mobile devices can be sent by

sensing the available space in the mobile devices. If the mobile device does not have enough space, it should send the data to the cloud immediately. Otherwise, when there is enough space in the mobile device, a certain predefined schedule can be set to transfer the data.

With the context of the origin of the mobile data, data can be categorized as follows:

- User Generated Data: User generated data can be referred by the data generated by the user according to the requirement of the user, such as contact information, text messages, captured photos and videos, created personal notes.
- Application data: All mobile application driven data can be classified as application data, such like email applications data, GPS information, map information, social networking data, various gaming and application data, etc. Some of the application data may require frequent access as per user demand basis or application requirement basis.
- System data: All data associated with the system information, system files, system configuration belong to this category.

4.10 Smart Load Balancer

In cloud computing, load balancing is defined as the ability for the system to take incoming application data from the user, measure the computational requirements and determine which of the availability zones it needs to be stored in. It is also required to handle any incoming data to an application so that the processing ability of that application is not overloaded (Tai *et al.*, 2011). The load balancer will have two main functions, finding the

best location that information should be stored and finding the best path a request should take to retrieve the information. In mobile cloud networks, this poses a problem, due to the heterogeneity of the incoming and outgoing data types and security. In current load balancing methods, the request from the user is granted based on the current availability in each of the zones and if the request can be filled without overcoming the system. The overall basis of how a cloud load balancer is deployed can be categorized as either in software or in hardware (Heinzl & Metz, 2013). From the related work, it is possible to classify sever key characteristics that are involved when developing a load balancer for mobile cloud computing. The first to discuss is the ability to scale up and down in the network. When many more machines are added to the system, the algorithm for load balancing must adapt to this change. The next characteristic to observe is time based load balancing. In the work presented in (Madhumathi & Ganapathy, 2015), the proposed algorithms are round-robin, equally-spread current execution load algorithm, and active VM load balancing. In the round-robin algorithm, a randomized list of all the virtual machines is generated and sorted into a list for processing. The fallback of this method is that certain nodes can be consistently missed in very large networks. In equally spaced current execution, it was noted that the load balancer was completely in charge of determining the selection of the VMs. This system works well in terms of overall execution time, however as addressed in (Madhumathi & Ganapathy, 2015), a minor fault in the load balancer would cause catastrophic problem to the entire system. In the active VM load balancing, all the requests made by each of the VMs would be logged and the least used VM would be placed at the top of the priority list when resources are allocated. The drawback to this system alone is that users in need

of using large amounts of data would have less privilege in acquiring resources compared to users that do not necessarily need access. Each of the methods stated above can serve as a foundation candidate for load balancing with appropriate modifications. A smart load balancer will be able to intelligently define the incoming requests by predicting the needs of the request based on the data type. It will be possible to utilize the discussed methods in part within the algorithm of the smart load balancer to effectively maintain large networks. This proposed method of a smart load balancer will establish a set level of need before greed (NBG) in the system when requests are made. This parameter is used to determine whether the request from the user should be granted based on total resource capacity required, type of data, or the priority level of the user. When allocating resources, NBG will consider priority users that absolutely require the system before any others. Examples of this would be mobile service providers granting a mass broadcast of emergency information to all users. The metric of NBG is discussed below. A problem is created on when and how to evenly distribute available resources to each user. Mobile cloud networks are intrinsically large with a varying amount of data types. To properly design a load balancer. The idea of need before greed is a method to establish a protocol where all users will agree that whoever truly needs to have the most resources will be granted it first. In the event of multiple users with a need for data, or when one user has constantly needed the resources, the system will then establish an algorithm for fairness. In many cases, the level of fairness in a system is dependent on the current usage of one user from another. For example, if the system is aware that one user has been granted a large amount of the resources for an extended period, that user may end up at the end of the queue when it sends another request. This proposed

method aims to solve issues seen from load balancers that allocate resource based on usage or systems that use timing as a metric by applying this need before greed metric.

4.11 Proposed Methodology

The proposed model presents a new algorithm that will take a users request and generate a response based on authenticity, trust level and multimedia data type, and correspondingly grant access to the stored data (Cushman *et al.*, 2017). In the event that the data is open to the public, such as video streaming applications, the algorithm will call the appropriate portion of the cloud network where data is stored as not to disrupt any current data being streamed from a cloud portion with private or higher security level. Figure 4.4 depicts the proposed model, where a private cloud network for Georgia Southern University contains secure (multimedia) data for official use by faculty, which is kept separate from the network where public data, such as information about sports events or academic news, is accessible by all. An algorithm to be built inside the SLBBS will serve as pathway for all user requests that wish to have access on the cloud and direct the request to the appropriate network.

Based on the above studies in this research project, the hardware of the proposed infrastructure requires a minimum of three rack servers: a controller node, a network node, and a compute node. The controller node runs the virtual machine Identity and Image services, management portion of Compute, and the dashboard. This server provides service to both Georgia Southern Security Private Cloud (GSSPC) and SLBBS. The network node runs the networking plug-in and several agents that provision tenant networks and provides

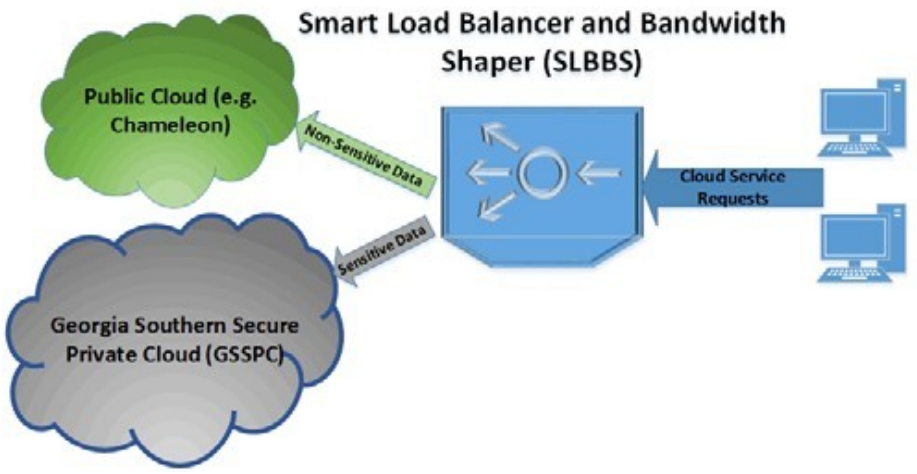

Figure 4.4: Proposed Infrastructure includes a Smart Load Balancer and Bandwidth Shaper (SLBBS) and Georgia Southern Secure Private Cloud

switching, routing, NAT, and DHCP services. This rack server will be configured and modified to function as the SLBBS. The compute node runs the hypervisor that operates tenant virtual machines or instances, using Kernel-based Virtual Machine (KVM) as the hypervisor. The compute node also runs the networking plug-in and an agent that connects tenant networks to instances and provide firewall (security groups) services. The software of the proposed infrastructure, including Red Hat Linux, OpenStack with KVM (Kernel-based Virtual Machine) and Linux-based open source software and tools are free of cost. OpenStack with KVM solution is one of the most popular open source cloud operating options with excellent scalability. In addition to the embedded security features provided by OpenStack, Linux based open source security and forensic software and tools, such as Snort IDS/IPS, The Sleuth Kit (TSK), and RainbowCrack, are all available free of cost. These combined advantages provide great potential for future collaborative research in multimedia

networking and cloud security and digital forensics with the flexibility of growth in scale.

4.12 Chapter Results

Each of the machines runs Ubuntu Server 14.04 and has OpenStack cloud software installed in order to store and provision virtual machines. In this model, the lead node is the controller, which maintains the communications between database storage of the users, software and permissions among the nodes. The main control station uses the Ubuntu Metal as a Service (MAAS) as a means of software installation and updates. MAAS allows for very easy scale-up and scale-down of physical machines, thanks to the fact that any server connected is simply seen as clusters of virtual machines. One cluster contains the nodes and in each node runs the required software for the cloud.

The MAAS controller has two network interface cards (NICs) to keep Internet communication with the university network IP address and to host the private set of IP addresses. The cloud computing system is hosted on the private network. For each machine to be added into the cluster, they are each given an IP address and the gateway IP is the same as the MAAS controller private IP. In this model, the MAAS controller is given a class A private IP address 10.0.0.41 and the gateway IP for each of the machines is configured with the same IP address. With each of the nodes on the private gateway, they are each set to boot from the network connection, such that the MAAS controller will automatically assign each an IP address while also running a script file to gather the machine information. Once the boot is successful, each machine will appear in the MAAS controller interface as

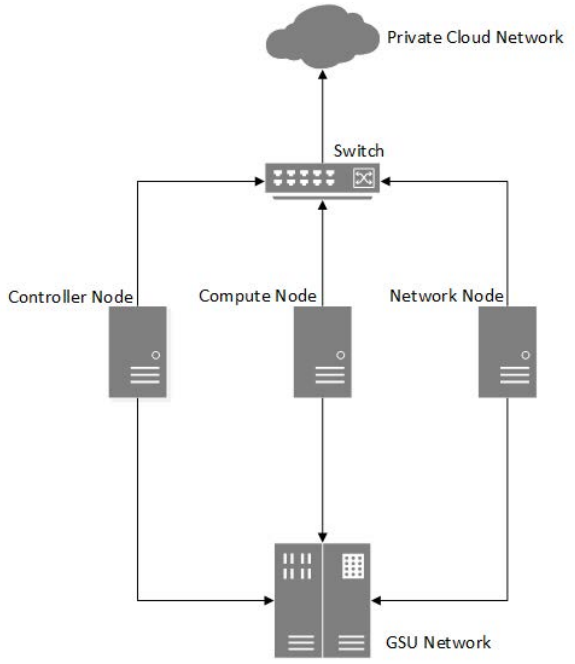

Figure 4.5: Hardware Network Model

shown in Figure 4.5. From this point, the MAAS controller can wake up the machines via Wake-on-LAN and finish gathering the machines specifications.

The MAAS controller uses an interface to host each of the machines that are connected to it. OpenStack Autopilot pulls from the same pool of machines to create the private cloud. OpenStack cloud allows for the cloud provider to access the overall configurations through the dashboard login and at the same time allows for users to log in only with their level of access (least privilege). From the dashboard, the cloud provider can issue resources in the form of virtual machines to meet the needs of their users. Similarly, the users are able to log into their designated portion of the cloud in order to access their data, operating systems and other applications they may have saved. One piece of software that allows for monitoring how resources are being distributed is the OpenStack Horizon Dashboard

Figure 4.6: Front View of Hardware used for Mobile Cloud Network

web interface. The physical system hardware of this proposed framework can be observed in Figure 4.6. The controller, compute and network nodes can be observed as a stack of three-rack system, each of which has at least 1 terabyte of available hard drive space. Each of the nodes are connected to a switch, which is connected to the MAAS controller. From the MAAS controller, the private cloud network can access an Internet connection through NAT between both network interface cards. The network administrator can log into OpenStack and provision data.

To test and implement data and resource allocation within a mobile network, it is possible to sanction portions of the clouds resources to recreate a real mobile network of multiple users. The mobile cloud framework will then be able to provision resources to sets of mobile users when needed and the load balancer can then be tested.

CHAPTER 5

ADDING A LAYER OF SECURITY IN SDRN: MOVING TARGET DEFENSE

The network design for moving target defense can be modeled with a varying number of available computing clusters, which the attacker will be trying to get into. This array of computing clusters will have a common controller used to keep synchronization during transition periods. This network is visualized in Figure 5.1.

The connected computers to the gateway represent any variety of applications that can exist on a network (physical or virtual machines, software defined radios, or mobile devices). The attacker represents any computer system that would attempt connection to any part of the network. Once in, would begin stealing or destroying information. The cloud cluster controller is tasked with keeping a synchronized connection for the entire network, which is done by establishing the changing interval, the current IP, port, or frequency, and what the next hop configurations will be.

5.1 Moving Target Defense Configurations

Two techniques of moving target defense (MTD) are host-based and network-based. In a network based MTD, network properties are periodically changed to increase difficulty for the attacker to get into the network, the most common is the IP address (Yeung *et al.*, 2016). Utilizing IPv6 in MTD provides the network with a wide array of possible IP addresses for hopping purposes. However, the challenge to using IPv6 is the increase in overhead on the network as hopping addresses generates network discovery protocol, NDP, and messages.

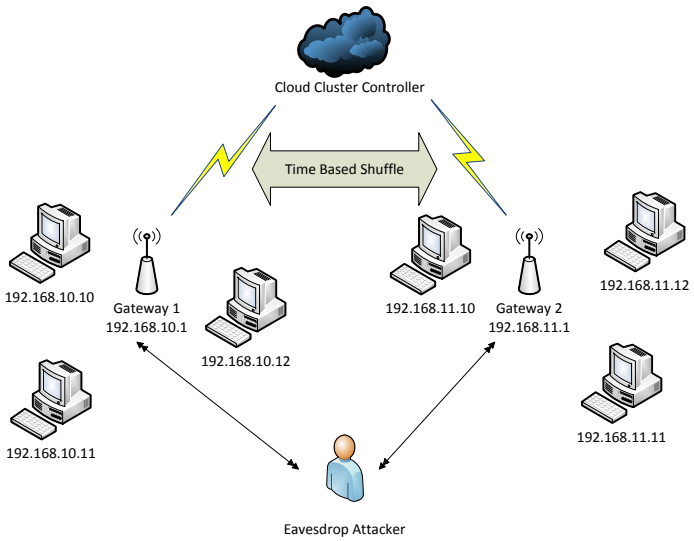

Figure 5.1: Moving Target Defense System with Attacker

IPv6 provides a strong advantage over IPv4 in very large network systems and services because IPv6 allows for a 128 bit address. A MTD network using IPv6, commonly referred to as MT6D, uses an encapsulation method to confuse the observer by generating a false sense of network activity. Using MT6D proposes a defense against an attacker by causing the attacker to spend a much higher amount of resources on reconnaissance (Yeung *et al.*, 2016). The moving property can be handled by the DNS server with a short time to live assigned value, so that IP addresses change frequently. The use of IPv6 is highly sought after in MTD systems, due to the fact that IPv6 offers a large array of varying IP addresses. The DNS server can also handle access control by assigning users to a unique portion of the mapped IP addresses, and revoking them when needed as well. Distinguishability is the most challenging to deploy in a system because of the ability for an attacker to passively access a system (Corbett *et al.*, 2014).

The mechanisms that change within an MTD system are categorized based on the

mechanisms and the type of pattern it follows. Three of the mechanisms are software transformations, dynamic platform techniques, and network address shuffling (Cai *et al.*, 2016). The idea of software transformations is to focus on the applications that are running on the system. In this case, the software or application will exist in different variants which will be randomly selected to be the active software version. Dynamic platform techniques involve dynamically changing properties in the operating system and hardware. Recent methods of dynamic platform techniques use cloud based systems to store the operating system variants and load them accordingly. In network address shuffling, the main goal is to prevent reconnaissance in the system. MTD has three fundamental patterns, hidden, variation and assisted. In the hidden pattern, the attacker can get into the network for a variable amount of time, however when repeating the reconnaissance stage again, the network will have appeared to be no longer active. The variation pattern is comparable to hidden, however when the attacker makes a second pass, the network will have a different set of security protocols, preventing access.

5.2 The Attack Process on a Network

The first stage of the attack process on a network is the reconnaissance stage, of which the focus is to determine the best angle of attack. It is because of this understanding that defenders of cyber-attacks work to make reconnaissance very difficult. Eavesdropping is one of the biggest challenges to stop malicious acts in a network system. It is the process of secretly listening to a network and copying the data as it is sent (Ma *et al.*, 2016).

Traditionally, encryption and authentication are backbone layer defenses that are passive in the network. This means that the encryption system does not dynamically change at any given time. Moving target defense is a new method that will allow for an active defense to stop eavesdropping (Ma *et al.*, 2016). In a traditional use of MTD, IP addresses or ports are changed to keep attackers from listening to the network, however little is done to change network protocols, due to the complexity. Eavesdropping is categorized into two types of attacks, Session attack and Packet attack. In a session attack, the entirety of the communication session is grabbed by the attacker and then analyzed based on the network protocol. In a packet attack, a series of packets from the session are grabbed and analyzed for their source IP and destination IP, which could be potentially suffiecent for the attacker to launch a Man in the Middle or Distributed Denial of Service attack (Ma *et al.*, 2016).

One proposed method is to use MTD with Protocol-Oblivious Forwarding (POF). POF will allow the network to simply forward the packet based on the key associated to it; otherwise it has to parse the packet first before determining what to do with it. In this setup, the clients use dynamic message packaging and dynamic routing paths in order to keep the attacker confused as to what the source and destination IP addresses are. This proposed method will block both session and packet attacks by continually keeping the attacker guessing which bits of data fit the right network protocol.

5.3 Operational Costs

Operational costs of MTD, as mentioned, can be the actual cost in currency of the system, however it can also affect overall system performance, network stability and effectiveness. Determining how much a physical system may require in capital is first determined based on compatibility. The physical hardware requirements for the level of security needed will increase the cost of the system. The biggest challenge to an efficient MTD system is the available bandwidth of the system. To have a secure system, the total number of channels available for the system to "hide" in, directly impact the difficulty of the attacker finding it. To clarify, if the system is designed so that it only has 10 available channels, then the attacker has an easy time to scan all channels to find the information, whereas if the system has 50 possible channels to pick from, the attacker will spend a much greater amount of time or resources trying to find the information.

Capital restrictions are a major driving force behind the decision to change systems. The MTD system requires constant synchronization based on CPU cycles and memory in the network, which could take away from the processing power the company may need to service their own demand. The effects on the performance metrics may also lead to a loss in availability in the system. In a CPU system, it is possible to over clock the synchronization, however it is not necessarily to best practice and can lead to system failures. Another operational cost of MTD is the effectiveness of the MTD system itself. Considering an MTD system with many access points and changes happening in the system, the controller has a very complex role in determining when and how requests should be

handled. Deploying a large network to handle minimal security work would be wasteful, and in contrast would be impossible to secure highly classified data in a small MTD system.

The main approach to determining the right cost functions of deploying an MTD is by observing possible network parameters. First, classification and value assignment of the system should be taken care of and then experimental bandwidth consumption can be handled (Leeuwen *et al.*, 2015). The classification step determined the work factors that the system requires including: operating expenses, capital expenses, performance in either network or host based and applications, service impacts and scalability. Depending on the classification, a metric and unit will be assigned. The metric for operating expenses would be operator workload and the unit would be in physical man hours. Whereas for scalability, number of nodes and count would be the metric and unit, respectively. This classification system leads to a concise requirement and possible prediction of whether a large or small network can be handled.

Observing the physical and cyber costs of deploying a new system is very crucial when defending the use of a new technology. There are very clear advantages to studying these metrics and optimizing where necessary to provide the strongest case as to why this technology is needed, other than just for more secure data transfer.

5.4 Obfuscation of the Attack Surface

A key advantage to using software defined networks in moving target defense is the ability to obfuscate the attack surface. Using software defined radios, it is possible to change each

of the network characteristics to protect the network. Two of the proposed network attacks, to protect from, are network reconnaissance and OS fingerprinting (Kampanakis *et al.*, 2014). Using an SDN controller, the traffic coming through a network is monitored. If the traffic is malicious, the SDN controller will attempt to quarantine that part of the network, by blocking off that group of devices. The main process of this defense is to open more possible ports than the ones which are already open. This causes the attacker to search more possible entries before finding the actual port. In the event that the attacker attempts to find network configurations through an HTTP GET eavesdropping method, it is possible to change different operating system information. The httpd service will work with the SDN controller to create a dummy service version.

The network firewall is normally used to keep out attackers by preventing attacks on operating system information, however the SDN will reassemble TCP into a spoofed version that will exist on the network. Users that have access by presenting the correct key, will be given the correct information from the SDN controller. In the SDN, route mutation and host randomization can also provide ideal possibilities to prevent an attacker from finding the correct path. Route mutation adds problems to the attacker in operational cost, due to aiming randomly when sniffing packets. If they wanted to be more effective, they would need more robust machines and algorithms.

Many proposed algorithms attempt to make it harder for an attacker to get into the network, however there is generally a problem for the network being defended when using high bandwidth applications (Li *et al.*, 2014). The focus is to make sure that the network can morph to prevent an attack, while maintaining a time sensitive goal for transmission

set by the application. Creating a real-time traffic morphing algorithm requires three pieces that work in the algorithm. The first piece is to create an adaptive packet generation, next is maintaining deadlines in the packet generation scheduler and last is to minimize redundancy. The adaptive packet generation is responsible to maintaining uniqueness in the system. The deadline schedule is responsible for not allowing the first part of the algorithm from taking too long or from generating packet combinations that will cause overhead. Minimizing redundancy is a final check that the system overhead does not take too long.

5.4.1 Encrypted Key Exchange

The transmitter and receiver will both require access to a database to keep a constant synchronization to one another. The transmitter will take the data packet and split into pieces and assign each to a frequency. This information is sent to the database and given to the receiver. This process opens a vulnerability in the system if the attacker was to just ”sniff” this information as it is passed to or from the database. Several cryptography methods are considered to solve this issue. The most common example is the ”Bob and Alice” method as presented in (Taha & Alsusa, 2015). This case presents two devices that will share information. This case provides that both devices have a preexisting code-book made up of n-bit matrices. When Bob sends a signal to Alice, Alice responds by generating a key as an acknowledgment to Bob. The acknowledgement is deciphered by Bob’s code-book and then the keys are exchanged by both Bob and Alice. This process requires a predefined encryption method that all devices in the network would need to possess, however in a highly dynamic network, this would case problems. To provide a secure key exchange between

two parties without a preexisting knowledge to each other by the Diffie-Hellman (D-H) key exchange (Li, 2010). The D-H key exchange is able to generate the shared secrets only when needed and only requires some global parameters. Common configurations of D-H are Message-Digest Algorithm 5 (MD5) and Secure Hash Algortihm 1 (SHA-1). The draw back to this method is a man in the middle attack. This case, the attacker appears to be a legitimate user requesting data exchange. LabVIEW offers a cryptography toolkit, similar to the USRP toolkit, where the encryption method can be selected and the input output can become the middle ground between when the data message is broken into pieces and when it is sent to the database.

5.5 Cloud Controller Characteristics

There are many common open source cloud controllers that have been used for synchronization in MTD systems. One of the most used is the OpenFlow API, which can work as a load balancer, handling requests in a round robin basis. The API works by picking from the pool and handling each request. The main method of changing network specific characteristics either by using an encrypted pseudorandom sequence between both the transmitter and receiver continuously or by simply using a look up table to keep track of the network changes on both ends; the latter being simpler but the table could be eventually broken into (Corbett *et al.*, 2014).

After the network controller has been setup, the next stage is to determine how packets will be sent from transmitter to receiver. Cloud based network systems offer a potential

increase in the effectiveness of a software defined radio. This is the case due to the ability to run larger applications on the cloud, while the radios can perform smaller but more rapid actions (Debroy *et al.*, 2016). Utilizing a cloud based system can amplify vulnerability detection by covering more of the attack surface however it does offer the ability to become a target to attack.

By combing SDN and cloud, it is possible to add complexity to an attacker's attempts by splitting what can be taken away from the system. Realizing a complete network which will utilize cloud based systems offers challenges in both total resource consumption and effective performance of the network. The operational cost of a system can greatly influence the true utilization of a system. If the cost to resource usage ratio is not perfect, there is an increase to potential lost capital for the service provider. In the presence of an attack, the controller will attempt to mark the attack path of IP addresses and blocking off the attack by severing the connection. The controller will either be proactive or reactive based on the advancement of the attacker in the network.

The goal of the system is to add another layer of complexity to frequency hopping by means of packet fragmentation, with the intentions to thwart two very popular attack types: denial of service and packet sniffing. An NxN Multiple Input Multiple Output (MIMO) system utilizes the ability to transmit and receive large amounts of data at a given time. Another advantage is the ability to spread the frequency spectrum across the four pairs. Using coordinated universal time, UTC, all pairs will know exactly when to hop to another frequency or request an entirely new array of frequencies

5.6 Packet Fragmentation with Frequency Hopping

The LabVIEW program is used to initialize each Universal Software Radio Peripheral (USRP) device and transmit each fragment using the algorithm displayed in Figure 5.2. Each time the program loops, the packet will be split into a pseudorandom order, and given to the transmitter. Simultaneously, the transmitter is prepped to take the packet by selecting its frequencies and reporting those frequencies to the database. It is also possible to have a varying number of total packets being sent from each transmitter. In this case, the attacker would need to determine what frequency the system is on, the total size of each packet and the order that it was sent in. At a designated time interval, all three of those parameters change to a new value. Synchronization between the transmitter and receiver can be achieved by using the database controller to store the current configuration set by the transmitter and give it to the receiver. It is also possible to use the cloud controller as the central source, so both transmitter and reciever are given all parameters. Once ready, the packet will be continuously transmitted by the USRP until all the frequencies are used. The program then halts, selects new frequencies, scrambles the packet in a new order and then continues.

5.7 3x3 MIMO Connection

A 3x3 MIMO system utilizes the ability to transmit and receive large amounts of data at one given time. Another advantage is the ability to spread the frequency spectrum across the four pairs. Using coordinated universal time, UTC, all pairs will know exactly when to

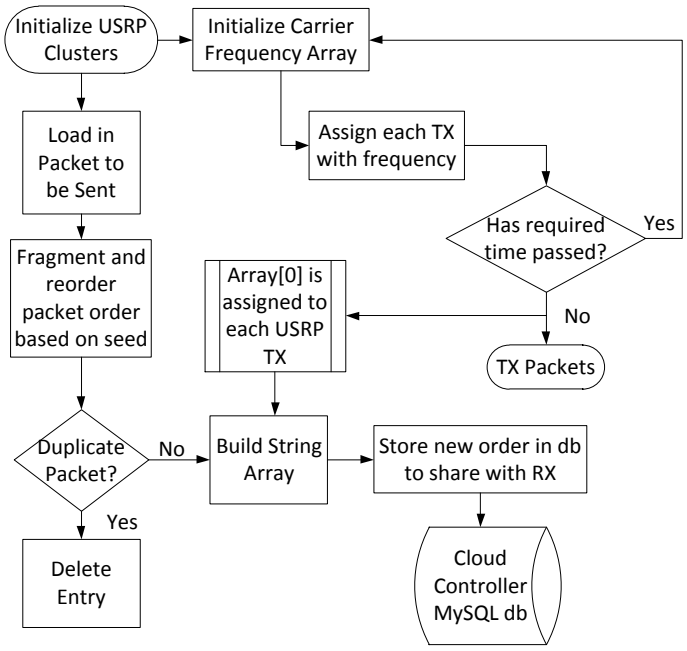

Figure 5.2: Flow Diagram for Transmission of Packet

hop to another frequency or request an entirely new array of frequencies. The system setup for a 3x3 USRP connection is shown in Figure 5.3.

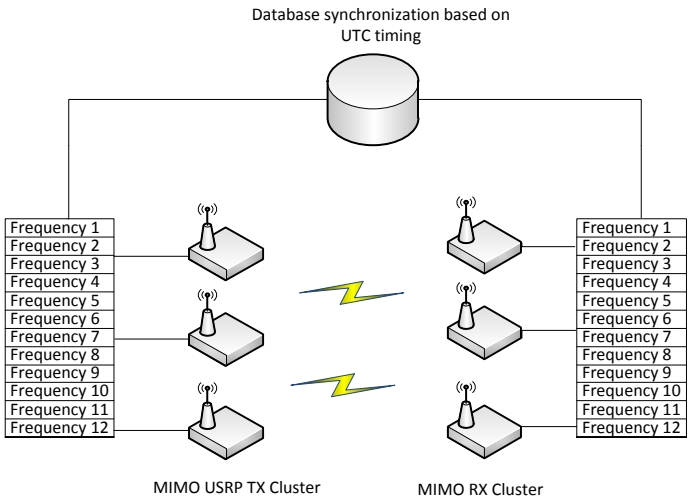

Figure 5.3: Structure of the 3x3 MIMO USRP network with Cloud Database Controller

5.8 Chapter Results

5.8.1 Experimental Design Specification

Figure 5.4 is the system setup for the Tx/Rx pair. When the experiment begins, a predefined number of allowed channels is given to the system (10, 25 or 50 channels). The frequencies are pseudo-randomly mixed and three frequencies are assigned to be transmitted on. A hopping interval is also predetermined (10, 30, or 60 sec) when the system begins. At the end of the hopping interval, the next frequency is selected. Once all three frequencies are used, the system will re-randomize the frequency list and select another three frequencies to be used.

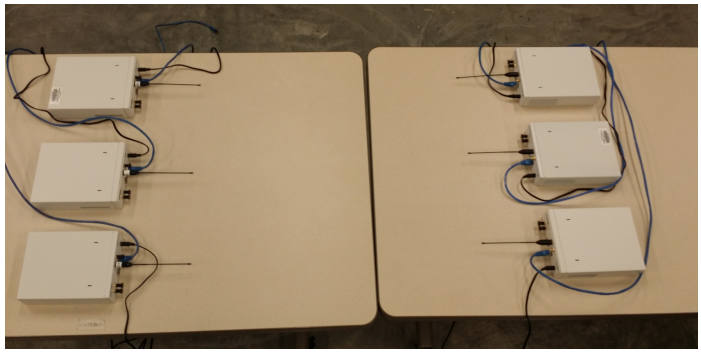

Figure 5.4: 3x3 MIMO USRP Experimental Setup

The eavesdropper will need to know or guess the possible length of the spectrum and the packet size that it is looking for. The attacker will then have to scan through the entire list of frequencies from start to finish, with the goal of trying to find the correct frequency. As mentioned in previous sections, the packet size influences the attackers hopping speed, too fast of a speed and it may miss most of the data. In a real world application, the packet

size would be considered the sufficient if the attacker knew at least how much to expect. The attack would simply stay connected to the network until it has collected enough data to estimate a successful attack. The attacker will scan the spectrum frequencies with a hopping interval of 3 sec in a incremental fashion.

5.8.2 Frequency Hopping WITHOUT Packet Fragmentation

In the first experiment, one transmitter is set up to broadcast a packet on varying frequencies to one receiver. In this case, the attacker will focus on finding the frequency that both are currently on and steal the packet. The attacker will sweep the network as quickly as possible to steal the packet. This experiment is tested on 3 trials, set to varying number of available channels. In each trial, the hopping interval of the Tx/Rx pair is set to either 10, 30 or 60 seconds. The attacker will scan the entire network 100 times, checking each channel for the packet. In this case it is a simple Hello World! message that is being sent. The relative frequency, f, of obtaining a packet is used to model the empirical probability p of successful eavesdropping using the following model:

$$f = \frac{n_c}{N} \tag{5.1}$$

where n_c is the number of captured packets, N is the total number of packets transmitted. When the total number of packet approach infinity the relative frequency will converge to model the probability of eavesdropping as follows:

$$p = \lim_{N \to \infty} f \tag{5.2}$$

The results from each pair of number of channels and hopping intervals are displayed in Table 5.1.

Table 5.1: Empirical Probability of Successful Eavesdropping using Various System Parameters

Hopping Interval (sec)	10 Channels	25 Channels	50 Channels
60	11.73×10^{-3}	4×10^{-3}	1.76×10^{-3}
30	7.33×10^{-3}	3.46×10^{-3}	1.68×10^{-3}
10	12×10^{-3}	2.08×10^{-3}	0.8533×10^{-3}

The experimental probabilities are determined by the total successful attempts by the attacker divided by the total number of packets that were sent during the transmission.

5.8.3 Frequency Hopping WITH Packet Fragmentation

The next experiment was design to test the ability of an eavesdropper to steal the entirety of the data being transmitted when it is sent broken into multiple parts and sent across three transmitters. For this trial, the network size was chosen to be 25 channels, while the hopping time intervals stayed the same from the previous experiment. As a brief side note, 25 channels was selected as good common ground. 10 channels would be considered a

low end system with a poor attack surface, whereas 50 channels potentially sees a larger cost to efficiency ratio. Therefore, 25 channels poses to be a right fit to prove a significant amount of security, without the need to span a very large network space. From the results in Table 5.2, it can be seen that when the packet is split into multiple parts, the probability of the attacker getting the entirety of the message decreases across all three of the hopping intervals.

Table 5.2: Empirical Probability of Successful Eavesdropping with/without Fragmentation for 25 Channel System

Hopping Interval (sec)	With Fragmentation	Without Fragmentation
60	2.72×10^{-3}	4×10^{-3}
30	2.187×10^{-3}	3.46×10^{-3}
10	1.76×10^{-3}	2.08×10^{-3}

CHAPTER 6

CONCLUSION AND FUTURE WORK

Although the research that has been presented throughout this document was completed as seperate entities, it is presented that the work all comes together for a fully operating software defined radio network for bandwidth allocation. It has been presented that using Universal Software Defined Radios, availability to a spectrum can be monitored and recoreded to a cloud database for secondary users to access. The method for storing and accessing this data has been presented as a three rack server farm that will allocate space for the geolocations and service the secondary user requests. Lastly, it has been presented that the software defined radio network will have vulnerabilities to attackers, which can be stopped. The packet fragmentation system set in the transmitter and receiver pairs aims to stop data from being stolen from the secondary users that are on the network. Concluding remarks are made and summarized into three separate parts, which follow respectively to the previous chapters. In each of the sections, the separate proposed methods are reiterated and the findings are presented.

6.1 Bandwidth Allocation using Software Defined Radio Networks Conclusion

An adaptive threshold based RF spectrum sensing approach using USRP Software Defined Radio (SDR) for real-time opportunistic spectrum access in cloud based cognitive radio networks where both signal energy and band-width of the signal were taken into account.

The performance of the proposed approach using probability of misdetection and false alarms was evaluated. The proposed approach can be particularized to a scenario with energy based detection or bandwidth based detection. The proposed approach is illustrated through numerical results obtained from both experiments. Signal energy detection based approach results in high false alarm since it does not consider the width of the signal spectra. Thus to avoid this it was considered both width of the signal spectra and energy level of the received signal to detect whether a given band is active or not. Once cognitive radio identifies active channels, it avoids those active channels while communicating or reports idle channels to database or uses the idle channels for opportunistic communications. It was evaluated the performance of the proposed adaptive approach using B200 GNU radios through false alarm and misdetection probabilities. By conducting experiments in different settings (walking speed and vehicular speed), it can be observed that the travelling speed of the SUs affect the opportunistic spectrum access and communications. Furthermore, short query interval results in low false list of channels for mobile devices with high speed. Furthermore, if the SUs were within the contour of idle channels, they got lists of channel, and transmitter and receiver chose a common channel for communications using quorum based rendezvous approach for opportunistic communications.

6.2 Private Cloud Cluster Controller Conclusion

Data security over the cloud/Internet is an essential part of sharing multimedia access to insure that confidential information is not stolen, distributed or destroyed. First presented

was current status and challenges in technology design and innovation in multimedia data access and storage in cloud networks. Then the existing cloud technologies and provided a comparison for potentially the most appropriate solution for implantation. It was also presented a new framework to for providing security to multimedia access in the cloud with a smart load balancer and bandwidth shaper. The possibilities for further applications for the proposed framework and how it will further benefit cloud computing networks and technology innovations as a whole. It has been shown that mobile devices will need to access the mobile cloud in order to save processing power and battery life. A simulated framework was then created in order to describe how the network will handle resources and a physical system has been implemented. Future work will then bring to use a smart load balancer to handle large data of varying degrees of security and necessity to help mobile devices operate more efficiently.

6.3 Adding a Layer of Security in SDRN: Moving Target Defense Conclusion

Frequency hopping provides a level of security to a network system, however as it is shown in the results of Table 5.1, it alone is not a full proof method. Eavesdropping has been shown to generally be an easy and very cheap method to achieve. This method adds another layer of confusion to the system, causing the attacker to work significantly harder to steal the information. This has also shown to increase the time and resources of the attacker. The results from experimentation showed that the difficulty of an eavesdropping

attacker to recover the packet increases as the system covers more of the spectrum at random frequencies.

It is also important to make note that when there are a limited number of channels for the system to exist on, like seen from the results in Table 5.1, the system pair may end up hopping too often and actually cross the attacker more frequently. This occurred with this system, when hopping between 10 channels every 10 seconds. At the same time, having the system on a very large spectrum may occur larger operational costs than are beneficial. The proposed method is able to save on spectrum space by fragmenting. The results also showed the effectiveness of the packet fragmentation method on bandwidth spectrum allocation. The probability of a successful attack when the system had 50 possible channels was significantly lower than when at 25 or 10, however to possible occupy that many channels may cause a very high operational cost. Using the packet fragmentation, the probability of success while using 25 channels instead of 50, yielded a difference of 0.96×10^{-3}, 0.507×10^{-3}, and 0.9067×10^{-3} for hopping intervals 60, 30 and 10, respectively. This showed the ability to be at comparable levels of security while existing on half as much of the total bandwidth.

Bibliography

FCC Table.

Akyildiz, Ian F., Lee, Won-Yeol, Vuran, Mehmet C., & Mohanty, Shantidev. 2006. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, **50**(13), 2127 – 2159.

Amazon. 2016. *Elastic Load Balancing*.

Bian, Kaigui, Park, Jung-Min, & Chen, Ruiliang. 2009. A Quorum-based Framework for Establishing Control Channels in Dynamic Spectrum Access Networks. *Pages 25–36 of: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. MobiCom '09. New York, NY, USA: ACM.

Cai, G., Wang, B., Luo, Y., Li, S., & Wang, X. 2016 (Jan.). Characterizing the running patterns of moving target defense mechanisms. *Pages 191–196 of: 2016 18th International Conference on Advanced Communication Technology (ICACT)*.

Chalaemwongwan, N., & Kurutach, W. 2016 (June). Mobile cloud computing: A survey and propose solution framework. *Pages 1–4 of: 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*.

Chen, J. L., Wuy, S. L., Larosa, Y. T., Yang, P. J., & Li, Y. F. 2011 (July). IMS cloud computing architecture for high-quality multimedia applications. *Pages 1463–1468 of: 2011 7th International Wireless Communications and Mobile Computing Conference*.

Corbett, C., Uher, J., Cook, J., & Dalton, A. 2014. Countering Intelligent Jamming with Full Protocol Stack Agility. *IEEE Security Privacy*, **12**(2), 44–50.

Cushman, I., Younis, A., Rawat, D. B., & Chen, L. 2016a (Feb). Adaptive threshold-based RF spectrum scanning through joint energy and bandwidth detection with USRPs in cognitive sensor networks for ROAR architecture. *Pages 1–5 of: 2016 International Conference on Computing, Networking and Communications (ICNC)*.

Cushman, I., Younis, A., & Rawat, D. B. 2016b (Jan). Experimental study of dynamic spectrum access for opportunistic mobile communications using USRP devices. *Pages 151–154 of: 2016 IEEE Radio and Wireless Symposium (RWS)*.

Cushman, I. J., Sadi, M. B. A., Chen, L., & Haddad, R. J. 2017 (April). A Framework and the Design of Secure Mobile Cloud with Smart Load Balancing. *Pages 205–210 of: 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*.

Dawoud, W., Takouna, I., & Meinel, C. 2010 (March). Infrastructure as a service security: Challenges and solutions. *Pages 1–8 of: 2010 The 7th International Conference on Informatics and Systems (INFOS)*.

Debroy, S., Calyam, P., Nguyen, M., Stage, A., & Georgiev, V. 2016 (Feb). Frequency-

minimal moving target defense using software-defined networking. *Pages 1–6 of: International Conference on Computing, Networking and Communications (ICNC)*.

Docs.OpenStack.org. 2016. *Chapter 1. Architecture - OpenStack Installation Guide for Ubuntu 14.04- juno*.

Haykin, S. 2005. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, **23**(2), 201–220.

Heinzl, S., & Metz, C. 2013 (June). Toward a Cloud-Ready Dynamic Load Balancer Based on the Apache Web Server. *Pages 342–345 of: 2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*.

Hu, H., Wen, Y., Wang, H., & Begen, A. 2016. Cloud mobile media. *China Communications*, **13**(8), iv–vi.

Huang, C. T., Qin, Z., & Kuo, C. C. J. 2011 (Oct). Multimedia storage security in cloud computing: An overview. *Pages 1–6 of: 2011 IEEE 13th International Workshop on Multimedia Signal Processing*.

Kampanakis, P., Perros, H., & Beyene, T. 2014 (June). SDN-based solutions for Moving Target Defense network protection. *Pages 1–6 of: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*.

Korotich, E., & Samaan, N. 2011 (Dec). A novel architecture for efficient management of multimedia-service clouds. *Pages 723–727 of: 2011 IEEE GLOBECOM Workshops (GC Wkshps)*.

Krautheim, F. John. 2009. Private Virtual Infrastructure for Cloud Computing. *In: Proceedings of the 2009 Conference on Hot Topics in Cloud Computing*. HotCloud'09. Berkeley, CA, USA: USENIX Association.

Krebs, R., Loesch, M., & Kounev, S. 2014 (June). Platform-as-a-Service Architecture for Performance Isolated Multi-tenant Applications. *Pages 914–921 of: 2014 IEEE 7th International Conference on Cloud Computing*.

Kulkarni, G., Mandhare, S., Bendale, D., Belsare, S., & Patil, N. 2012 (Aug). Software as Service Cloud. *Pages 442–445 of: 2012 International Conference on Computer Science and Service System*.

Leeuwen, B. Van, Stout, W., & Urias, V. 2015 (Oct.). Operational Cost of Deploying Moving Target Defenses Defensive Work Factors. *Pages 966–971 of: Military Communications Conference (MILCOM 2015)*.

Li, Nan. 2010 (April). Research on Diffie-Hellman key exchange protocol. *Pages V4–634–V4–637 of: 2010 2nd International Conference on Computer Engineering and Technology*, vol. 4.

Li, Y., Dai, R., & Zhang, J. 2014 (June). Morphing communications of Cyber-Physical Systems towards moving-target defense. *Pages 592–598 of: IEEE International Conference on Communications (ICC)*.

Liu, F., Shu, P., Jin, H., Ding, L., Yu, J., Niu, D., & Li, B. 2013. Gearing resource-poor

mobile devices with powerful clouds: architectures, challenges, and applications. *IEEE Wireless Communications*, **20**(3), 14–22.

Ma, D., Wang, L., Lei, C., Xu, Z., Zhang, H., & Li, M. 2016 (Dec.). Thwart Eavesdropping Attacks on Network Communication Based on Moving Target Defense. *Pages 1–2 of: 35th International Performance Computing and Communications Conference (IPCCC)*.

Madhumathi, C., & Ganapathy, G. 2015 (Jan). An effective time based load balancer for an academic cloud environment. *Pages 1–6 of: 2015 International Conference on Computer Communication and Informatics (ICCCI)*.

Mauri, Jaime Lloret, Ghafoor, Kayhan Zrar, Rawat, Danda B., & Perez, Javier Manuel Aguiar. 2014. Cognitive Networks: Applications and Deployments. CRC Press.

Microsoft. 2016. *What is AzureâĂŤthe Best Cloud Service from Microsoft | Microsoft Azure*.

OpenStack.org. 2016. *Software Âż OpenStack Open Source Cloud Computing Software*.

Rawat, D. B., & Yan, G. 2009a (Nov). Signal processing techniques for spectrum sensing in cognitive radio systems: Challenges and perspectives. *Pages 1–5 of: 2009 First Asian Himalayas International Conference on Internet*.

Rawat, D. B., & Yan, G. 2009b (Nov). Signal processing techniques for spectrum sensing in cognitive radio systems: Challenges and perspectives. *Pages 1–5 of: 2009 First Asian Himalayas International Conference on Internet*.

Rawat, D. B., & Yan, Gongjun. 2011. Spectrum Sensing Methods and Dynamic Spectrum Sharing in Cognitive Radio Networks: A Survey. *International Journal of Research and Reviews in Wireless Sensor Networks*, **1**(1).

Rawat, D. B., Reddy, S., Sharma, N., Bista, B. B., & Shetty, S. 2015a (March). Cloud-assisted GPS-driven dynamic spectrum access in cognitive radio vehicular networks for transportation cyber physical systems. *Pages 1942–1947 of: 2015 IEEE Wireless Communications and Networking Conference (WCNC)*.

Rawat, D. B., Amin, T., & Song, M. 2015b (April). The impact of secondary user mobility and primary user activity on spectrum sensing in cognitive vehicular networks. *Pages 588–593 of: 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*.

Rawat, D. B., Shetty, S., & Xin, C. 2016. Stackelberg-Game-Based Dynamic Spectrum Access in Heterogeneous Wireless Systems. *IEEE Systems Journal*, **10**(4), 1494–1504.

Rawat, Danda B., Song, Min, & Shetty, Sachin. 2015c. *Adaptive Resource Allocation in Cognitive Radio Networks*. SpringerBriefs in Electrical and Computer Engineering. Springer.

Rawat, Danda B., Rodrigues, Joel J.P.C., & Stojmenovic, Ivan. 2015d. Cyber Physical Systems: From Theory to Practice. CRC Press.

Selvi, S. Thamarai, Valliyammai, C., Sindhu, G. P., & Basha, S. Sameer. 2014 (Dec).

Dynamic resource management in cloud. *Pages 287–291 of: 2014 Sixth International Conference on Advanced Computing (ICoAC)*.

Sharma, N., Rawat, D. B., Bista, B. B., & Shetty, S. 2015 (March). A Testbed Using USRP(TM) and LabView(R) for Dynamic Spectrum Access in Cognitive Radio Networks. *Pages 735–740 of: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications*.

Sharma, R. K., & Rawat, D. B. 2015. Advances on Security Threats and Countermeasures for Cognitive Radio Networks: A Survey. *IEEE Communications Surveys Tutorials*, **17**(2), 1023–1043.

Su, Z., Xu, Q., Fei, M., & Dong, M. 2016. Game Theoretic Resource Allocation in Media Cloud With Mobile Social Users. *IEEE Transactions on Multimedia*, **18**(8), 1650–1660.

Taha, H., & Alsusa, E. 2015 (May). A MIMO Precoding Based Physical Layer Security Technique for Key Exchange Encryption. *Pages 1–5 of: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*.

Tai, J., Zhang, J., Li, J., Meleis, W., & Mi, N. 2011 (Nov). ArA: Adaptive resource allocation for cloud computing environments under bursty workloads. *Pages 1–8 of: 30th IEEE International Performance Computing and Communications Conference*.

Veness, Chris. 2002. *Calculate distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript*.

Wang, X., Sui, Y., Yuen, C., Chen, X., & Wang, C. 2016 (July). Traffic-aware task allocation for cooperative execution in mobile cloud computing. *Pages 1–6 of: 2016 IEEE/CIC International Conference on Communications in China (ICCC)*.

Yan, G., Rawat, D. B., & Bista, B. B. 2010 (Nov). Provisioning Vehicular Ad Hoc Networks with Quality of Service. *Pages 102–107 of: 2010 International Conference on Broadband, Wireless Computing, Communication and Applications*.

Yeung, F., Cho, P., Morrell, C., Marchany, R., & Tront, J. 2016 (Nov.). Modeling Network Based Moving Target Defense Impacts Through Simulation in Ns-3. *Pages 746–751 of: Military Communications Conference (MILCOM 2016)*.

Younis, A., Cushman, I., Rawat, D. B., & Bista, B. B. 2016 (March). Adaptive threshold based combined energy and spectrum-width detection for RF channel sensing in cognitive networks using USRP B200 GNU radios: An experimental study. *Pages 1–7 of: SoutheastCon 2016*.

Yucek, T., & Arslan, H. 2009. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys Tutorials*, **11**(1), 116–130.

Zhang, Hongli, Ye, Lin, Du, Xiaojiang, & Guizani, M. 2013 (Dec). Protecting private cloud located within public cloud. *Pages 677–681 of: 2013 IEEE Global Communications Conference (GLOBECOM)*.