Masters Project Final Report

December 2015

Project Title	Investigation of Privacy Violations in Wi-Fi Band Using Software Defined Radio		
Student Name	J.A.M.M Jayasooriya		
Registration No. & Index No.	2013/MIS/011 13770116		
Supervisor's Name	Dr.Chamath Keppitiyagama		
Please Circle the appropriate	Master's Program	Type	
	MIS	Research	Implementation
For Office Use Only			

Investigation of Privacy Violations In Wi-Fi Band Using Software Defined Radio

J.A.M.M Jayasooriya

2015

Investigation of Privacy Violations In Wi-Fi Band Using Software Defined Radio

A dissertation submitted for the Degree of

Master of

Science in Information Security

J.A.M.M Jayasooriya

University of Colombo School of Computing

2015

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Students Name: J.A.M.M Jayasooriya

……

Signature: Date:

This is to certify that this thesis is based on the work of Mr. J.A.M.M Jayasooriya under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr.Chamath Keppitiyagama

_____

Signature: Date:

Abstract

A radio communication system which is capable of tuning to any frequency band and accepts various modulations through a large frequency spectrum by tactics of a programmable hardware which is controlled by software is called as a software-defined radio (SDR) system.

Significant amounts of signal processing in a general purpose computer or a reconfigurable piece of digital electronics can be performed by an SDR. SDR is capable of providing the protocol engineer with wireless testbeds. The advantage which follows due to this flexibility is extremely subjected to the performance and usability of the particular SDR.

Through the study which I've engaged I wanted to explore the viability of using GNU Radio (an open source SDR implementation) by way of a testbed to quantify the variations of RSSI values.

My major intention of succeeding this research is to contribute an introduction to the comparably new field of Investigation of Privacy Violations in Wi-Fi Band. Even though my exploration in Gesture Recognition Systems that leverages wireless signals (ISM band signal) which has developed within these recent years, I encountered an issue with lack of research studies previously done regarding privacy violations that can happen through ISM band. Only few number of works exist which distinguish activities practicing this approach. Thus, I have given an overview over the most important tasks investigating Privacy Violations in Wi-Fi Band.

Finally, the discussion is focused on how Privacy Violations in Wi-Fi Band may profit from the current state of research and which kind of challenges need to be addressed.

Acknowledgement

First and foremost, I would like to thank my advisor, Dr: Chamath Keppitiyagama, for introducing me to this fascinating field of wireless networks, and encouraging me to seek my research topic and for all his guidance through the period of time which I was involved in my research. While providing his students a large scope of independence and flexibility to accomplish their time and projects, he is always available and enthusiastic to guide his students all the time in any ways and means possible. Whether reviewing presentations, acquiring the necessary equipment or brainstorming about projects, his enthusiasm for helping out with student's work by feeding new ideas and his genuine interest are exemplary and sincerely appreciated.

Besides my advisor, I would like to pay my gratitude to Mr.Asanka Sayakkara who was continuously providing guidance through the period of time I was engaged in my research which lead to this dissertation and played a major role in influencing this thesis. His continuous focus and supervision on the practical aspects and applications which are related to this research was an important fact which leads my project to have a good impact.

At the same time, I sincerely thank my staff members (Network Operations Centre-University of Colombo School of Computing) for their insightful comments and encouragement.

I am also grateful towards Dr: Kasun De Soyza and all the other lecturers, assistant lecturers, demonstrators and other staff members in the University Of Colombo School Of Computing who sincerely assisted me in numerous ways

Without their precious support it would not be possible to overcome this research.

Last but not the least, I would like to thank my family: my parents and to my brother for supporting me morally while I was engaged in this research, writing this thesis and my life in general.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SDR	Software Defined radio
GRC	GNU Radio Companion
RSSI	Received Signal Strength Identification
RSS	Received Signal Strength
OFDM	Orthogonal Frequency Division Multiplexing
Wi-Fi	Wireless Fidelity
AP	Access Point
WLAN	Wireless Local Area Network
CSI	Channel State Information
USRP	Universal Software Radio Peripheral
WSN	Wireless Sensor Network
DFL	Device Free Localisation
COTS	Commercially off the self
NIC	Network Interface Card
RFID	Radio frequency identification
FFT	Fast Fourier Transform

Chapter 1

1. INTRODUCTION

Over the past two decades there has been a rapid proliferation of wireless signals, such as Wi-Fi and cellular that support mobile devices. As a result, wireless signals are available everywhere in our environment, at home, at work, and etc.

The goal of this project is to handle these Wi-Fi signals around us to detect movements of particular object to investigate privacy violations in Wi-Fi band. This system allows users to predict the movement of particular object through an implemented gesture recognition system. This research focuses on using wireless signals to achieve gesture-sensing capabilities that are currently impossible, particularly gesture recognition in the home without the need for specific movement detectors (such as cameras or user-held devices). The project will address fundamental questions about the feasibility and limits of using existing wireless signals for extracting rich sensing information, and will explore deep connections between the users of the system and other moving objects in a particular environment. This work has the potential to inspire the design of new systems that can open up new research questions in multiple domains, including ubiquitous computing, mobile systems, computer networking, and human-computer interaction.

This proposed research takes a novel approach of leveraging existing wireless systems to enable novel applications such as gesture sensing and recognition. In this system it is very important to concentrate on how we capture the movements and how we predict that movement. We consider the multi-path reflections from the human body as waves from a source, then a human performing a gesture or moving, results of RSSI value variations at the wireless receiver. To detect that movement, I'm going to use a Software Defined Radio peripheral, capable of transmission or reception of radio signals of wireless communication.

Further, this research will lay the foundation and also motivate the researchers to consider about new areas of wireless security. In the modern scenarios most people consider about how to secure the connection between the user and mainly the device. By this project I will motivate others to concentrate on methods to avoid security breaches done using radio signals.

1.1 Hardware Devices

1.1.1 Infrastructure mode (Controller-based WLAN)

WLAN can be deployed by using two different architectures such as peer-to-peer mode and Infrastructure mode. Most industrial deployments are based on infrastructure mode which is commonly known as centralized controller based implementation. In this mode mainly the central device called *Controller Node* and service devices are called *Access Points*.

In infrastructure mode APs are working as a hub and stations are communicating through that wireless hub. The controller node is working as the central processing unit which control the all receiving and transmitting data.

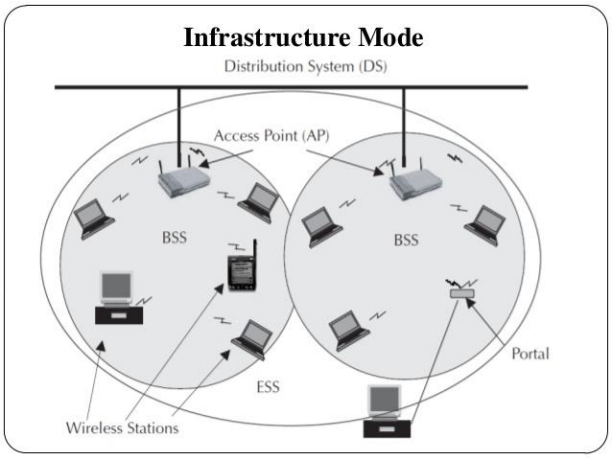


Figure 1Controller Based AP Structure & Used AP

3Com Wireless LAN Controller WX4400 based Access point.

Functionalities of the wireless controller are Association, Authentication of wireless clients who are connected through the access points, Load Balancing, Packet Routing, AP Management and Security etc. In the association function controller manage the frequency hopping with channel selecting, SSIDs distributions and client's association. As the authentication function, controller provides the authentication mechanism, protocols and authorization.

1.1.2 Access Points

Wireless Access Point is a device which helps to connect other wireless devices to the wireless network. It will work as a station by sending and receiving the data and also can serve as the interconnection point between WLAN and Local Area Network. Each access point can serve multiple users within a defined network. Also access point automatically handed over the users connected to next access point when they are roaming between defined

network areas. No of access points depends on physical size of the network and the no of the wireless clients.

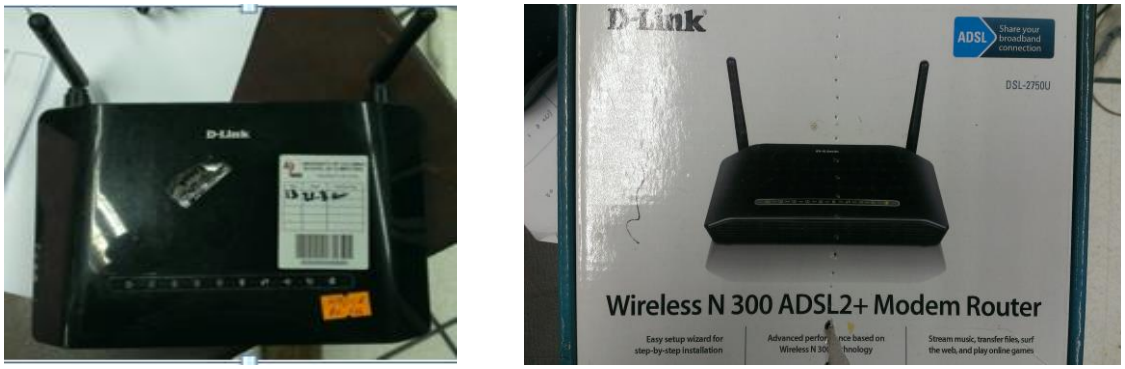


Figure 2Wireless N 300 ADSL2 Routers

First station connected with a cell, the cell sends a probe request on each channel. This request is contained with ESSID which cell is allowed to use and traffic volume that its wireless adapter can support. At regular intervals each of access points broadcast a signal known as beacon which contains the characteristics and BSSID also the ESSID by default. When probe request matched with ESSID and traffic volume information, AP sends the response to the stations with synchronization data and information on its traffic load. This is the way how the base stations and APs communicate. The distance between AP and base station, the data rate will vary because of the signal strength. At the same time clients are connected to the access points by considering the best balance of the capacity and current traffic load provide by the access point.

1.1.3 HackRF One

When the topic of modern and next generation radio frequency technologies come in to consideration, the HackRF plays an important role. Because, the HackRF One is a test equipment module for experiments and measurements used in these. It has the capability of transmitting and receipting radio signals from 1 MHz to 6 GHz as a Software Defined Radio peripheral. HackRF One is utilized as a USB peripheral or programmed for stand-alone operation since it is an open source hardware platform.

According to previously mentioned facts, the system is able to cover up a wide range of frequencies from 1 to 6000 MHz, and the coverage includes many licensed and unlicensed radio bands as well. Tuning of hardware can be reached up to an utmost sample rate of 20MS/s, which is even adequate to measure wide band signals like Wi-Fi. It supports only half-duplex operation. For the synchronization of several HackRF One boards, the system

offers connectors for clock input and output. Usage of these signals can be applied to use multiple boards in parallel, for example for measurements on MIMO systems or full duplex systems.

Figure 3 HackRF One Hardware Device

1.2 Motivation

In the modern era there are many security implementations to preserve privacy of a person when connected and using a network. Researches show when an object moves inside a wireless area that objects causes the received signal strength indicator (RSSI) to be dropped. By tracking these drops we can explain an object has moved. Other than providing internet and network access Wi-Fi is used in sensor networks to transfer data gathered from sensors. These sensor networks power and enable modern concepts like smart homes, smart power, smart cities, Internet of things etc. The usage of Wi-Fi increases rapidly and in a small arena there can be many Wi-Fi networks. Issue with this is by using wireless tomography technologies we can track movements of people unknowing to them that someone is tracking them.

Currently there are no security implementations to safeguard people for above type of privacy breaches and many are unaware about the security issues related to available tomography techniques.

My research will lay the foundation and also motivate the researchers to consider about new areas of wireless security. In the modern scenarios most people consider about how to secure the connection between the user and mainly the device. By this project I will motivate others to concentrate on methods to avoid security breaches done using radio signals

1.3 Thesis Contribution

1.3.1 Goals and Objectives

The goal of this project is to handle the signals around us to detect movements of particular object. This system allows users to predict the movement of particular object through an implemented gesture recognition system.

1.3.2 Research Question

In this study I am researching to answer two questions that arise with above mentioned situations.

1. Is there a significant amount of wireless signal strength drop when there is a human inside the line of sight of the wireless access point and data gathering computer?
2. How accurately we can identify human presence by observing received signal strength indicator (RSSI) values.

1.3.3 Scope and Limitations

This research is supposed to conduct using the indoor scenario and won't be focused about outdoor scenario. The research is aligned to convince the concept of object tracking and localization based on the RSSI value of the Wi-Fi signal using specific device called HackRF One. Localization and object tracking is only considered about the variation of the RSSI value when the signal is interfered by an object but will not take account of other circumstances such as Wi-Fi signal overlapping due to noise of radio waves and other transmitting devices, environmental variations. Also this object and data gathering device is in line-of-sight with the transmitting base station.

Chapter 2

2. Literature Survey

Device-free activity recognition solutions use the variations in wireless channel to recognize human activities in a given environment. Existing solutions can be grouped into three categories:

1. Received Signal Strength (RSS) based,
2. Channel state information (CSI) based, and
3. Software Defined Radio (SDR) based.

2.1 RSS Based

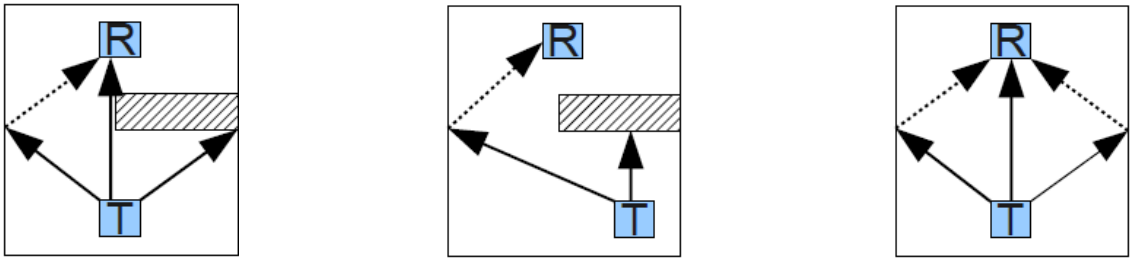
Proposed activity recognition schemes that utilize RSS values of Wi-Fi signals to recognize four activities including crawling, lying down, standing up, and walking [6, 7]. They achieved activity recognition rates of over 80% for these four activities. To obtain the RSS values from Wi-Fi signals, they used USRPs, which are highly expensive specialized hardware devices compared to the HackRF device that I used in my work. While RSS values can be used for recognizing macro-movements, they are not suitable to recognize the micro-movements such as those of fingers and hands in keyboard typing because RSS values only provide coarse-grained information about the channel variations and do not contain fine-grained information about small scale fading and multi-path effects caused by these micro-movements.

In year 2006, Woyach et al. [17] found that the localization of objects in-between nodes without wireless connectivity could be achievable by conducting RSSI-based experiments. In this experiments they observed a difference in RSSI changes by an object moving among (resulting in shadowing of signal paths) and in the surrounding area (causing small-scale fading) of two transceivers. This allowed them to identify that the RSSI inconsistency as a feature allowing understandings into the type of movement. Also fluctuations in variance were different depending on the route of an object, network topology and geometry of the surroundings.

They have found that the movement of a node in the network has a stronger impact on RSSI than the movement of an object external to the network since the latter case typically a smaller number of signal paths are affected. Woyach et al. referred to as spatial memory by

using same experiments to undermine a radio wave property. Conducting the experiment further they showed that any kind of temporary alteration in the environment the RSSI returned to the same parameter values seen prior to the alteration.

Experiments conducted by Woyach et al. regarding WSN node movement and changes in the environment.



A B C

A) A signal receiver R and a transmitter T are positioned inside a room. An object in the room is partially shadowing signals from T.

B) The transmitter is moved behind the object so that R will now only receive reflected signals. While the shadowing will affect RSSI, the movement of T to this new location will also create strong RSSI fluctuations.

C) The object is removed adding additional reflected and LOS paths between T and R. Reverting from situation C) to A) will give similar RSSI values as before.

In this very same paper accelerometer and RSSI measurements were compared and presented that in some scenarios the radio sensor seemed to be more sensitive than the accelerometer.

Zhang et al. [18] used a test environment with 870MHz WSN nodes organized in a grid and attached to the ceiling in 2.4m to study the influence of movement on radio links. They found that for each link on the ceiling an elliptical area on the floor exists for which RSSI fluctuation caused by a moving object through this area exceeds measurements in a fixed environment.

They also investigated the interconnection between received signal powers, node distance and transmit power. They found out that the impact of the objects on RSSI between 2m and 3m can regulate by adjusting the transmission power. Setting transmission power too low will

lead to weak effects on RSSI by an object. The transmission power has no influence on longer distance regulation since the transmission power cannot be increased enough to allow a considerable impact of objects on the signal. For the tested WSN nodes they have identified that valid region for detecting the impact for transceiver distances varies from 2m to 5m. As the conclusion of this experiment they have presented that adjusting the power in respect to the localization algorithm and topology can improve accuracy.

Youssef et al.[16] paid attention on the detection of people. They set up classical 802.11b Wi-Fi nodes for their experiments. They set up nodes in the corners of a room. Packets at 100ms were sent from two of these nodes while the RSSI of the received packets was recorded by the two other nodes. For detection the (features 1 moving average RSSI and 2) moving average RSSI variance were compared. Two types of moving averages were calculated for both approaches: a longer window (long-term behavior) which was used to compare against a shorter window (short-term behavior) in order to spot a change. They attained 100% accuracy with both approaches by testing among the different window lengths. Also the first localization system based on fingerprinting was presented by them in this work. Therewith they achieved an accuracy of 90% for the given setup

By creating visualizations of measurements from WSN node arrays Wilson and Patwari were able to approach DFL. They use the two-way RSSI variance [13] or RSSI mean fluctuations [14] between nodes arranged in a rectangle surrounding the monitored area as source for their radio tomographic imaging (RTI). By making use of the previously mentioned system they were able to show the robust localization of two people concurrently. In all of their experiments WSN nodes are positioned on stands in approx. the height of a human torso (in order to maximize human motion effect on RSSI).

As a method to approximate the position of a person based on RSSI variance they additionally introduced a statistical model [5].

Below mentioned simplifications are made by this model which combines two previously known radio channel models.

1) both transceivers have an omnidirectional antenna,

2) modeled effects include scattering and reflection only,

3) all scattering objects (all static obstructions which either cause scattering or reflection) are located in a single plane parallel to the ground and,

4) only a single interaction of each multipath component is modeled.

In this model the movement of an object in the observed area is causing a certain quantity of multipath power to be affected. This quantity can be supplied to the model as the measured variance on a link is approximately linearly related to the total affected power [5].

2.2 CSI Based

CSI values obtained from COTS Wi-Fi network interface cards (NICs) have been recently proposed for activity recognition [1-5,8] and localization [9,11]. WiFall that detects fall of a human subject in an indoor environment using CSI values [1]. Another proposed system about a passive human detection scheme which exploits multi-path variations for detecting human presence in an indoor environment using CSI values [3]. The Electronic Frog Eye that counts the number of people in a crowd using CSI values by treating the people reflecting the Wi-Fi signals as "virtual antennas"[4]. The system called E-eyes that exploit CSI values for recognizing household activities such as washing dishes and taking a shower [2]. Proposed WiHear that uses CSI values recognizes the shape of mouth while speaking to detect whether a person is uttering one of a set of nine predefined nine syllables [5]. While WiHear can capture the micro-movements of lips, it uses special purpose directional antennas with stepper motors for directing the antenna beams towards a person's mouth to obtain a clean signal for recognizing mouth movements.

2.3 SDR Based

Researchers have proposed schemes that utilize SRDs and special purpose hardware to transmit and receive custom modulated signals for activity recognition [12-15]. Proposed WiSee that uses a special purpose receiver design on USRPs to extract small Doppler shifts from OFDM Wi-Fi transmissions to recognize human gestures. Another system describes to use a special purpose analog envelop detector circuit for recognizing gestures within a distance of up to 2.5 feet using backscatter signals from RFID or TV transmissions [13].In 2010 Lyonnet and researchers use micro Doppler signatures to classify gaits of human subjects into multiple categories using specialized Doppler radars [14]. System called WiTrack that uses a specially designed frequency modulated carrier wave radio frontend to track human movements behind a wall [15].

WiSee [16] is the major research paper that I fallowed and it has shown how to extract gesture information from wireless signals. These systems require power-hungry ultra-wideband transceivers, interference nulling hardware, or multiple antennas. Further, they

require receivers with power-consuming analog components such as oscillators and high-speed ADCs and impose significant computational requirements such as 1024-point FFT and frequency-time Doppler profile computations. I also note that these systems were implemented on USRPs. And also this system requires quite a bit of computational processing power.

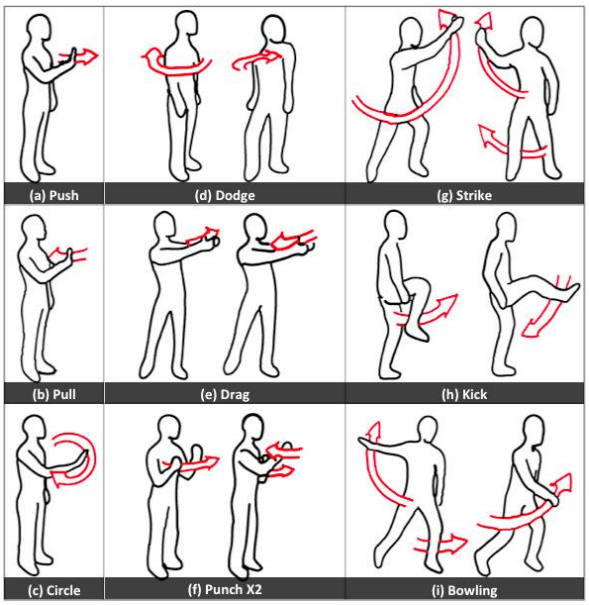


Figure 4 Identified gestures In Wisee

But in my research I used cheaper hardware called HackRF when it comparing with USRP Hardware device. And also I am using this device without any power-consuming components such as oscillators and high-speed ADCs. I am basically using only GRC with HackRF to solve my research problem.

Reschke at al [22, 23]. The results of an extended measurement campaign using machine learning based classifiers were reported. Therein two or three SDRs were installed in a medium sized room. They used one SDR as Transmitter of a sine signal which was received by the other SDRs. Carrier frequencies that were tested in between 900MHz and 2.4GHz. Features were root mean square power, signal to noise ratio and average magnitude squared. Features were calculated at 16 kHz. Tested ML algorithms were Bayes, k-NN, C4.5 and the rule and tree learners build into the Orange data mining tool. Training and evaluation was performed offline.

Examined situations and results were as follows:

Presence/Room state

For this trial the defined situations were door open/closed, empty and presence of a single person. The best achieved accuracy was 93% (2.4 GHz, three SDRs). For 900MHz accuracy was 83% using k-NN. With only two SDRs the accuracy dropped to 76% (2.4 GHz, k-NN) and 56% (900 MHz, Bayes, and C4.5).

Activity of a Person

For this trial the following activities performed by a single person were defined as classes: sit, walk and stand. Additionally, the empty room was defined. The best results were 64% (2.4 GHz, three SDRs) compared to 62% (900 MHz, three USRPs). When reducing the number of SDRs accuracies dropped to 17% (2.4 GHz, Tree/Rule Learner) and 61% (900MHz).

2.4 Used Technologies

2.4.1 Software Defined Radio

Semiconductor technology has reached up to a whole new level mostly in terms of performance capability and cost, over past decade while new radio technologies have come out from military and R&D labs and become mainstream technologies. Software-defined radio also known as Software Radio or SDR is included as one of these technologies.

The SDR Forum, functioning in association with the Institute of Electrical and Electronic Engineers (IEEE) group, has made effort to establish a definition of SDR which offers consistency and a clear overview of the technology and its connected benefits.

Briefly, Software Defined Radio can be defined as:

"Radio in which some or all of the physical layer functions are software defined “A radio can be defined as any kind of device which is capable of transmitting and receiving signals within the radio frequency (RF), part of the electromagnetic spectrum to facilitate transferring of information.

Traditional radio devices which are hardware based has limited cross functionality and can be modified only through physical involvement. This causes disadvantages such as higher production costs and minimal flexibility in supporting multiple waveform standards. Unlike this situation, software defined radio technology is capable of providing an efficient and comparatively inexpensive solution to this problem. A properly developed SDR will have the

power of navigating a broad variety of frequencies with programmable channel bandwidth and modulation characteristics

The idea behind software-defined radio is to do all that modulation and demodulation with software instead of using dedicated circuitry.

The method of extracting information in software requires the received signal to be transformed into the digital domain for processing (i.e., Analog to Digital Conversion), in contrast the transmit path does the opposite (i.e., Digital to Analog Conversion) while sending the signal to the antenna.

Digital Signal Sampling

The process of exchanging an analog signal (a function of continuous time or space) into a numeric sequence (a function of discrete time or space) is called as Sampling or digitization.

This process is done in accordance with the Nyquist Criterion, whichStates

"Exact reconstruction of a continuous-time baseband signal from itssamples is possible if the signal is bandlimited and the samplingfrequency is greater than twice the signal bandwidth."

The samples are being processed through signal processing techniques for the extraction of Information. Preferably, to process the signal fully in software, digitization should be done right after the antenna, i.e. before the RF front end and it is considered as the best approach but, as a result of the state-of-the-art of analog-digital converters (A/D) and the limitations on computational capacity of contemporary processors, implementation of this kind of digitalization is currently impossible. Digitization will be in practice at other points in the traditional

Radio architecture: after the IF (Intermediate Frequency) filter or after the demodulator at the baseband stage, since digitization or baseband digitization are not in use with Traditional radios currently. IF digitization is the solution currently implemented in SDRs.

RF Digitization

Digitizing of the radio waves collected at the antenna is done by an analog-digital converter (A/D) in RF digitization. Extraction of the information from the digital samples is done by Signal processing software. In this manner, the whole hardware based radio chain can be replaced with A/D converters, general purpose processors and signal processing software.

This path is highly formative and optimal because the same piece of equipment may be used for any new frequency, standard and application with simple software upgrades but is

restricted by the present state of the art of A/D converters and the limitations on computational scope of present processors.

IF Digitization

After the IF stage, SDR designers place A/D converters in to position. The requirement of this design is a RF front-end, which consists of a RF filter, a RF/IF converter and an IF filter. As in traditional radios, the RF front-end chooses and converts the signal in to IF. An A/D converter digitizes the signal before demodulation. Then Signal processing software module extracts the information.

Clearly Two main advantages are evident in this configuration

1- Current A/D converters can achieve enough speed and resolution at IF frequencies.
2- This design necessitates less computational assets because the tunable RF filter of the front-end restricts the number of received channels which reduces the weight of software channel selection.

Baseband Digitization

In traditional transceivers, digitization at baseband level is a common practice. Information extorted from the analog signal and baseband sampling is applied in subsequent stages to profit from signal processing techniques. Music equalization is a good example for that.

In widely used devices such a stereo music equipment, etc. this is a commonly used practice because none of the radio functions for information extraction is carried out in software. Radios using baseband digitization are not regarded as to be software defined radios, but conventional equipment with a software component to fine-tune the demodulated signal, which are phrased as Software Controlled Radio (SCR).

SDRs Architecture

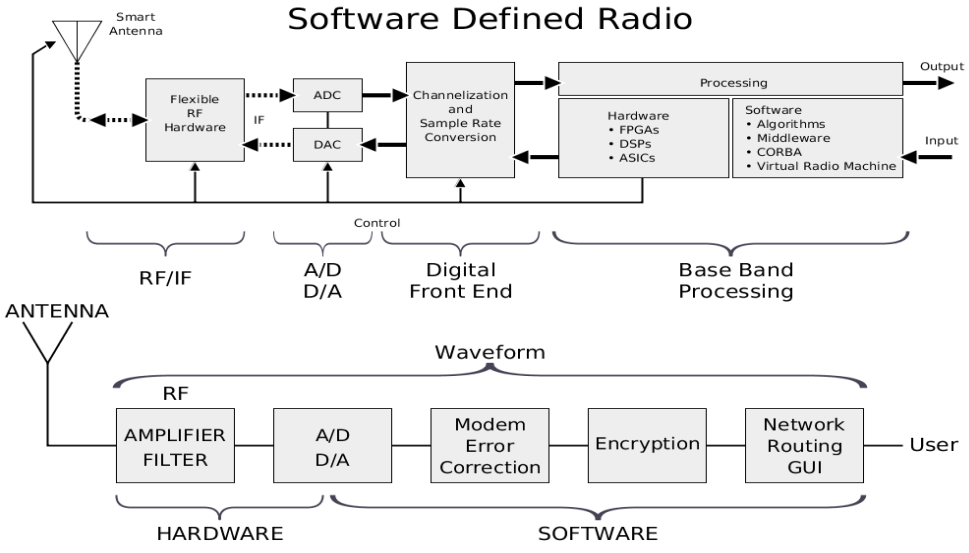


Figure 5 SDR Architecture

2.4.2 GNU Radio

GNU Radio is a package that processes signals, which is distributed within the conditions of the GNU General Public License and performs all the signal processing.

For the implementation of software and signal processing systems GNU Radio is used as a free software development toolkit that provides signal processing blocks. It is also used with external Radio Frequency hardware to form software-defined radios, or with no hardware in a simulation-like environment. Nowadays this is mostly in use of commercial environments in order to assist both wireless communications research and real-world radio systems.
This is used to write applications to receive data out of digital streams or to push data into digital streams, which is then transmitted using hardware. GNU Radio consists of elements like filters, channel codes, synchronization elements, equalizers, demodulators, decoders, and many other components, which can be usually found in radio systems. The most important fact about it is that it includes a method of connecting these blocks and then controls how data is passed from one block to another.

Only the management of digital data can be done by GNU Radio since it is software. Generally, complex baseband samples act as the input data type for receivers and the output data type for transmitters. Later Analog hardware can be used to change the signal to the coveted centre frequency.

Essentially GNU Radio applications are written using the Python programming language, while the supplied, performance-critical signal processing path is put in to action in C++ utilizing processor floating point extensions, where obtainable. As a result, the developer is capable of implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment

2.4.3 GNU Radio Companion

GNU Radio Companion can be described as an open-source Visual programming language for signal processing using the GNU Radio libraries. GRC can be fully integrated into a desktop environment that supports free desktop standards icons, mime type, and menu items.
GRC is an efficient tool for Python code-generation. When a flow graph is 'compiled' in GRC, it is capable of generating Python code which forms the desired GUI windows and widgets, and creates and connects the blocks in the flow graph.

Single executables are no longer in practice of GRC to load a file and dynamically build the flow graph. Rather Cheetah templates are used to generate the python source code for the flow graph by GRC. Generation of source codes for WX GUI GUI can be done by GRC (wxPython is a GUI toolkit for the Python programming language) and non-GUI flow graphs, hierarchical blocks too.

The Documentation extraction for GNU Radio Blocks can be directly done by GRC. In addition to that, it is capable of creating hierarchical blocks out of the built-in blocks.

2.4.4 Wi-Fi

'Wi- Fi' is the name given for the most popular wireless network technology which is in use nowadays. This is a wireless network technology which uses radio waves to provide wireless high-speed Internet and network connections. There is a fact to be noted that, 'Wi- Fi' is not the shortened form for "wireless fidelity," it is just a common misconception circulates among terms. Wi-Fi is simply a trademarked phrase that means IEEE 802.11x.

Wi-Fi is directed to use within unlicensed spectrum. Users are allowed to access the radio spectrum without the need for the regulations and restrictions that might be applicable elsewhere due to above mentioned feature of Wi- Fi.

The drawback of this aspect is that this spectrum is also shared by lots of other users and as a result the system has to be resilient to intrusion.

LOWER FREQUENCY MHZ	UPPER FREQUENCY MHZ	COMMENTS
2400	2500	Often referred to as the 2.4 GHz band, this spectrum is the most widely used of the bands available for Wi-Fi. Used by 802.11b, g, & n. It can carry a maximum of three non-overlapping channels.
5725	5875	This 5 GHz band or 5.8 GHz band provides additional bandwidth, and being at a higher frequency, equipment costs are slightly higher, although usage, and hence interference is less.It can be used by 802.11a & n. It can carry up to 23 non-overlapping channels, but gives a shorter range than 2.4 GHz.

Table 1Wi-Fi Frequency Ranges

The 802.11 workgroup currently documents use in five distinct frequency ranges: 2.4 GHz, 3.6 GHz, 4.9 GHz, 5 GHz, and 5.9 GHz bands. Each range is getting divided into a large number of channels. The key bands used for carrying Wi-Fi are those in the table below.

2.4 GHz Wi-Fi channel frequencies

CHANNEL NUMBER	LOWER FREQUENCY MHZ	CENTER FREQUENCY MHZ	UPPER FREQUENCY MHZ
1	2401	2412	2423
2	2404	2417	2428
3	2411	2422	2433
4	2416	2427	2438
5	2421	2432	2443
6	2426	2437	2448
7	2431	2442	2453
8	2436	2447	2458
9	2441	2452	2463
10	2451	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483
14	2473	2484	2495

Table 2 2.4 GHz Wi-Fi Channel Frequencies

2.4.5 Privacy

Someone's right to be isolated can be briefly defined as Privacy. It is a basic and comprehensive human right and it is the right most treasured by free and independent people. If there was a society which privacy conditions and levels hit rock bottom, would be unbearable; then again if there was a society which had plenty of privacy conditions and levels then there would never form a 'society' at all. Therefore the balance between extremes and minimums are needed in privacy levels. Privacy is the right of individuals to make personal decisions regarding their own intimate issues, it is the right of people to lead their lives in a manner that is reasonably secluded from public scrutiny, and it is the right of people to be free from such things as unwarranted drug testing or electronic surveillance.

What is information privacy?

Information privacy is the capability of an individual or group to stop leaking information about themselves to a third party and restrict becoming known to people other than those they decide to provide the information to. Privacy is sometimes related to anonymity although it is often most highly valued by people who are publicly known.

2.5 Measurements

2.5.1 OFDM

OFDM is the shortened form for Orthogonal Frequency Division Multiplexing. OFDM is a subset of frequency division multiplexing in which a single channel utilizes multiple sub-carriers on adjacent frequencies. Furthermore, overlapping of the sub-carriers in an OFDM system increases the spectral efficiency. Normally overlapping adjacent channels is causing interference with one another. Nevertheless, sub-carriers in an OFDM system are precisely orthogonal to one another. As a result, overlapping among them occurs without any interference. Because of that, OFDM systems are able to maximize spectral efficiency without causing adjacent channel interference.

OFDM has been adopted in the Wi-Fi arena where the standards like 802.11a, 802.11n, 802.11ac and more. Some definite advantages in terms of data transmission, especially where high data rates are needed along with relatively wide bandwidths are provided by that.

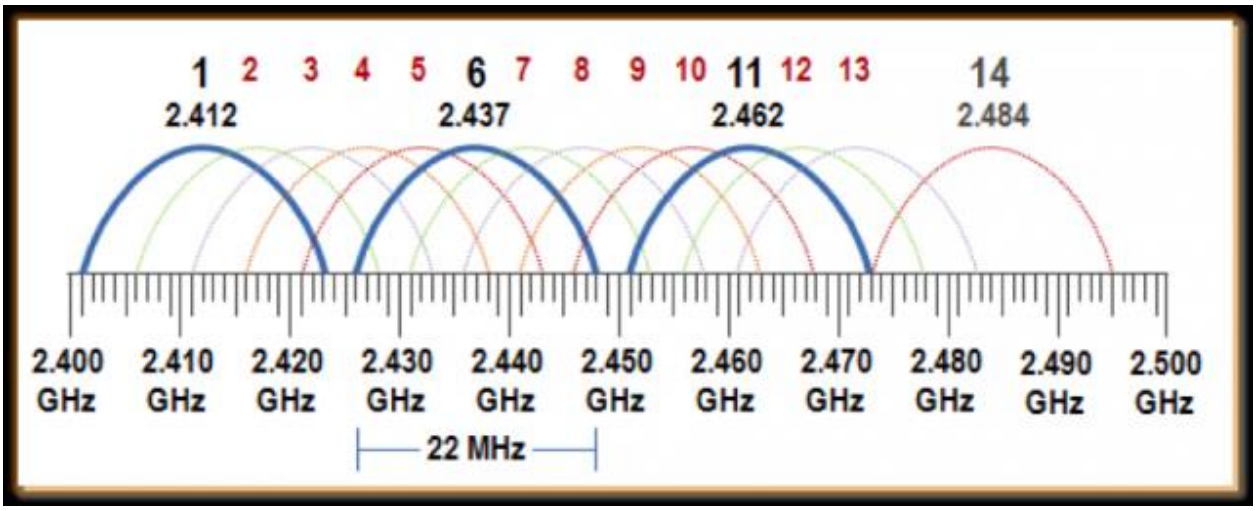


Figure 6Non over lapping OFDM Signals

2.5.2 RSSI

The term ‘Received Signal Strength Indicator’ is abbreviated as RSSI. It is the strength of the beacon's signal as displayed on the receiving device. The strength of the signal is dependent on the distance and Broadcasting Power value. At maximum Broadcasting Power (+4 dBm) the RSSI ranges from -26 (few inches) to -100 (40-50 m distance).

RSSI tends to fluctuate due to external factors influencing radio waves like, absorption, interference or diffraction. The further apart the device is from the beacon, the more unsteady the RSSI turns out to be.

For the RSSI reading there is no standardized affiliation of any particular physical parameter. The 802.11 standard does not define any relationship between RSSI value and power level in mW or dBm. Vendors and chipset makers offer their own accuracy, granularity, and range for the actual power and their range of RSSI values

The signal strength is represented by both dBm and RSSI which are different units of measurement. The only dissimilarity is that RSSI is a relative index, while dBm is an absolute number representing power levels in mW (milliwatts).

Chapter 3

3. Design and Implementation Phase

3.1 Design Phase

Electromagnetic waves with a frequency ranging from 3 kHz to 300GHz are called as Radio waves. When Electromagnetic waves are propagating through space, there are numerous chances for these waves to get affected by various effects. The consequence of these effects mostly is a function of signal frequency, transmission medium and objects come across during propagation.

Aforesaid effects consist of

- Reflection (when the wave partially bounces off an object),
- Refraction (change of direction when passing from one medium to another),
- Absorption (loss of energy when an object is hit),
- Diffraction (when waves are bend and spread around an obstacle),
- Scattering (wave bounces off in multiple directions) and
- Polarization (orientation of the oscillations of the waves can change upon interaction).

In addition, in free space electromagnetic waves are following the inverse-square law

"The power density of the signal is in inverse ratio to the square of the distance from the transmitter"

The multi-path propagation is another aspect to be considered in radio wave propagation. Usually antenna of the transmitter discharges radio waves in various directions (e.g. Omni-directional antenna) or angles (e.g. directional antenna). In addition some of above mentioned effects tend to take place at the same time. As an example, there is a possibility for a radio wave to propagate through an object but part of it will be reflect on its surface and some of its energy is captivated by the object. This interplay causes to produce at least two signals which are propagating in different directions but originating from the same source. Radio signals that are generated from the same source, which reach the receiver by two or more paths, are

labeled as multi-path signals or components. In general the summation of some of these destructive or constructive multi-path components is the power of the received signal.
The power of the received signal is indicated usually by a single parameter available in WSN. This is called as the received signal strength indicator (RSSI). In general it is a scalar value. This parameter can be accepted as an intricate function of the above described effects more than the course of the signals through space until entering the receivers' antenna.

Data gathering is a key step in this research and scenarios are used to simulate real world examples. Then based on the data we do the analysis and determine the results of the project.

In my research I mainly concentrated on carryout the process in four main steps. The main reason for widening the research for many steps is due to what I observed in each face as well as the obstacles faced that prevented from achieving the expected results.

Scenario Diagrams

In these diagrams we have used symbols to represent objects below table describes the symbols.

Symbol	Resemblance
	Wireless access point
	Data gathering computer (HackRF One)
	Human
	Moving Human all over the area
	Moving Human (line of sight)

Table 3Symbols and Resemblance

3.1.1 Phase 1

- I started capturing the data initially in a closed room at (15ft*15ft) roof top in UCSC premises. The main reason for selecting that particular room is that it's a place where there are no Wi-Fi signal interferences in that room. So I could make sure that only Wi-Fi signal which is generated is by the source I used.

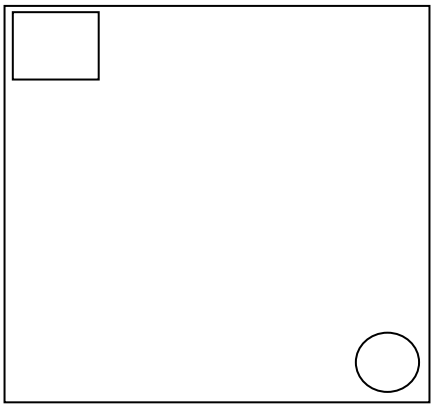
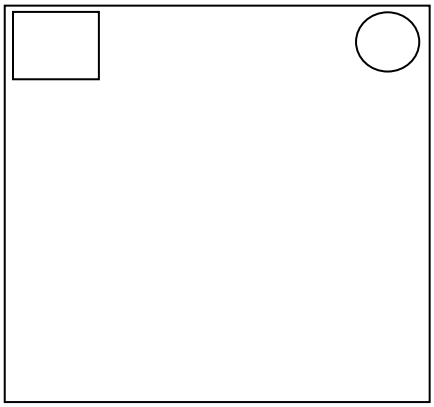
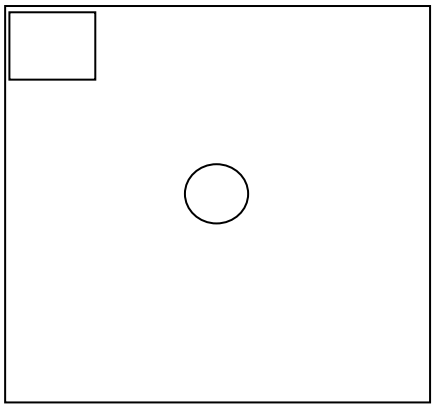
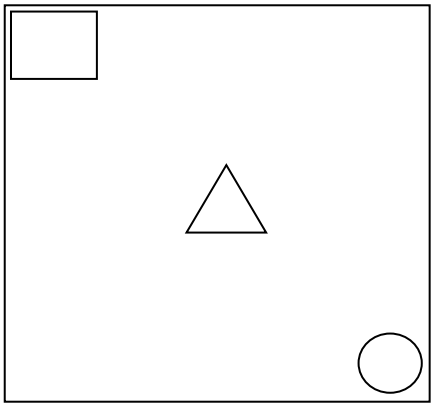
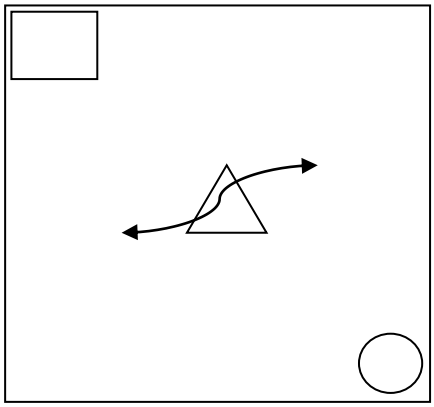
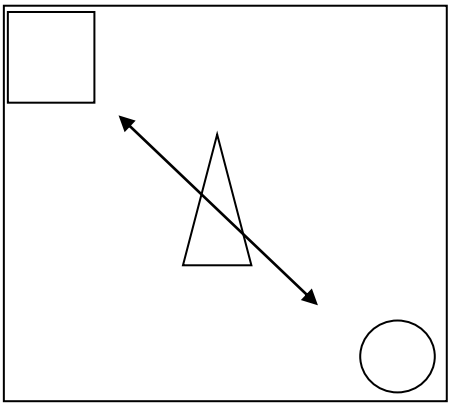
This step was carried out by using;

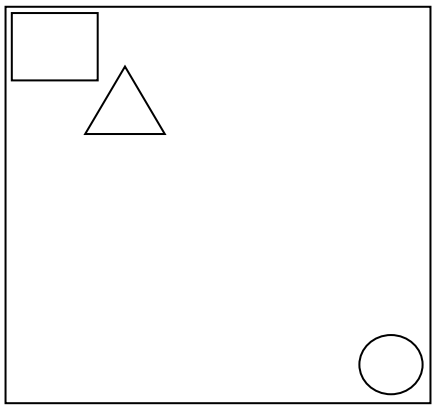
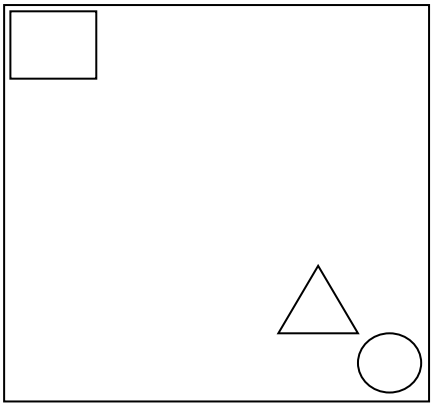
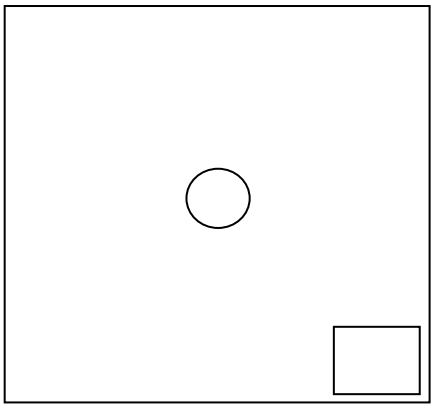
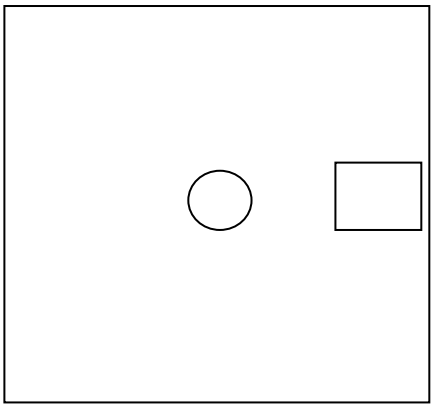
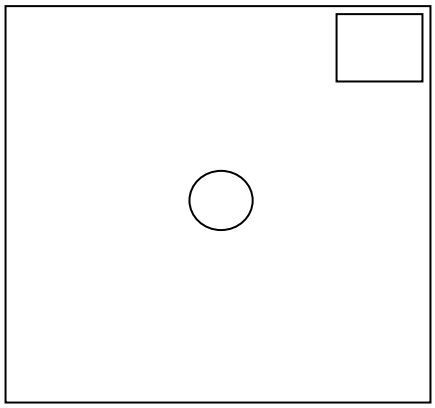
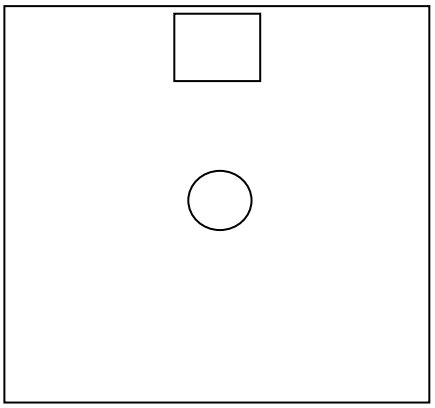
1. HP ProLiant DL380 G5 Server (4TB hard disk, 24GB RAM)
2. Desktop Computer
3. HackRF One
4. 3com Wireless Access Point (Controller Based Access Point)

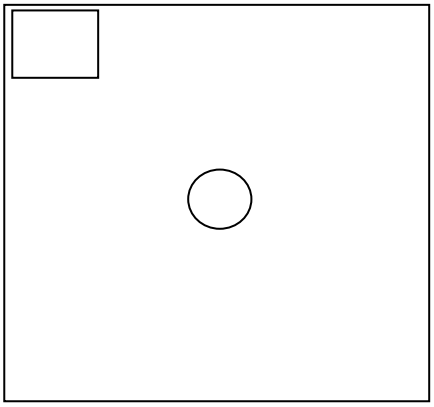
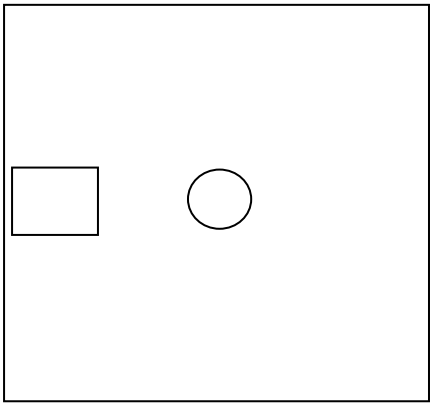
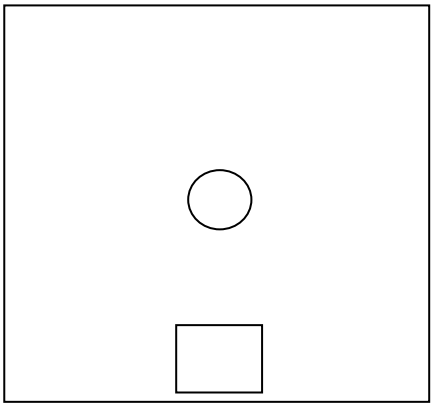
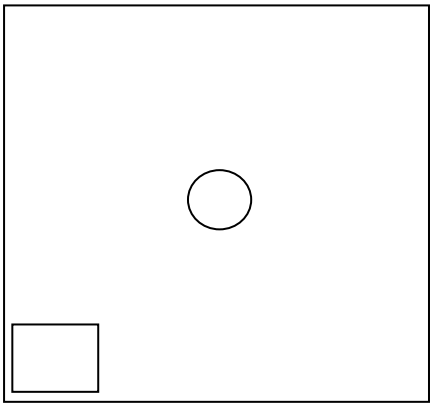
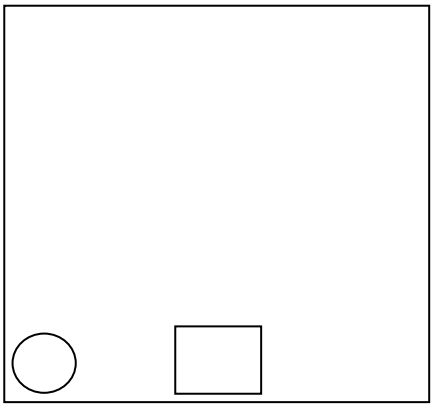
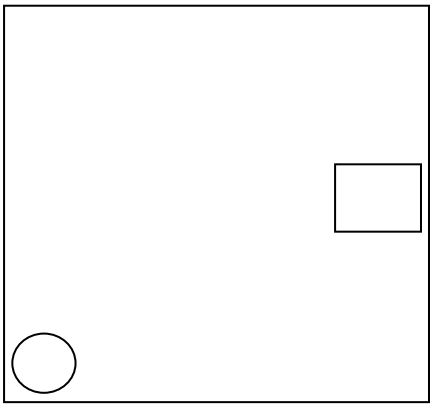
Figure 7Phase 01 data gathering

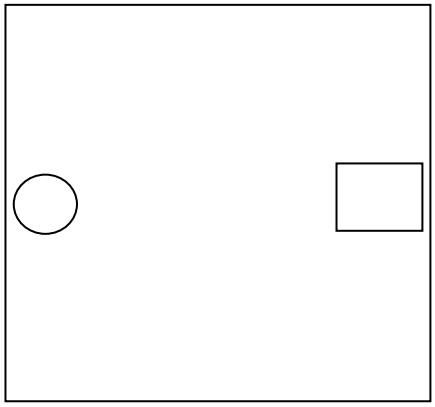
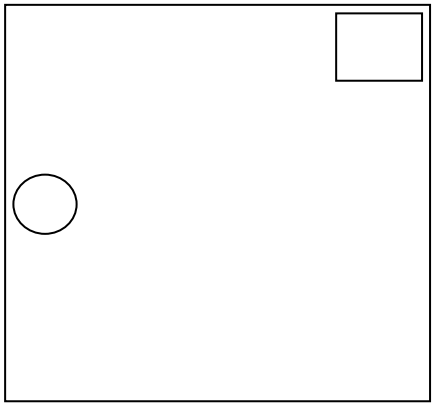
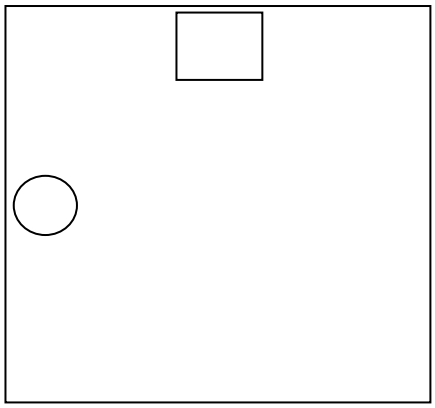
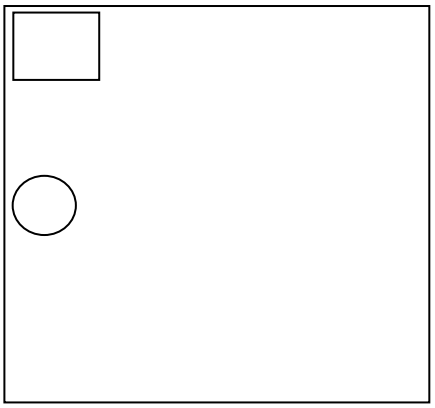
I carried out this step under 22 scenarios where I made slight changes to the position of equipment and human movement. The main expectation in that phase is that I was hoping to figure out if there is any variation of the RSSI value that I captured in each scenario.

Here are the 22 scenarios that I did in my data gathering.

Scenario 01 	Scenario 02 
Scenario 03 	Scenario 04 
Scenario 05 	Scenario 06 

Scenario 07 	Scenario 08 
Scenario 09 	Scenario 10 
Scenario 11 	Scenario 12 

Scenario 13 	Scenario 14 
Scenario 15 	Scenario 16 
Scenario 17 	Scenario 18 

Scenario 19 	Scenario 20 
Scenario 21 	Scenario 22 

The results that I observed and the obstacles faced led me to extend my research to phase 2.

3.1.2 Phase 2

- The drawbacks and difficulties that I faced and also with the observations I made in the earlier phase led me to make some important alterations to data gathering. Among the changes that I did are the follows.

1. Changed the place of data gathering.

   With what I observed in the earlier phase I decided to alter the environment. This time I selected a large scale room. (33ft*15ft). The reason for selecting a large room is that in the earlier stage by changing the locations of the devices did not provide the expected variation in RSSI value.

2. Changed the wireless signal transmitting device.

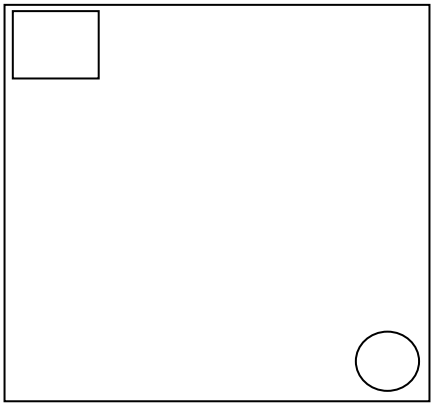
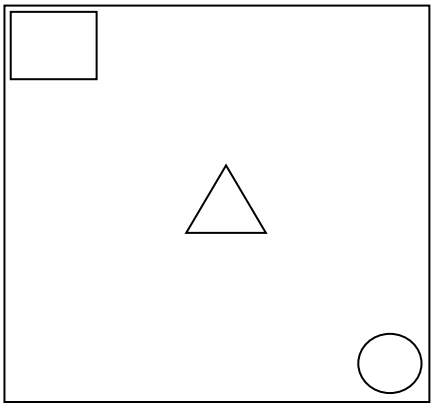
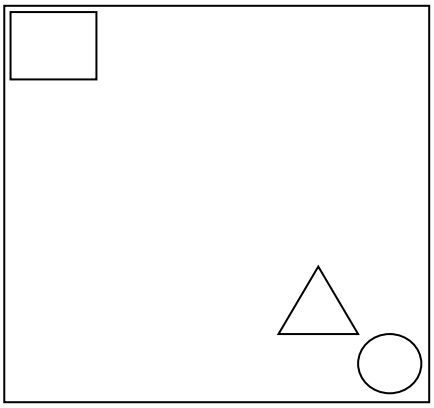
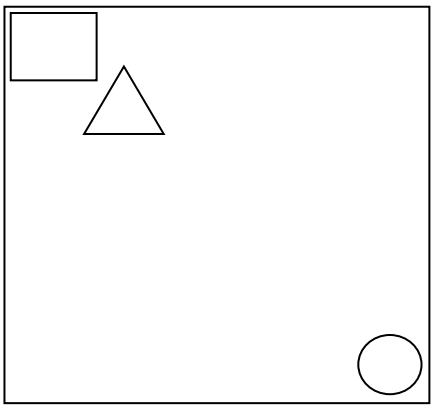
   I decided to change the source device to a wireless router rather than using a controller based AP. The main reason for omitting the controller based AP is that I could not control the signal strength, OFDM and many other features of wireless signal because it is generalized at the university premises. I could not change those features on that particular device on that exact location.

   Furthermore this time I used a static frequency range (channel 1, 2.412 GHz) and also I limited the signal transmission rate to 20%.

3. Made a human movement inside the premises and observed the results.

   By making a human movement in between the transceiver and receiver the main expectation was that a huge interference between those two devices could make a significant change in the RSSI value.

Under this phase we collect data under 4 scenarios.

Scenario 01 	Scenario 02 
Scenario 03 	Scenario 04 

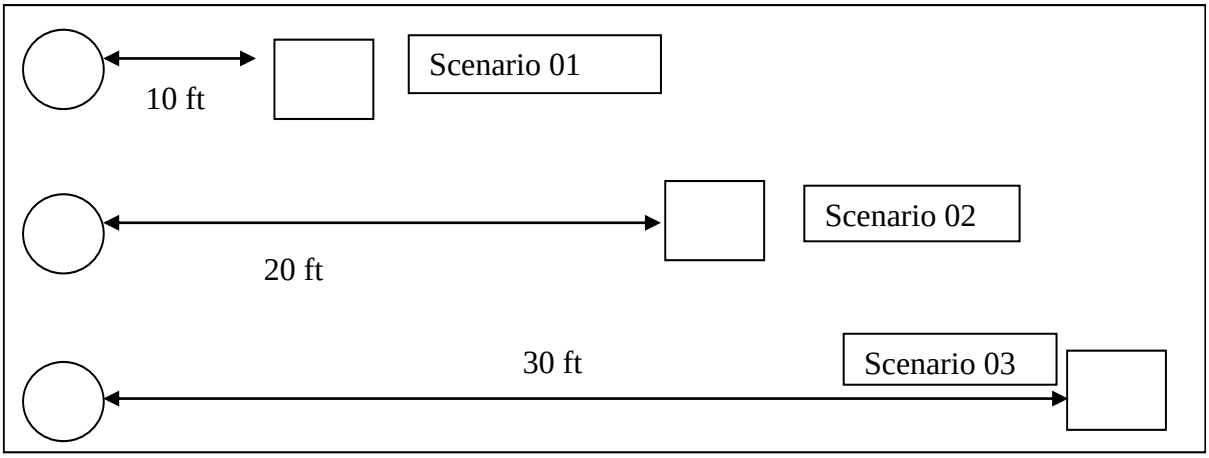
The results of this phase convinced me to make another few alterations to the methodology that I was following.

3.1.3 Phase 3

- In this stage the major difference I made compared to the earlier phases is that;

1. Changed the size of data capturing area. (35ft*15ft)
2. I made some changes to the python script that I used. This is done to identify whether there is any fault in the code and by making necessary changes in the script whether I could achieve the expected results.
3. Changed the scenarios used to capture data& data extraction method.
   This time the method I used to capture the data was to increase and decrease the distance between the transceiver and receiver. By doing that what I expected was to figure out whether there is any variation in the RSSI value.

   Here are the scenarios that I used to capture data in this particular phase.



3.1.4 Phase 4

- Since the expected results were not achieved I decided to make another few alterations to the methodology.

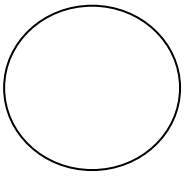
1. I removed the antenna of Hackrf One. I did this to clarify whether there is any fault in the device.
2. I removed the router and checked the results. There was no output

After carrying out all these processes I could come to a conclusion of what are the results and how they have happened.

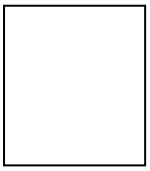
3.2 Implementation

Under implementation stage, gathering data is done according to the designed scenario. First of all, we collect raw data before calculate RSSI values. Because there is no way to collect RSSI values at once by using HackRF One device with SDR.
This is done using two python scripts; one is SDR generated one and other one is hardcoded. All the setup of the test bed environments are university rooms. Under the first phase, by doing all the scenarios I tried to capture some variations of RSSI values. And after that we mainly focused our attention on gathering data with line of sight by a wireless access point and the HackRF One. These sets will act as control scenarios in the analysis stage to show the distribution when there is not human present in the room.

Control Scenario – All the other scenarios are compared with this scenario

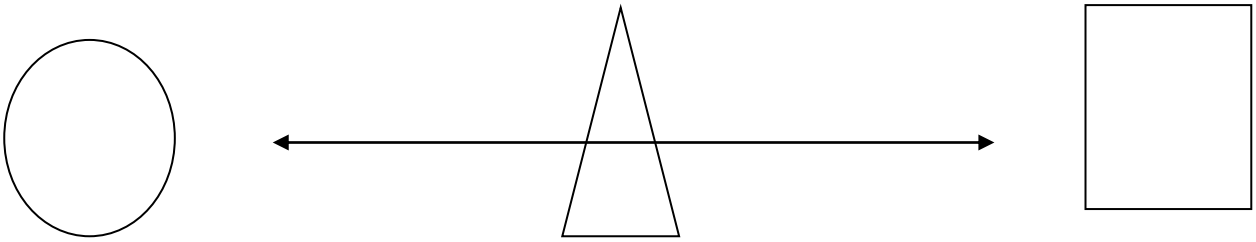


Wireless access point



HackRF One Device

line of sight to both devices, hence obstructing the direct line of sight of both devices.



Wireless access point

Human (moving object)

HackRF One

Data gathering is done for a 15 minutes time period. As mentioned earlier, initially we are gathering the raw data in complex data format (non-human readable) for 15 minutes for each scenario. Data gathering part is done by using HachRF One device combining with GNU Radio Companion. Then, these data are saved to a text file for future analysis. Root permission or super user privileges are needed to the run the scripts.

Below screenshots show the running stage of the designed flow graph of GNU Radio companion while gathering data and visualizing data in real-time.

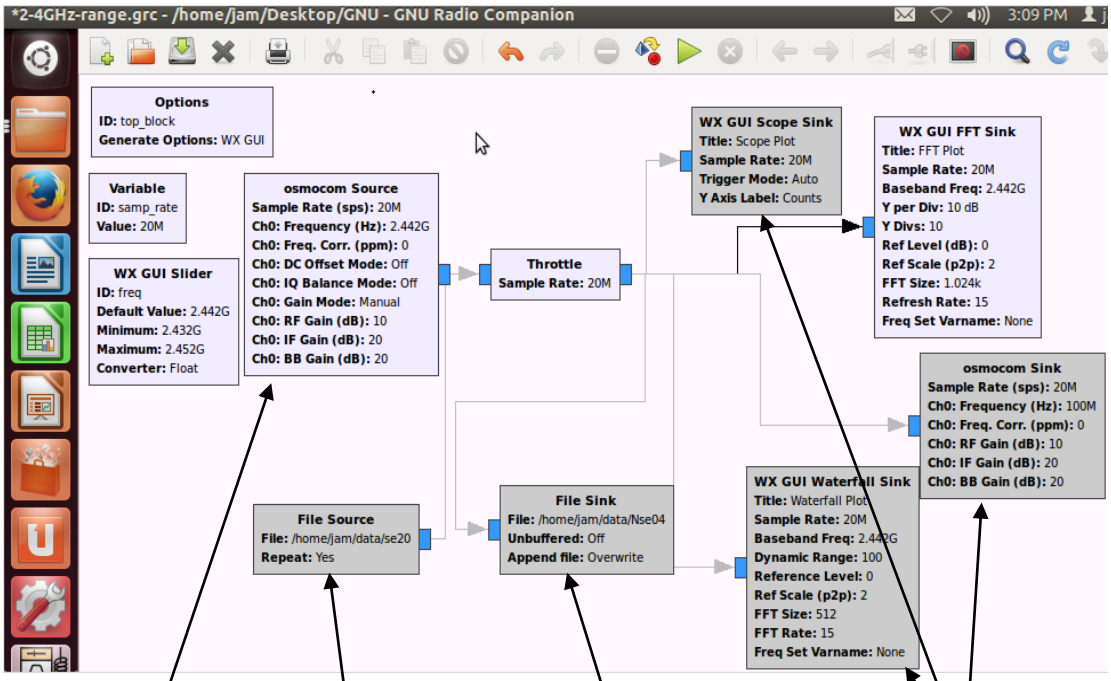


Figure 8Created GRC Flow graph for data gathering

HackRF One Device

We can give stored data as input data

Captured data by HackRF One is stored here

Visualizing data in real-time

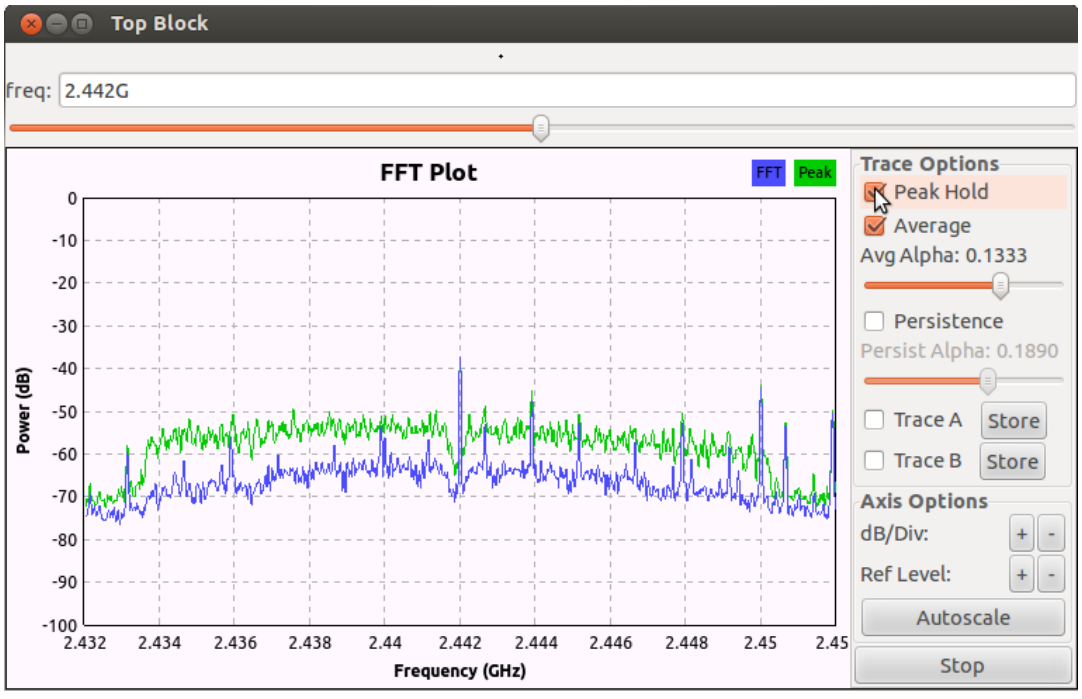


Figure 9Real time FFT plot

Chapter 4

4. Analyse and Evaluation

In the stage we will processed the gathered data in the previous mentioned stages to obtain a significant output.

15-minute raw data set is around 150 GB capacities.
Total row data storage

Stage 01 – 22 Scenarios (22 x 150 = 3300 GB)
Stage 02 – 3 Scenarios (3 x 150 = 450 GB)
Stage 03 – 4 Scenarios (4 x 150 = 600 GB)
Stage 04 – 2 Scenarios (2 x 150 = 300 GB)

Total Data Storage actually around 4650 GB

Stored data can't be read at once by using GNU Radio Companion. Because of that we had to create another flow graph to generate a python script through the GRC.

It will acquire data from a HackRF device and apply a Fast Fourier Transform (FFT) operation on the data stream. Finally, it turns that FFT data in to message block which are ready to be sent out of the flow graph though a special block. I saved this flow graph as **fft_data.grc** and clicked on the 'generate flow graph' button to generate the python script which implements this flow graph is code. It generates the script called **top_block.py(**Python script appendix 01**)** automatically.

Once this is done, we have a python script which will acquire data and provide us FFT data . For our purpose, create a new python program and add the generated python script as an import script to newly created python script.

Below screenshot shows that flow graph.

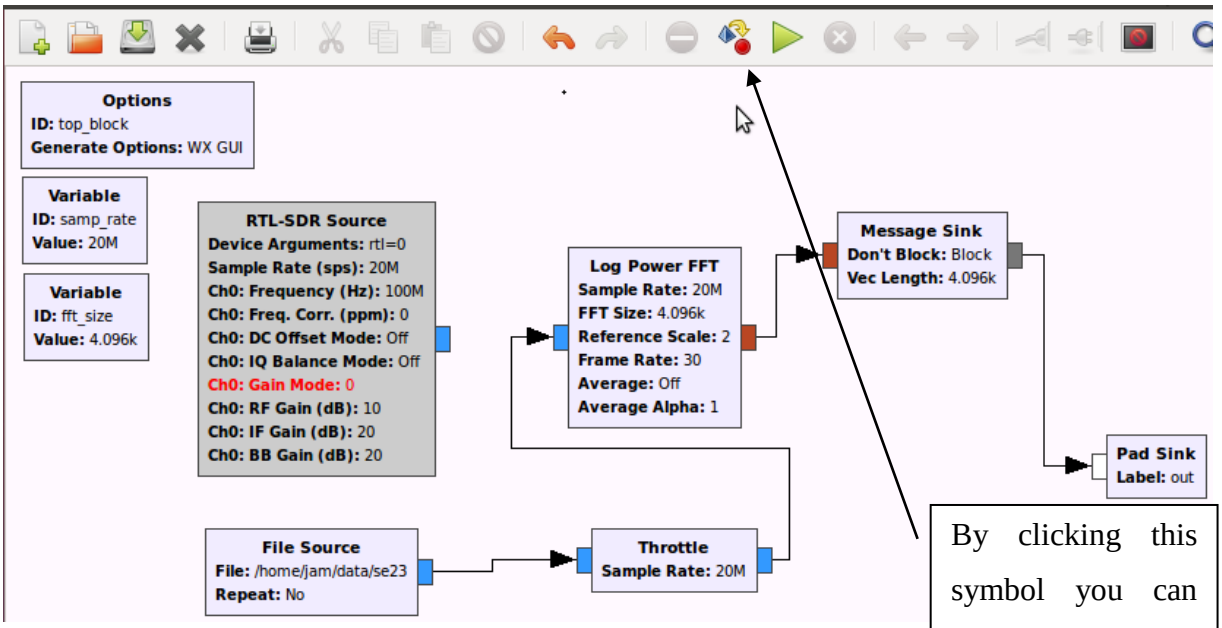


Figure 10GRC Flow graph for generate python script

Analysing FFT Data

There is no way to calculate RSSI values from HackRF One with GNU Radio Companion. But the thing is FFT values gives the Decibel values of separated frequencies.

Decibel value is not a constant every time. It keeps on changing throughout the data gathering period as RSSI.

When the HackRF One device is tuned into 2.412GHz Frequency distribution is 2.402 GHz – 2.422GHz. (Signal transceiver is also tuned into 2.412 as we mentioned earlier). 2.412GHz is the frequency Chanel No. 1 in the non over lapping Wi-Fi OFDM signal distribution. This frequency distribution consists of 20 million frequencies. Because of lack of the computational power it is really hard to extract all the decibel values of all the frequencies. Because of that we create the python script to extract 4096 decibel values of equally divided frequencies from that frequency range. (Python script Appendix 02)

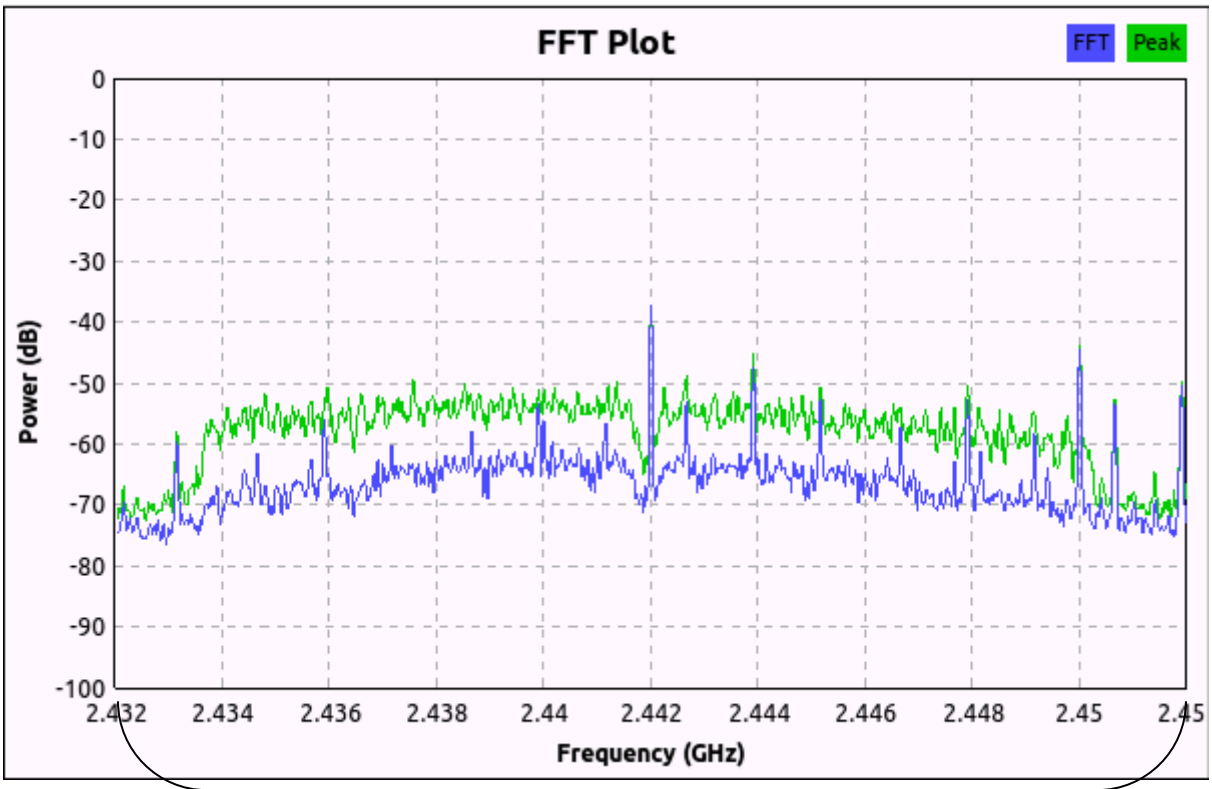


Figure 11 FFT plot for discribe frequency range

Range 2.432 GHz – 2.452GHz = 20 million frequencies.
These 20 million frequencies divided into 4096 frequencies

4.1 Phase 01

- Under the first phase we capture the decibel values of middle frequency of that distribution (Python script appendix 02). I used a statistical approach to analyze and evaluate the gathered data in hoping to find answers to the first research question.

First all the data is plotted to normal distribution graphs of the decibel values. So to analyse we need to have an overview of how decibel values are being distributed with respect to each scenario. Below are the normal distribution graphs with respect to their scenario. Axis of the graph will carry below information

- X axis = Signal Strength in db
- Y axis = Density of probability

Distribution graphs for scenarios

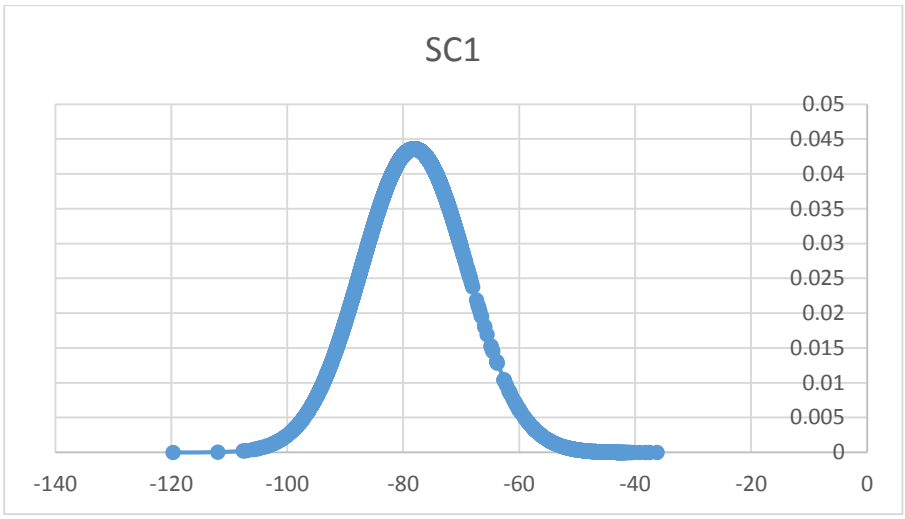


Figure 12 Decibel value distribution graph for scenario 01 phase 01

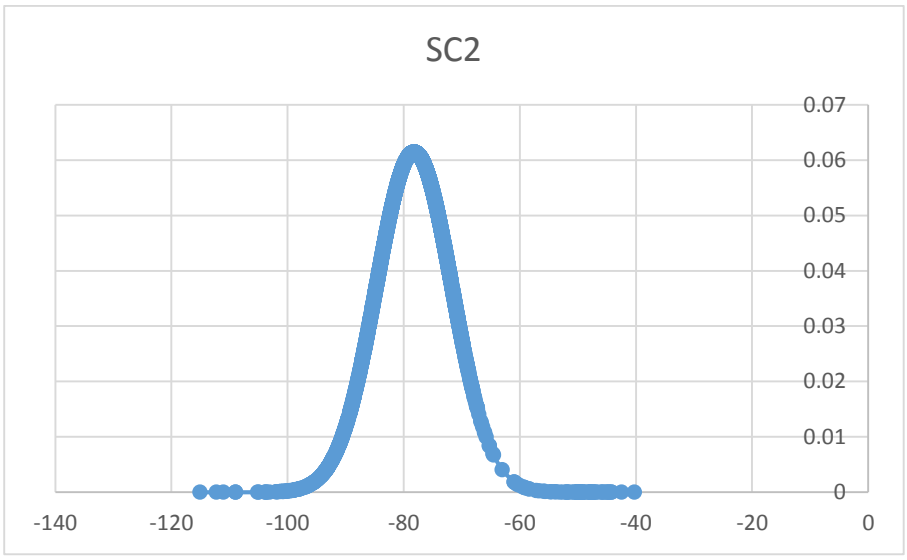


Figure 13Decibel value distribution graph for scenario 02 phase 01

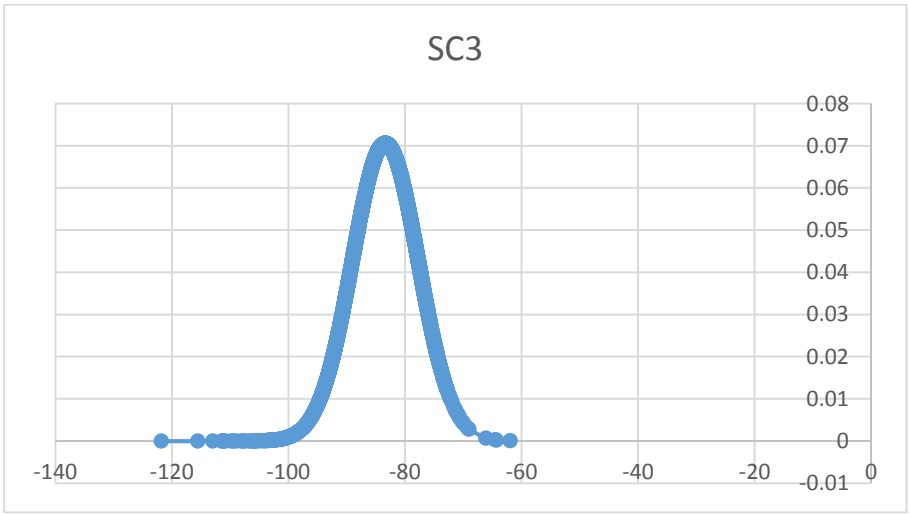


Figure 14Decibel value distribution graph for scenario 03 phase 01

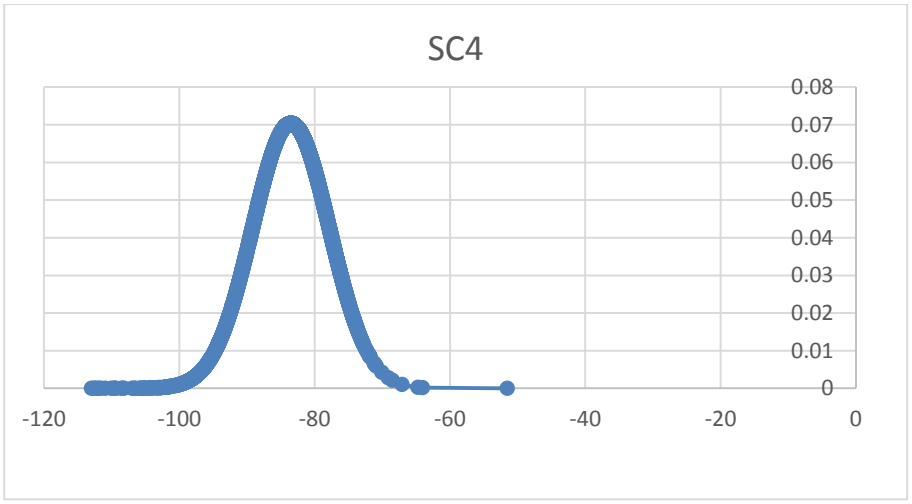


Figure 15Decibel value distribution graph for scenario 04 phase 01

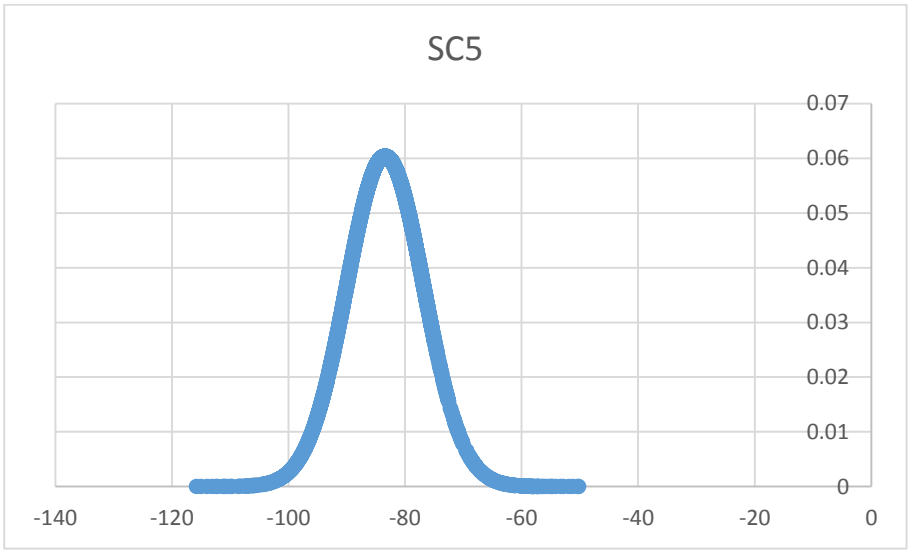


Figure 16Decibel value distribution graph for scenario 05 phase 01

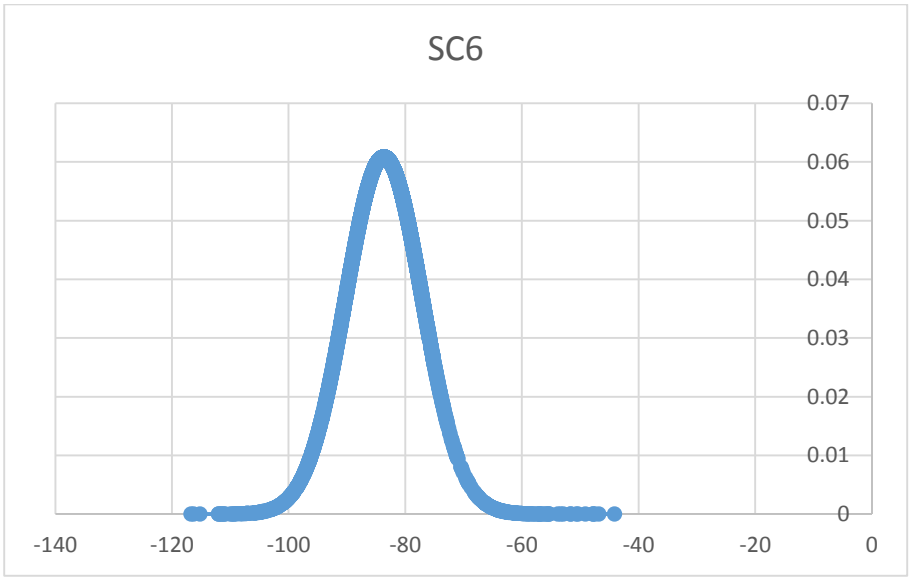


Figure 17Decibel value distribution graph for scenario 06 phase 01

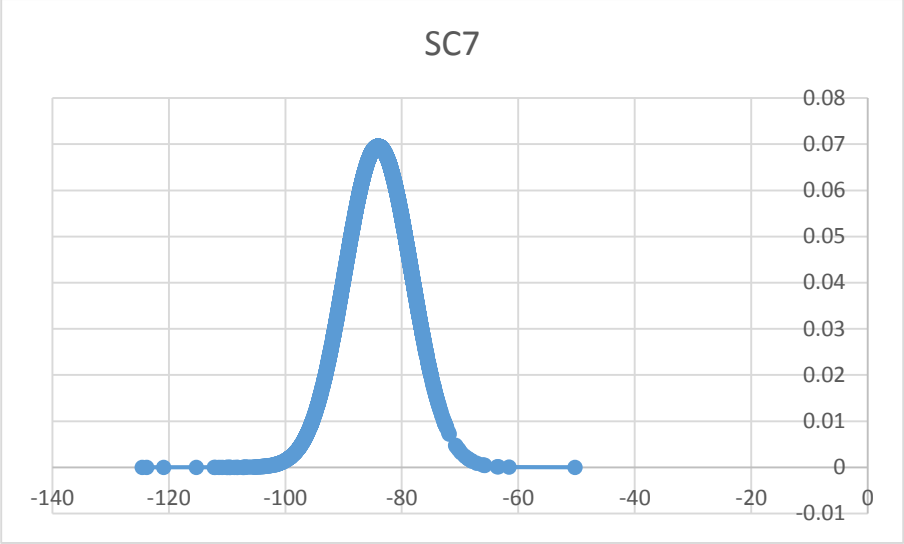


Figure 18Decibel value distribution graph for scenario 07 phase 01

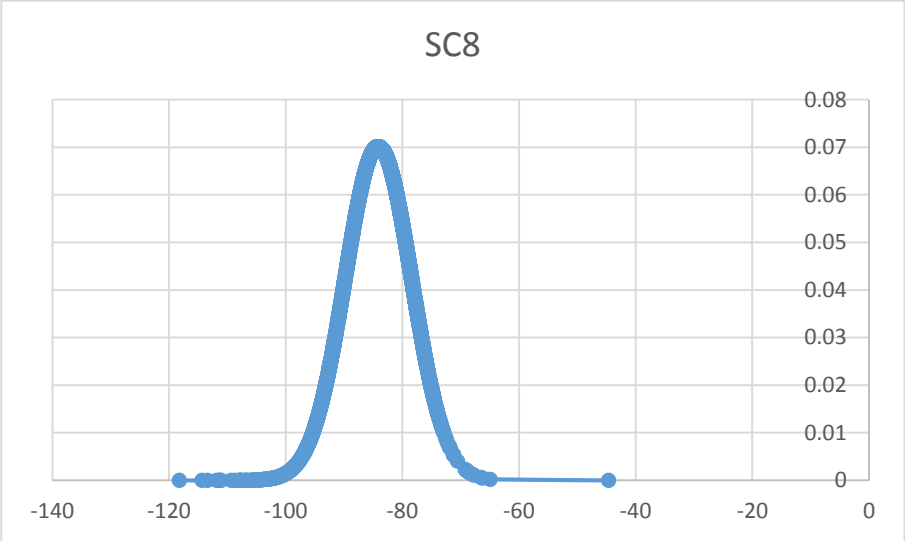


Figure 19Decibel value distribution graph for scenario 08 phase 01

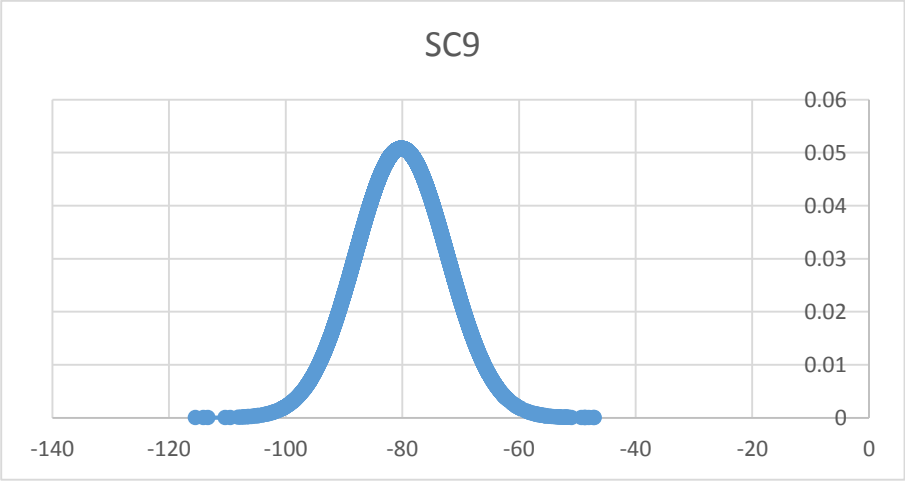


Figure 20Decibel value distribution graph for scenario 09 phase 01

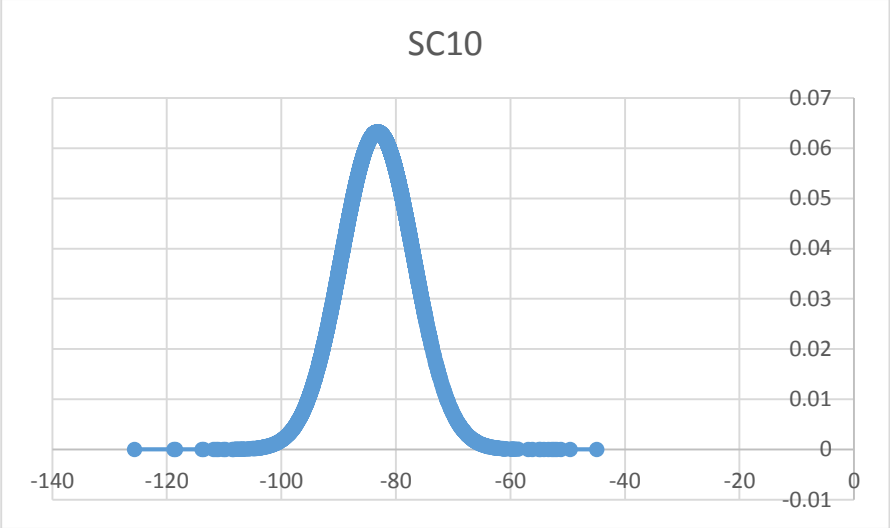


Figure 21Decibel value distribution graph for scenario 10 phase 01

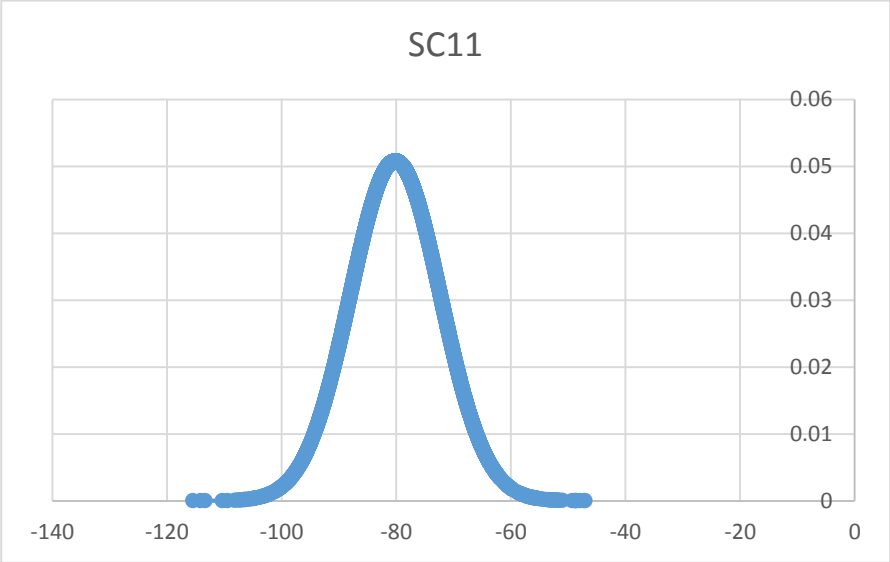


Figure 22Decibel value distribution graph for scenario 11 phase 01

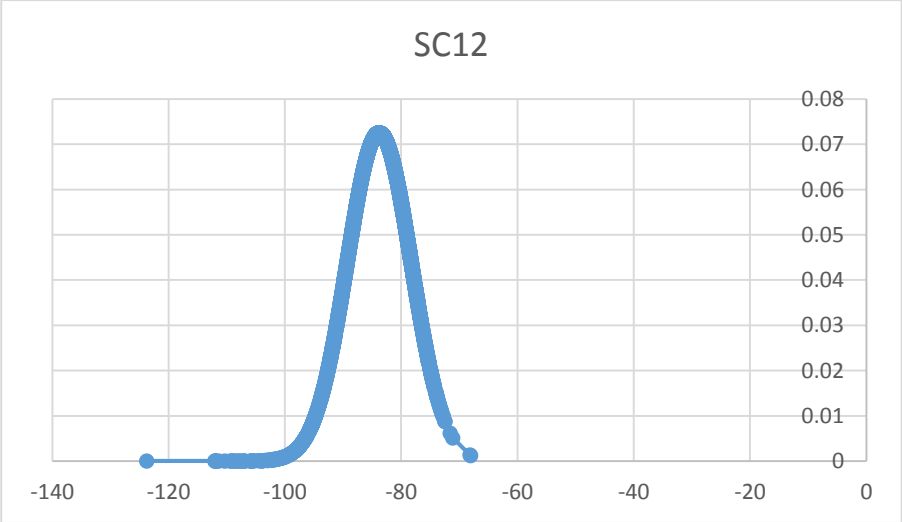


Figure 23Decibel value distribution graph for scenario 12 phase 01

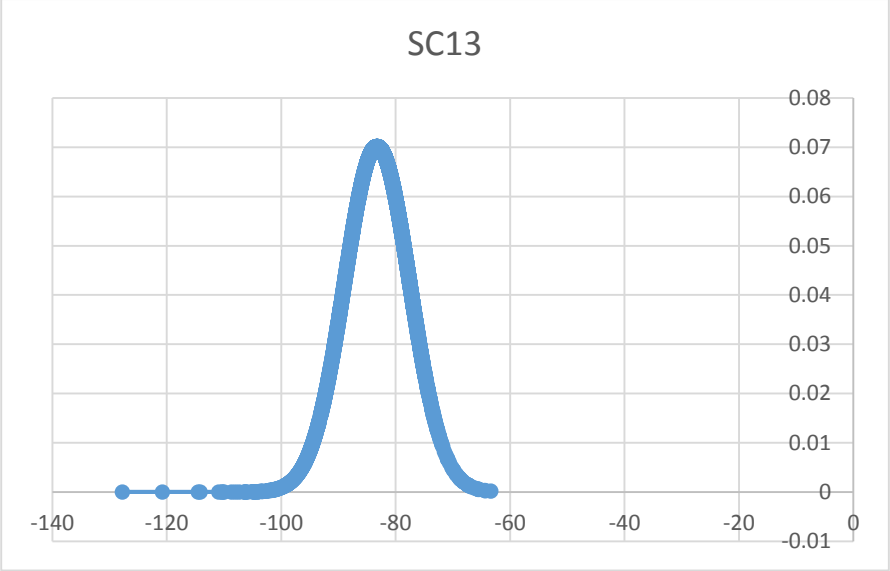


Figure 24Decibel value distribution graph for scenario 13 phase 01

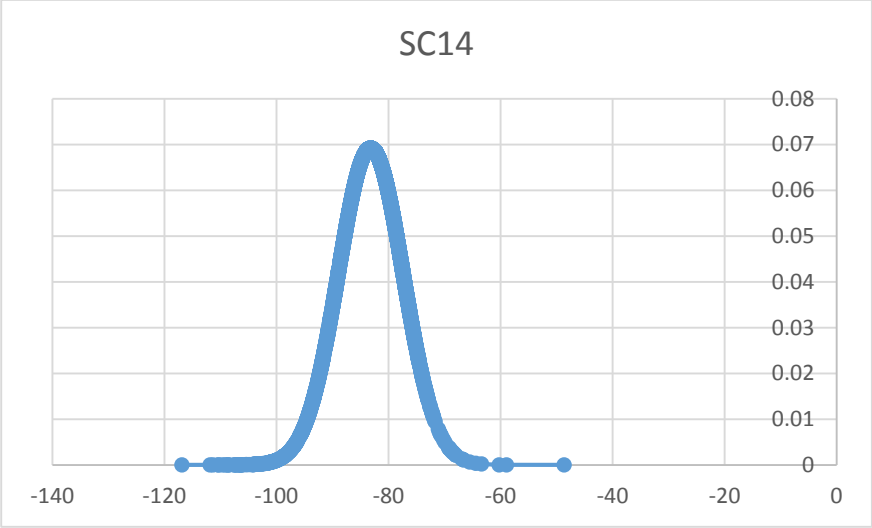


Figure 25Decibel value distribution graph for scenario 14 phase 01

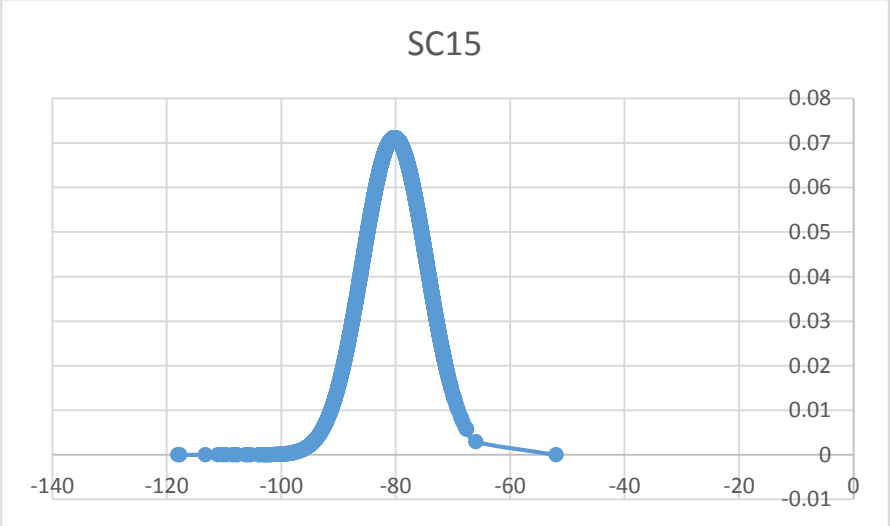


Figure 26Decibel value distribution graph for scenario 15 phase 01

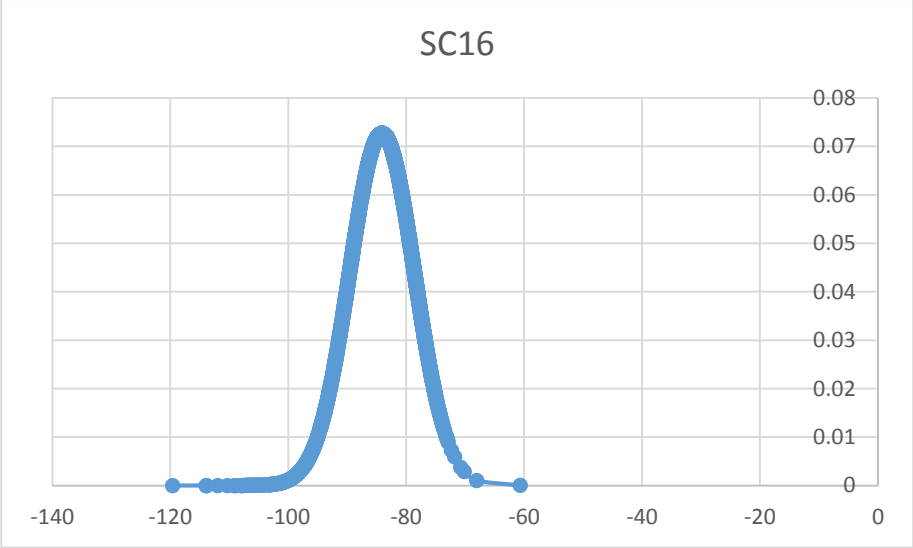


Figure 27Decibel value distribution graph for scenario 16 phase 01

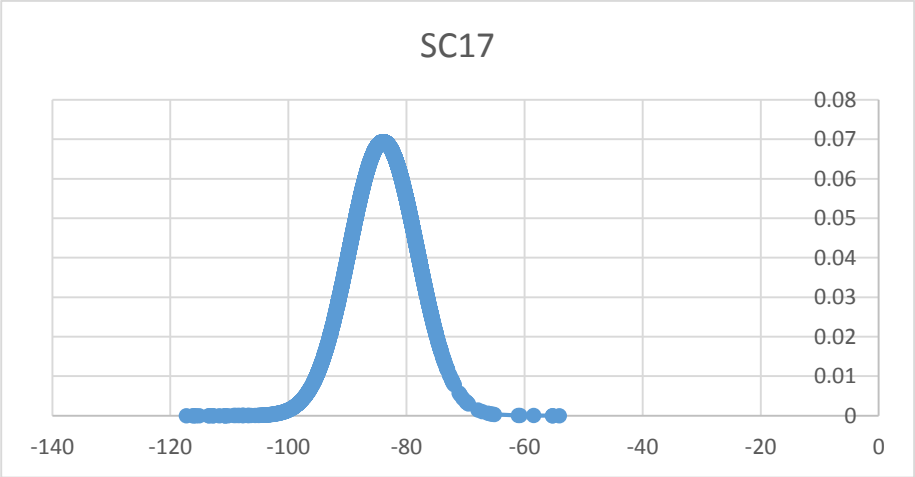


Figure 28Decibel value distribution graph for scenario 17 phase 01

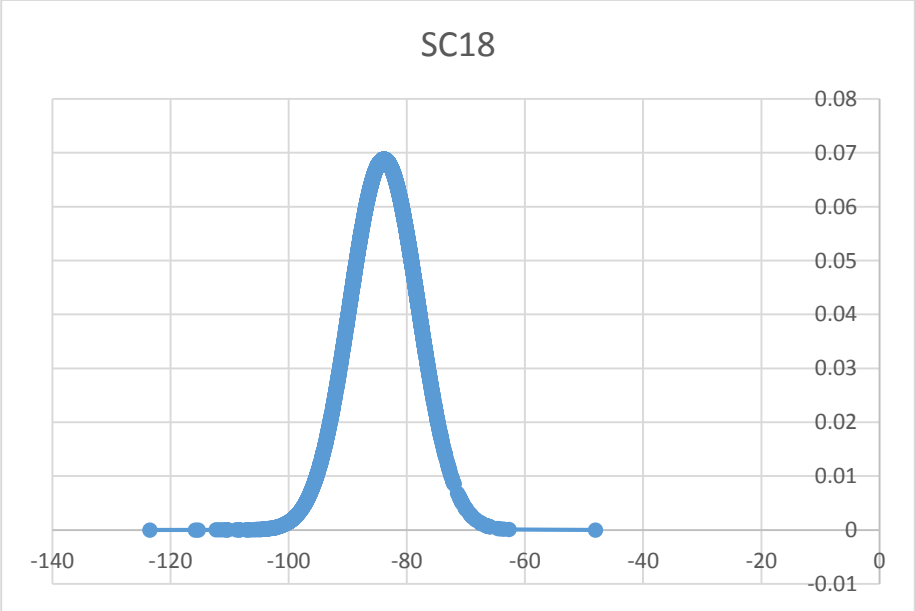


Figure 29Decibel value distribution graph for scenario 18 phase 01

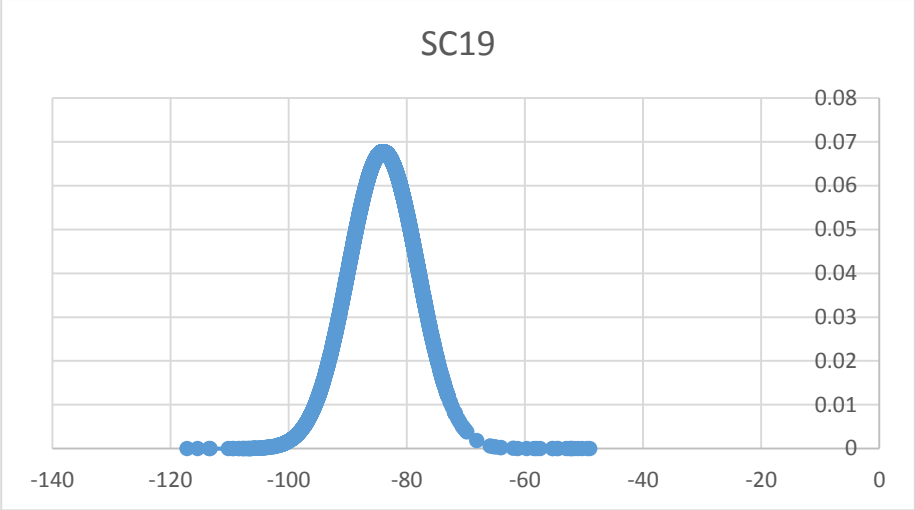


Figure 30Decibel value distribution graph for scenario 19 phase 01

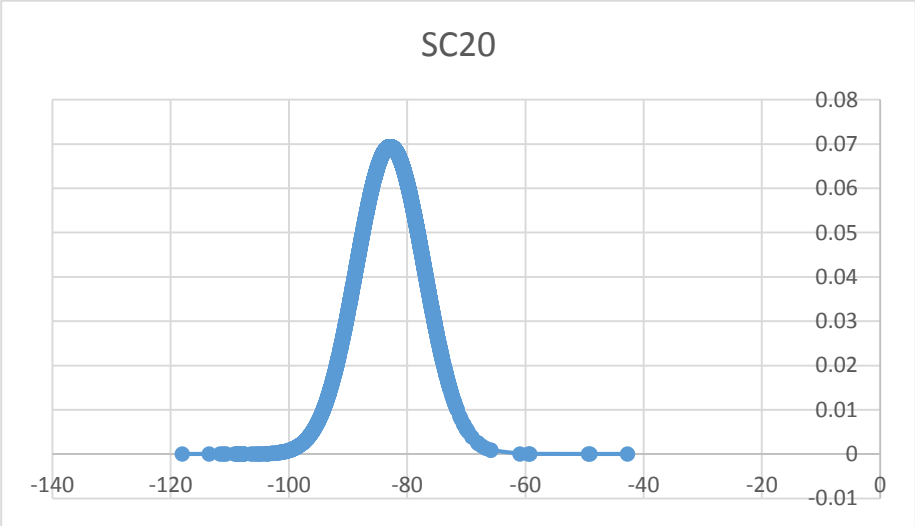


Figure 31Decibel value distribution graph for scenario 20 phase 01

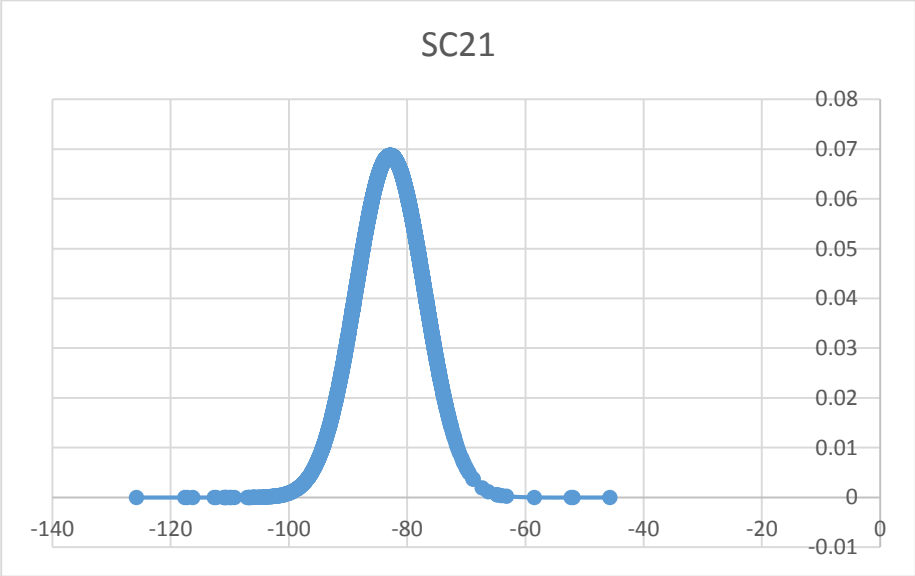


Figure 32Decibel value distribution graph for scenario 21 phase 01

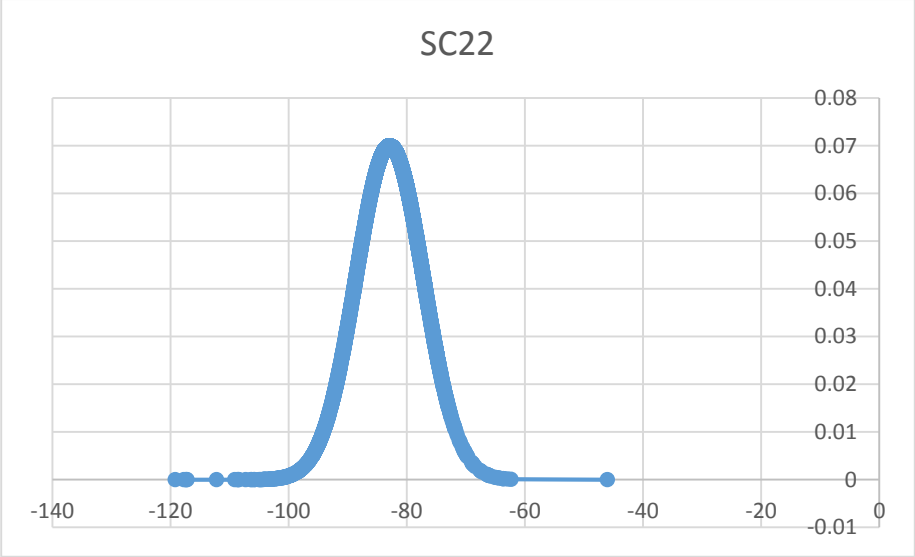


Figure 33Decibel value distribution graph for scenario 22 phase 01

At the same time average value of the Decibel value and standard deviation value is calculated. Below is the summarized table of average value and standard deviation value of the Decibel value with respect to each scenario.

Standard deviation is calculated using below formula.

$$s = \sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}$$

x = Decibel values

$\bar{x}$ = Mean value

n = Total number of values

Scenario	Average(mean)of Decibel Values	Standard Deviation Value
scenario 1	-83.47945031	5.650242323
scenario 2	-83.30253898	6.104447906
scenario 3	-83.35404991	5.655987723
scenario 4	-83.48741414	5.671347367
scenario 5	-83.38397882	6.611336494

scenario 6	-83.6355182	6.559258284
scenario 7	-84.0875789	5.737843093
scenario 8	-84.0901395	5.687938788
scenario 9	-82.13592054	7.857704636
scenario 10	-83.17407132	6.311691993
scenario 11	-83.64628702	5.736051261
scenario 12	-83.75574169	5.502426177
scenario 13	-83.21271962	5.686700222
scenario 14	-83.14263967	5.7699919
scenario 15	-80.17550898	5.61061478
scenario 16	-84.06840805	5.488852735
scenario 17	-83.91001825	5.75074265
scenario 18	-83.84692781	5.799711388
scenario 19	-83.95294161	5.889400442
scenario 20	-82.82778471	5.748594065
scenario 21	-82.82952737	5.805481144
scenario 22	-82.86227111	5.702038483

Table 4 Average (mean) of Decibel Values & Standard Deviation of Scenarios Phase 01

Evaluation of Phase 01

In the evaluation stage we are focusing on the answer the research questions using the analysed data.

You can see the Average of Decibel values and Standard deviation values are not significantly changed. Before you put those values into statistical hypothesis testing methodologies there should be a significant change of above Average of Decibel values and Standard deviation values.

- As I mentioned earlier these unexpected results led me to extend my research to phase 2.

4.2 Phase 02

To answer my research questions now I move into phase 02 with several changes comparing with the phase 01 as I mentioned under Design and implementation phase. There are four scenarios under this phase.

Distribution graphs for scenarios

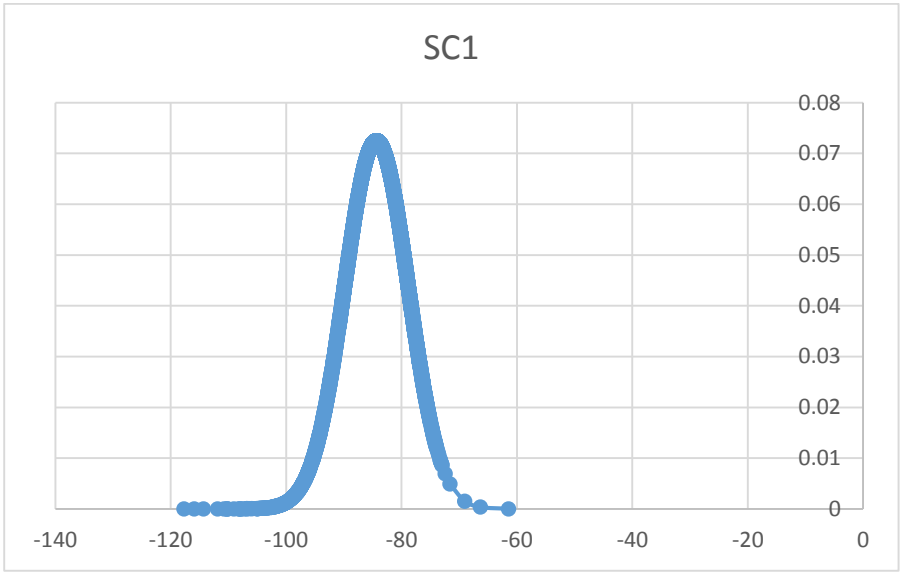


Figure 34 Decibel value distribution graph for scenario 01 phase 02

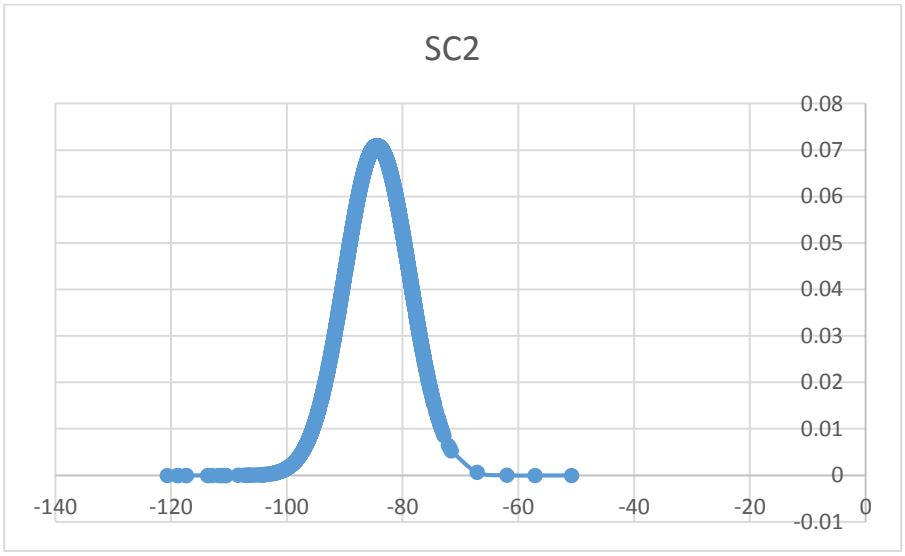


Figure 35 Decibel value distribution graph for scenario 02 phase 02

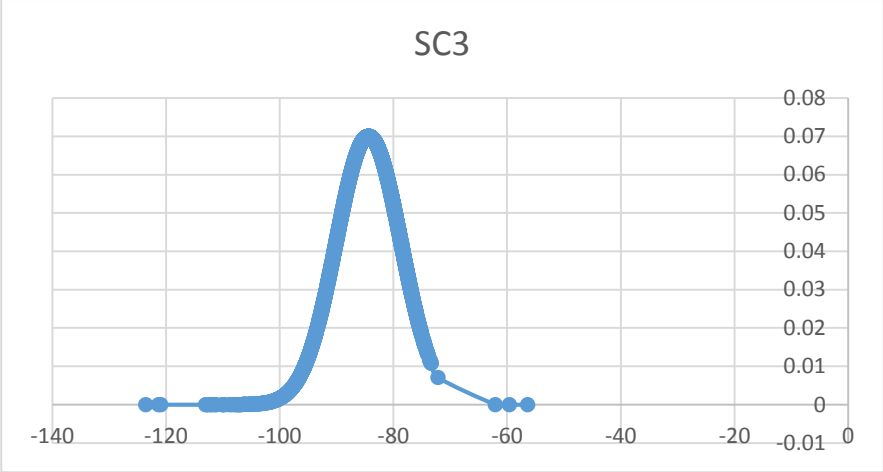


Figure 36 Decibel value distribution graph for scenario 03 phase 02

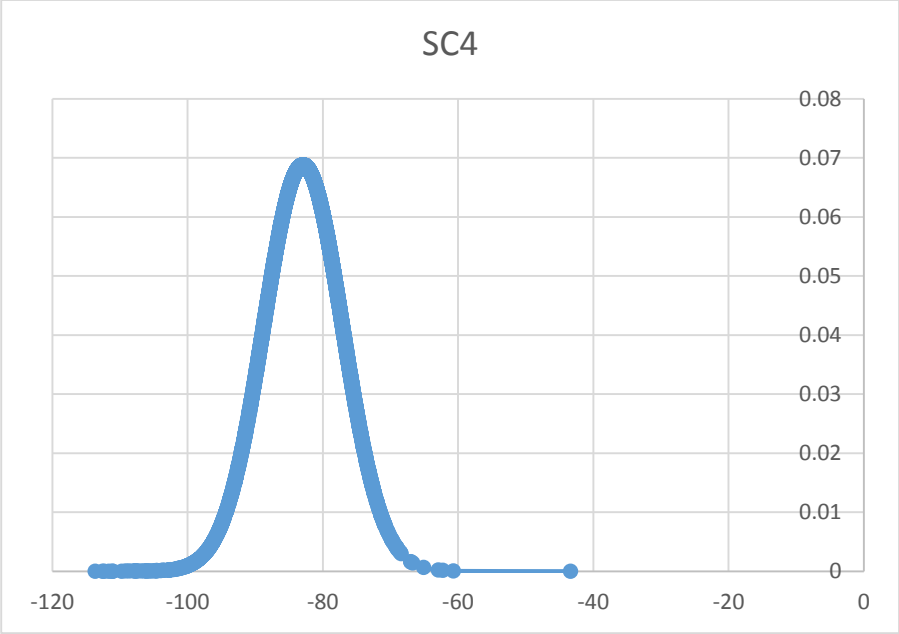


Figure 37 Decibel value distribution graph for scenario 04 phase 02

Standard Deviation and Average value has also done as previous calculations.

Scenario	Average(mean)of Decibel Values	Standard Deviation Value
scenario 1	-84.33101207	5.504711441
scenario 2	-84.37103717	5.626180508
scenario 3	-84.34505301	5.702064805
scenario 4	-82.90127629	5.801837580

Table 5 Average (mean) of Decibel Values & Standard Deviation of Scenarios Phase 02

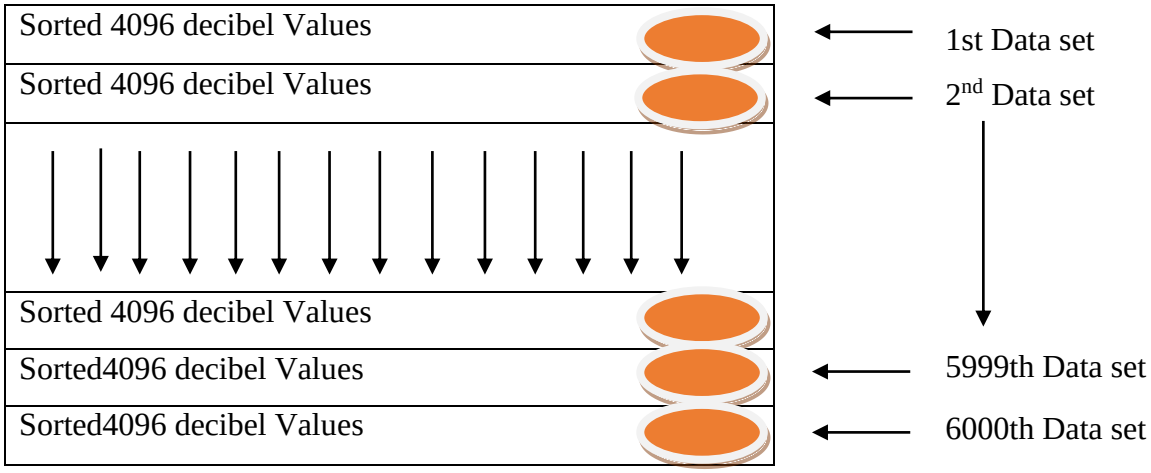
As I mentioned before since the expected results were not achieved even this phase I decided to make another few alterations to the methodology.

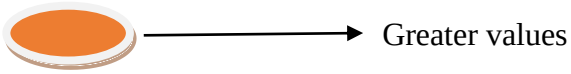
4.3 Phase 03

Another special drawback of HackRF One was found at the initial stage (Phase 01, and 02). When we tuned the device into an exact frequency the device will automatically shows a peak value into that frequency in all the Graphs and stored the values as it is.

Because of that another change is added to the phase 03 apart from the above mentioned changes.

That change is, sort all the values what we got and get the average value of grater four Decibel values among those values (Python script appendix 03).





Distribution graphs for scenarios

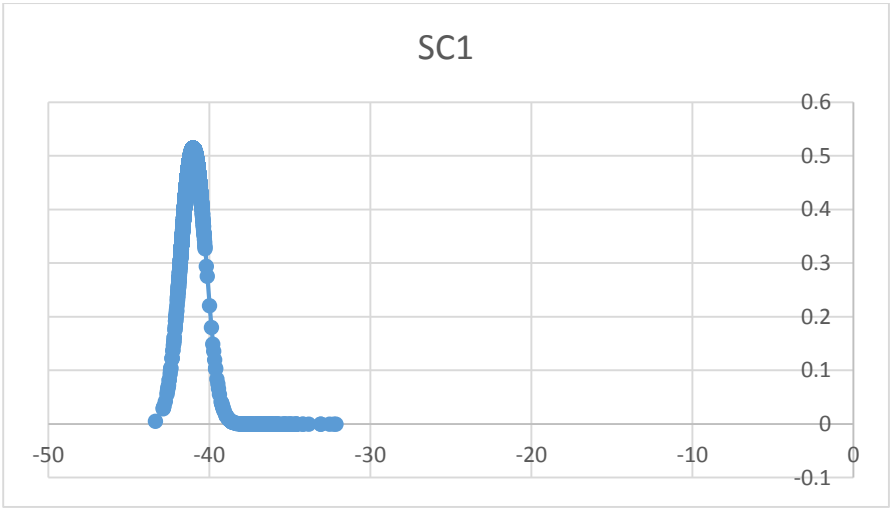


Figure 38 6Decibel value distribution graph for scenario 01 phase 03

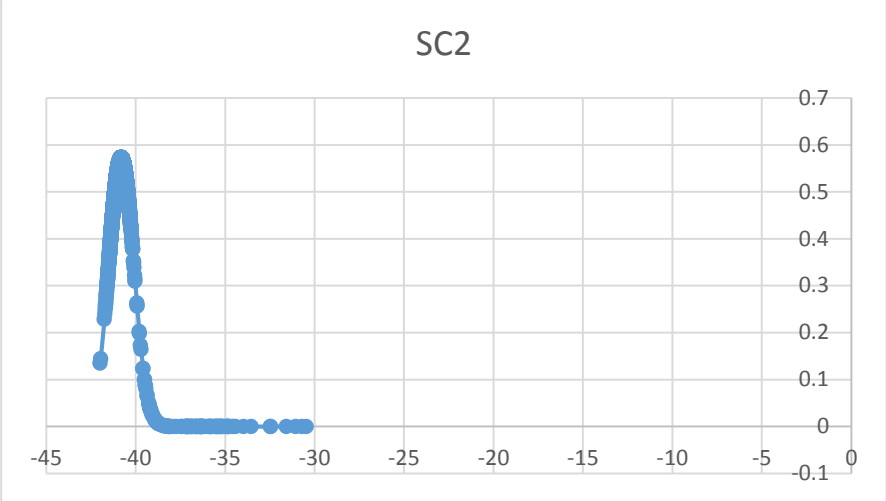


Figure 396Decibel value distribution graph for scenario 02 phase 03

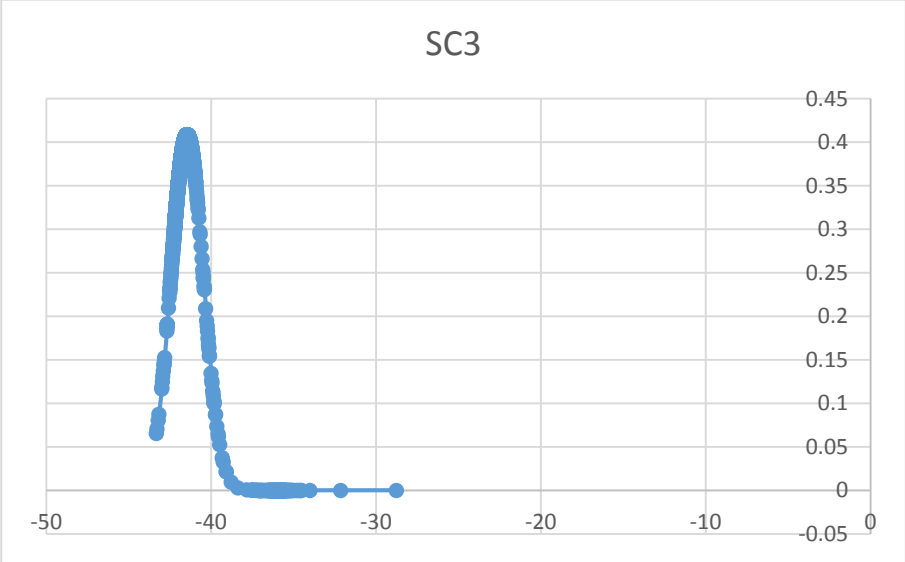


Figure 406Decibel value distribution graph for scenario 03 phase 03

Standard Deviation and Average value has also done as previous calculations.

Scenario	Average(mean)of Decibel Values	Standard Deviation Value
scenario 1	-40.9894713	0.775966505
scenario 2	-40.80270189	0.695988144
scenario 3	-41.4450919	0.976138079

Table 6 Average (mean) of Decibel Values & Standard Deviation of Scenarios Phase 03

- You can see though the average value is calculated, it was unable to attain the expected significant change.

- As I mentioned earlier these unexpected results led me to extend my research to phase 04.

4.4 Phase 04

This phase is basically done for the purpose of prove that,

- Is there any error with the HackRF One device?
- Is there any error with the wireless access point?

1. I removed the antenna of Hackrf One. I did this to clarify whether there is any fault in the device.

   You can see by below screen shots there is no any significant change, though I remove the antenna of the HackRF One.

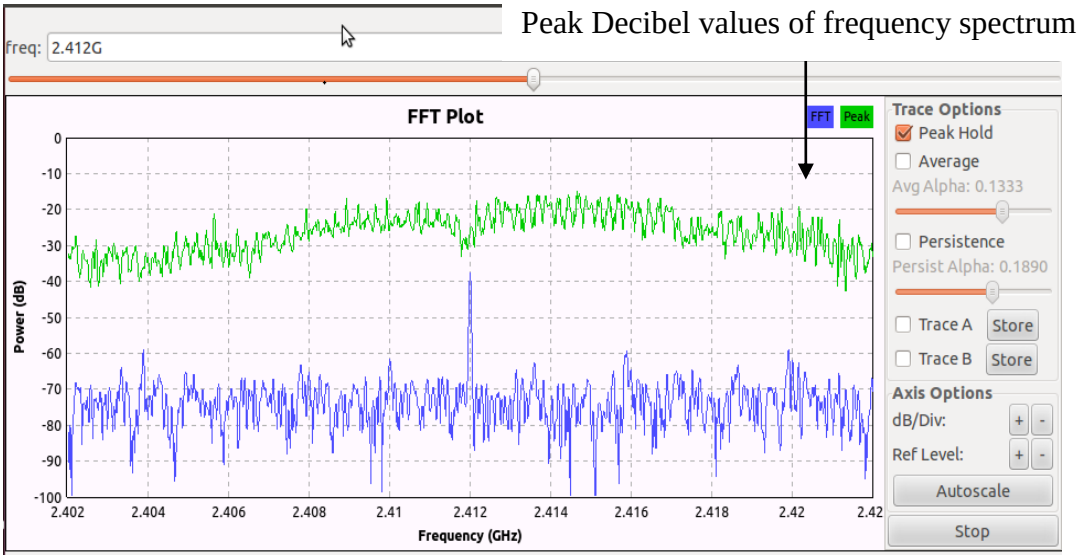


Figure 41 FFT plot Peak values distribution of frequencies, phase 04 – Without antenna of HackRF One

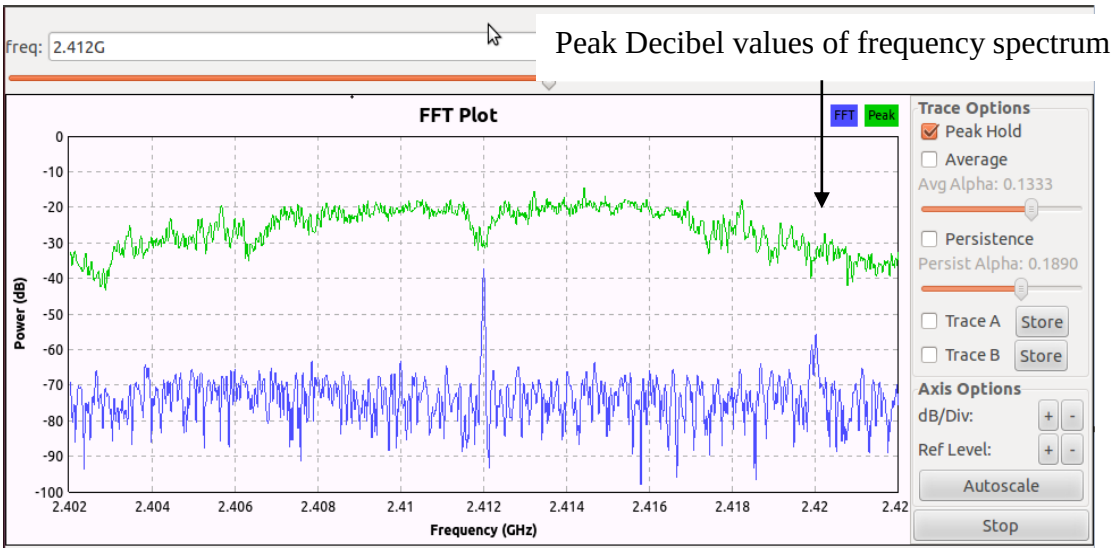


Figure 42 FFT plot Peak values distribution of frequencies, phase 04 – With antenna of HackRF One

2. I removed the router and checked the results.

At that moment HackRF one device is not giving an output, which means there is no any Wi-Fi signal to capture. Considering these output results, we can say there is no any error within the HackRF One device and also Wi-Fi router.

Chapter 5

5. Conclusion

This is the last chapter of this dissertation and also this part is often what a reader remembers best. This section provides a comprehensive overview of the research and its results and what type of decisions have taken by analysing those results. Also this contains possible future works that can be done using this research.

5.1 Contribution

Software-Defined Radio (SDR) is a research-oriented field in today's world. With the open source project GNURadio and the hardware platform HackRF One, very complex wireless transmission systems can be explored, even with a relatively small budget. Principally all of the needed modules and information can be found in the internet. This dissertation tries to gives you an Investigation of Privacy Violations in Wi-Fi Band Using Software Defined Radio.

At the start of the study there were two research questions. First one is. Is there a significant drop in the RSSI levels when there is a human present obstructing the wireless access point and the data gathering computer? The next questing is how accurately we can identify human presence by observing received signal strength indicator (RSSI) values? To come up with a fine solution I have stimulated my research through several designed scenarios under four designed phases. After implementing those scenarios I have analysed and evaluated that gathered data.

Those gathered data can't be categorized as single data elements. These data is a distribution of data values. Because of that I used lot of methods and calculations to get meaningful results before analysing and evaluation. After evaluating all the analysed data, I could find there is no any significant change among those results as I expected.

Because I didn't get the expected results I furthermore studied the reasons for this mislead and then came to find the actual reason. Wi-Fi uses Orthogonal Frequency Division Multiplexing method to transfer large amounts of data that uses over 52 separate, evenly spaced frequencies. OFDM splits the radio signal into these separate frequencies and simultaneously transmits them to the receiver. Splitting the signal and transferring over different frequencies reduces the amount of crosstalk interference.

Also when the packet transfers through the frequencies channels noise is increased beyond those frequencies. But this is not badly affected to the data packets, but it is badly affected to the outside data capturing devices (like HackRF One). Because of that all the data that I captured from HackRF One can be called as noise data.

To overcome this obstacle and to continue the research we need to measure the real data packets variations. Because of that we have to demodulate all the frequencies and get the actual data packets and then apply to the above calculations.

5.2 Challenges

- Lack of resources for SDR

  Software defined radio is a novel area on the field of research. Because of this I had to done extensive work just to learn the inner workings of SDR. At the beginning of the project the device (HackRF One) I used for this project was not available in the University or the Sri Lankan market. Due to this I had to import the device from USA. This took about three months and during that time I could only do the design and do the literature review on the field I'm about to do the research.
- Lack of previous research done using the HackRF One

  There were some researches done in the area of human tracking using SDR. But all the researches I found used USRP device. This is a very expensive and advanced hardware device. Also there is very support available in the part of integration of HackRF One with GNURadio. All the available flow graphs in GRC are for USRP because of that I had to design and self implement flow graphs that are compatible with HackRF One.
- Need of high performance computing

  In the data gathering phase HackRF One device captures about 20 million data samples per second. When tried to capture data using a normal laptop it fails because that laptop does not have required throughput. I was forced to use a high end server computer because of this.
- Mobility issues

  According to the above mentioned facts I used high end server for my data gathering phases. This server weighs about 45Kg because of that it is a challenge to move that server to several locations. This high end server also required power resources and in

some places there were difficulties providing required power. If it was a laptop I would not face this difficulty.

- Storage and processing of gathered data

  As mentioned earlier for one scenario gathered data size is around 150 GB and I have tested around 30 scenarios. This is around 4.5TB. To process and store such a large amount of data server is required. Again I had to use the server I used for data gathering because of this data gathering and processing could not be done simultaneously.

5.3 Future Work

- Demodulate the signal

  As mentioned earlier Wi-Fi uses OFDM method to transfer large amounts of data that uses over 52 separate, evenly spaced frequencies. When these data is captured using HackRF One due background frequency noise interference that the data carrying signals strength cannot be identified. To decode the original signal it is yet to be find a methodology using HackRF One and SDR.

- Privacy violations through SDR's

  There are localization projects done using SDR's. Still Research's does not initiated any projects concerning the privacy violation that are possible using SDR's.

- Implement projects by HackRF One

  Many human tracking and gesture tracking projects are achieved using the expensive and advanced device USRP. There are very low number of projects carried out in this area using the HackRF One. One can start implementing projects using HackRF One that are previously implemented by using USRP. Then the benchmarking and comparisons can be carried out to review effectiveness of HackRF One.

- Usage of multiple HackRF One devices.

  Theoretically accuracy of tracking increases when we gather data from multiple locations. By applying this project can be started to find out the effects on the accuracy of the results when multiple HackRF One devices are being used.

REFERENCES

[1] Chunmei Han, Kaishun Wu, Yuxi Wang, and Lionel M Ni. Wifall: Device-free fall detection by wireless networks. In INFOCOM, 2014 Proceedings IEEE, pages 271–279. IEEE, 2014.

[2] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In Proceedings of the 20th annual international conference on Mobile computing and networking, pages 617–628. ACM, 2014.

[3] Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, and Yunhao Liu. Towards omnidirectional passive human detection. In INFOCOM, 2013 Proceedings IEEE, pages 3057–3065. IEEE, 2013.

[4] Wei Xi, Jizhong Zhao, Xiang-Yang Li, Kun Zhao, Shaojie Tang, Xue Liu, and Zhiping Jiang. Electronic frog eye: Counting crowd using wifi. In INFOCOM, 2014 Proceedings IEEE, pages 361–369, April 2014.

[5] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M Ni. We can hear you with wi-fi! In Proceedings of the 20th annual international conference on Mobile computing and networking, pages 593–604. ACM, 2014.

[6] Stephan Sigg, Shuyu Shi, Felix Buesching, Yusheng Ji, and Lars Wolf. Leveraging rf-channel fluctuation for activity recognition: Active and passive systems, continuous and rssi-based signal features. In Proceedings of International Conference on Advances in Mobile Computing & Multimedia, page 43. ACM, 2013.

[7] Stephan Sigg, Markus Scholz, Shuyu Shi, Yusheng Ji, and Michael Beigl. Rf-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals. Mobile Computing, IEEE Transactions on, 13(4):907–920, 2014.

[8] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. Wi-fi gesture recognition on existing devices. arXiv preprint arXiv:1411.5394, 2014.

[9] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In Proceeding of the 11th annual international conference on Mobile systems, applications, and services, pages 249–262. ACM, 2013.

[10] Jiang Xiao, Kaishun Wu, Youwen Yi, and Lionel M Ni. Fifs: Fine-grained indoor fingerprinting system. In Computer Communications and Networks (ICCCN), 2012 21st International Conference on, pages 1–7.
IEEE, 2012.

[11] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssito csi: Indoor localization via channel response. ACM Computing Surveys (CSUR), 46(2):25, 2013.

[12] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, andShwetak Patel. Whole-home gesture recognition using wireless signals. In Proceedings of the 19th annualinternational conference on Mobile computing & networking, pages 27–38. ACM, 2013.

[13] Bryce Kellogg, Vamsi Talla, and ShyamnathGollakota. Bringing gesture recognition to all devices. In Usenix NSDI, volume 14, 2014.

[14] Bastien Lyonnet, Cornel Ioana, and Moeness G Amin.Human gait classification using microdoppler time-frequency signal representations. In RadarConference, 2010 IEEE, pages 915–919. IEEE, 2010.

[15] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert CMiller. 3d tracking via body radio reflections. In Usenix NSDI, volume 14, 2013.

[16] Pu, Q., Gupta, S., Patel, S. and Gollakota, S. (2013). Whole-Home Gesture Recognition Using Wireless Signals. In: *The 19th Annual International Conference on Mobile Computing and Networking (Mobicom'13)*. [online] Available at: http://wisee.cs.washington.edu/wisee_paper.pdf

[17].K. Woyach, D. Puccinelli, and M. Haenggi. Sensorless sensing in wireless networks: implementation and measurements. In Proceedings of the Second International Workshop on Wireless Network Measurement (WiNMee), 2006.

[18].Dian Zhang and L.M. Ni. Dynamic clustering for tracking multiple transceiver-free objects. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2009), march 2009.

[19].Moustafa Youssef, Matthew Mah, and Ashok Agrawala. Challenges: device-free passive localization for wireless environments. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking, MobiCom '07, pages 222–229, New York, NY, USA, 2007. ACM.

[20]. Joey Wilson and Neal Patwari. Through-wall tracking using variance-based radio tomography networks. CoRR, abs/0909.5417, 2009.

[21]. Joey Wilson and Neal Patwari. Radio tomographic imaging with wireless networks. IEEE Transactions on Mobile Computing, 9:621–632, 2010.

[22]. N. Patwari and J. Wilson. Spatial models for human motion-induced signal strength variance on static links. Information Forensics and Security, IEEE Transactions on, 6(3):791 –802, sept. 2011.

[23]. Markus Reschke, Sebastian Schwarzl, Johannes Starosta, Stephan Sigg, andMichael Beigl. Context awareness through the rf-channel. In Proceedings of the2nd workshop on Context-Systems Design, Evaluation and Optimisation, 2011.

[24]. Markus Reschke, Johannes Starosta, Sebastian Schwarzl, and Stephan Sigg. Situation awareness based on channel measurements. In Proceedings of the fourthConference on Context Awareness for Proactive Systems (CAPS), 2011.

Appendices

Appendices 1

GRC Generated python script (top_block.py)

```
#!/usr/bin/env python2
##################################################
# GNU Radio Python Flow Graph
# Title: Top Block
# Generated: Fri Nov 13 14:47:26 2015
##################################################

if __name__ == '__main__':
    import ctypes
    import sys
    if sys.platform.startswith('linux'):
        try:
            x11 = ctypes.cdll.LoadLibrary('libX11.so')
            x11.XInitThreads()
        except:
            print "Warning: failed to XInitThreads()"

from gnuradio import blocks
from gnuradio import eng_notation
from gnuradio import gr
from gnuradio.eng_option import eng_option
from gnuradio.fft import logpwrfft
from gnuradio.filter import firdes
from grc_gnuradio import wxgui as grc_wxgui
from optparse import OptionParser
import wx

class top_block(grc_wxgui.top_block_gui):
```

```
    def __init__(self):
        grc_wxgui.top_block_gui.__init__(self, title="Top Block")
        _icon_path = "/usr/share/icons/hicolor/32x32/apps/gnuradio-grc.png"
        self.SetIcon(wx.Icon(_icon_path, wx.BITMAP_TYPE_ANY))

        ##################################################
        # Variables
        ##################################################
        self.samp_rate = samp_rate = 20e6
        self.fft_size = fft_size = 2000000

        ##################################################
        # Message queues (added by grcconvert)
        ##################################################
        self.msgq_out = blocks_message_sink_0_msgq_out = gr.msg_queue(2)

        ##################################################
        # Blocks
        ##################################################
        self.logpwrfft_x_0 = logpwrfft.logpwrfft_c(
            sample_rate=samp_rate,
            fft_size=fft_size,
            ref_scale=2,
            frame_rate=30,
            avg_alpha=1.0,
            average=False,
        )
        self.blocks_throttle_0 = blocks.throttle(gr.sizeof_gr_complex*1, samp_rate,True)
        self.blocks_message_sink_0 = blocks.message_sink(gr.sizeof_float*fft_size,
blocks_message_sink_0_msgq_out, False)
        self.blocks_file_source_0 = blocks.file_source(gr.sizeof_gr_complex*1,
"/home/jam/data/Nse03", False)

        ##################################################
```

```
        # Connections
        ##################################################
        self.connect((self.blocks_file_source_0, 0), (self.blocks_throttle_0, 0))
        # removed by grcconvert: # self.connect((self.blocks_message_sink_0, 'msg'), (self, 0))
        self.connect((self.blocks_throttle_0, 0), (self.logpwrfft_x_0, 0))
        self.connect((self.logpwrfft_x_0, 0), (self.blocks_message_sink_0, 0))


    def get_samp_rate(self):
        return self.samp_rate

    def set_samp_rate(self, samp_rate):
        self.samp_rate = samp_rate
        self.blocks_throttle_0.set_sample_rate(self.samp_rate)
        self.logpwrfft_x_0.set_sample_rate(self.samp_rate)

    def get_fft_size(self):
        return self.fft_size

    def set_fft_size(self, fft_size):
        self.fft_size = fft_size


if __name__ == '__main__':
    parser = OptionParser(option_class=eng_option, usage="%prog: [options]")
    (options, args) = parser.parse_args()
    tb = top_block()
    tb.Start(True)
    tb.Wait()
```

Appendices 2

```
import grcconvert
import struct
grcconvert.main("top_block.py")
import top_block
import matplotlib.pyplot as plot
import os
from collections import deque

def do_something(with_this):

        #Madupa: trying to play with this
        #print "with_this:"
        os.system('clear')
        print "------------------------------"
        print "length       =", len(with_this)
        print "start point  =", with_this[0][0], "dB"
        print "center point =", with_this[len(with_this)/2][0], "dB"
        print "end point    =", with_this[len(with_this)-1][0], "dB"
        print "------------------------------"

        plot.clf()
        # it seems logpwrfft exchanges the first and second part of the FFT output, we correct
it:
        plot.plot(with_this[len(with_this)/2:]+with_this[:len(with_this)/2])
        plot.draw()


def plot_power(data_array):

        os.system('clear')
        print '\n\n'
```

```
        print "signal strengh at the center of OFDM symbol:", data_array[len(data_array)/2][0],"dB"

        #Edited by Madupa....
        f = open('outputFile.txt', 'a+')

        if data_array[len(data_array)/2][0] <= -60:
                a1.appendleft(-80)
        else:
                a1.appendleft(data_array[len(data_array)/2][0])
        datatoplot = a1.pop()
        if datatoplot != 0:
                dt = str(datatoplot)+"\n"
                f.write(dt)
        f.close()

        line.set_ydata(a1)
        plot.draw()
        plot.pause(0.00001)



plot.ion()
tb=top_block.top_block()
tb.start()

#Madupa: experiment
a1 = deque([0]*100)
ax = plot.axes(xlim=(0, 100), ylim=(-100, 0))
ax.grid(True)
line, = plot.plot(a1)
plot.ylim([-150,0])
plot.show()

while True:
```

```
fft=tb.msgq_out.delete_head().to_string() # this indeed blocks
floats=[]
for i in range(0,len(fft),4):
        floats.append(struct.unpack_from('f',fft[i:i+4]))
print "got",len(floats), "floats; FFT size is", tb.fft_size
i=0
while i<len(floats): # gnuradio might sometimes send multiple vectors at once
        pack=floats[i:i+tb.fft_size-1]
        #do_something(pack)
        plot_power(pack)
        i+=tb.fft_size
```

Appendices 3

```
import grcconvert
import struct
grcconvert.main("top_block.py")
import top_block
import matplotlib.pyplot as plot
import os
from collections import deque

def do_something(with_this):

        #Madupa: trying to play with this
        #print "with_this:"
        os.system('clear')
        print "------------------------------"
        print "length       =", len(with_this)
        print "start point  =", with_this[0][0], "dB"
        print "center point =", with_this[len(with_this)/2][0], "dB"
        print "end point    =", with_this[len(with_this)-1][0], "dB"
        print "------------------------------"

        plot.clf()
        # it seems logpwrfft exchanges the first and second part of the FFT output, we correct
it:
        plot.plot(with_this[len(with_this)/2:]+with_this[:len(with_this)/2])
        plot.draw()


def plot_power(data_array):
        x=0
        value2 = 0
        allData = []
```

```
        #os.system('clear')
        #print '\n\n'
        #print "signal strengh at the center of OFDM symbol:", data_array[len(data_array)/2][0],"dB"
        while(x<2048):
                value = data_array[x][0]
                allData.append(value)
                #dataMetrix[index][x] = value

                #fle = open('outputData/file'+str(x)+'.txt', 'a+')
                #dtval = str(value)+"\n"
                #fle.write(dtval)
                #fle.close()

                x=x+1
                #print value
                #print (len(data_array))
                #print (data_array[x][0])
        #allData.sort()
        #print "avarage = ", value2/4094
        #print(allData[len(allData)-1])
        #print('test :',allData[2047])
        #print(allData)
        #print("\n\n\n\n\n\n\n\n\n")
        #Edited by Madupa....
        #allData
        #quit()
        f = open('outputFile.txt', 'a+')

#       if data_array[len(data_array)/2][0] <= -60:
#               a1.appendleft(-80)
#       else:
```

```
        avrg = 0
        for x in range(1, 5):
                avrg = avrg + (allData[len(allData)-x])

        avrg = avrg/4
        #print "Average Val: ",avrg
        a1.appendleft(avrg)

        #a1.appendleft(data_array[len(data_array)/2][0])

        datatoplot = a1.pop()
        if datatoplot != 0:
                dt = str(datatoplot)+"\n"
                f.write(dt)
        f.close()

        line.set_ydata(a1)
        plot.draw()
        plot.pause(0.00001)



plot.ion()
tb=top_block.top_block()
tb.start()

#experiment
a1 = deque([0]*100)
ax = plot.axes(xlim=(0, 100), ylim=(-100, 0))
ax.grid(True)
line, = plot.plot(a1)
plot.ylim([-150,0])
plot.show()
indx = 0
```

```
#Create multi-dimentional array to keep data( 2048 cols * 6000 rows)
#dataMetrix = [[0 for j in range(2048)] for j in range(6000)]

while True:
        fft=tb.msgq_out.delete_head().to_string() # this indeed blocks
        floats=[]

        for i in range(0,len(fft),4):
                floats.append(struct.unpack_from('f',fft[i:i+4]))
        #print "got",len(floats), "floats; FFT size is", tb.fft_size
        i=0
        while i< len(floats): # gnuradio might sometimes send multiple vectors at once
                pack=floats[i:i+tb.fft_size-1]
                #do_something(pack)
                plot_power(pack)
                i+=tb.fft_size
        #print dataMetrix[indx][2047]
        #indx = indx + 1
```