University of Southern Queensland
Faculty of Engineering and Surveying

Investigation of Next Generation IEEE 802.11n Wireless Local Area Networks (WLAN)

A dissertation submitted by

Zee Hong CHAN

In fulfillment of the requirements of

Courses ENG4111 and 4112 Research Project

Towards the degree of

Bachelor of Engineering (Electrical and Electronic)

Submitted: October, 2009

Abstract

The aim of this project is to investigate the new IEEE 802.11n WLAN standard. The IEEE 802.11n is a new amendment to the already popular IEEE 802.11 standard, this new amendment aims to create a seamless working environment by the mobility of wireless with the performance of the wired networks. It provides a higher throughput, increased coverage, and improved reliability of the over-the-air WLAN transmission.

The main techniques used to achieve these improvements, based on the physical layer, are the use of multiple transmit and receive antennas, referred to as MIMO, and multiple subcarrier transmission scheme, or OFDM. With the improved PHY, together with the use of wider bandwidth, known as channel bonding, and increase transmission efficiency by aggregating multiple packets, the 802.11n can achieve a potential throughput beyond 200 Mbps.

Methods of channel modelling are discussed, with an investigation to the OFDM and MIMO to analyse the improvement of this new amendment in the PHY transmission. Simulation results that illustrate the advantages of the new techniques are presented, as well as the comparison of various transmission schemes, for example antenna numbers, modulations type, and etc.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 & ENG4112 *Research Project*

Limitations of Use

The council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

The dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof Frank Bullen
Dean
Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Zee Hong CHAN

Student Number: 0050074413

Signature

Date

Acknowledgements

I would like to acknowledge the efforts and encouragement given by my project supervisor, Wei Xiang. I would like to thanks him for his continuing support and also his supervision that always lead me to the proper direction throughout this project. I really appreciate his guidance during the process of research and also in the writing of this dissertation.

I would also like to thanks my family and friends for giving me their precious mental support, especially my housemates Wei and Kim who given me a lot of support and idea during the write up of the dissertation.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

Wireless Local Area Network (WLAN) has been so important in our daily life nowadays, thanks to the aggressive development of portable digital gadgets, as well as the freedom to be unwired, WLANs have evolved from an interesting idea to an essential and important technology that many individuals and businesses cannot live without.

But why do we need IEEE 802.11n? One of the purpose of having a computer network is about information sharing, and since nowadays we have those high quality digital contents which generally have a relatively larger file size, effective sharing could be limited if we are connected to a slower connection. Also, due to the success of the IEEE 802.11g which then drove the use of Wi-Fi to a new height and then expanded the demands for the WLAN technology.

And as a consumer, we always ask for more, we need more speed, also we need more coverage. Under these circumstances, it then built up the foundation for the next generation WLAN, and therefore the development of the IEEE 802.11n amendment. The IEEE 802.11n is the latest amendment to the original IEEE 802.11 standard, due to the continuing development of this WLAN technology since the early 2003, the IEEE 802.11 Working Group started defining the improved Physical Layer (PHY) and Media Access Control (MAC) features as part of the IEEE 802.11n standard.

One of the mandatory improvements in the PHY layer is the use of multiple transmit and receive antennas technology, or more commonly known as Multiple-Input Multiple-Output (MIMO), this PHY improvement built on top of the Orthogonal Frequency Division Multiplexing (OFDM), which is another key feature used in this standard that was already popularly known in IEEE 802.11a/g standard. MIMO actually take advantage of the multipath fading, use it constructively by applying some complex mathematical calculations, which will be discussed later, to increase the throughput dramatically compare to the conventional single antenna system and also to improve the range of reception.

1.2 Project aim

The aim of this project is to do a research for the WLAN, or more importantly, investigating on the new IEEE 802.11n WLAN standard, identifying some of the key features that this new standard implemented, how and why it can outperform the existing 802.11a/b/g standard which is widely being used now all around the world. Then a simulation is being done to the PHY layer to identify the improvement achieved by this new amendments.

1.3 Research Objectives

To accomplish the above aims, objectives have been set to make sure the research process is advance in a systematic manner. The main objectives of the project are outlined below:

- Research the background information relating to the IEEE 802.11 standard, and study the evolution between different variants of amendment for the standard.
- Investigate the advanced technology implemented into the new IEEE 802.11n standard that allows it to achieve a better performance.
- Further investigate into the PHY implementation, understanding that the PHY improvement, including the OFDM and MIMO, are the major advancement in this new amendment.
- Analyse and implement a software platform to simulate the wireless transmission.
- Report and discuss the results from the simulation, then comparing and identifying the improvement with the older standards.

1.4 Dissertation Overview

This dissertation can basically be divided into six main chapters, below are some of the overview of each chapter.

Chapter 1

An introduction to the dissertation, which also give some general idea to the topic. Project aim and the research objectives to be achieved are also presented.

Chapter 2

Literature Reivew on the background and some history of the development of the 802.11, a brief introduction to the 802.11n amendment is also presented.

Chapter 3

Detailed theory and formula is presented in this chapter, an insight study on the specification of the 802.11n amendments on how it can achieve the improved performance, for example, the increased throughput, larger coverage area, as well as the enhanced stability.

Chapter 4

A simulation to model the PHY layer of 802.11 radio transmission process is discussed.

Chapter 5

This chapter shows the result from the simulation, together with the analysis and discussion of the results.

Chapter 6

This chapter concludes the whole project work.

1.5 Chapter Summary

Some of the fundamental idea of the project is presented here, this include the introduction to the topic, also the project aim is set, together with some objectives to be achieve throughout the end of the execution of the project.

Chapter 2 Literature Review

2.1 History of IEEE 802.11

The very first version of the 802.11 standard was rectified in the year of 1997, mainly driven by the successful of the use of Ethernet, or the IEEE 802.3 standard. Also, because of the simplicity of this wireless protocol, and its similarity to the Ethernet, implementation of this new wireless technology to replace the old Ethernet network is very easy and consistent to every station, therefore increasing the adoption of this wireless technology to the industry, and slowly it also expanded to the general consumer.

As one of the member of the Local Area Network (LAN), or the IEEE 802 family, the IEEE 802.11 wireless communications need to interface with 802.1 architecture, management, and interworking, and 802.2 Logical Link Control (LLC). Initially the 802.11 wireless task groups defined the specifications with the combination use of 802.11 MAC and PHY layer that communicates up to the 802.2 LLC. Figure 1 shows the OSI model of the 802.11, with MAC, PHY, and LLC as the main architecture components in the 802.11 that make up the physical and data link layers of the Open Systems Interconnection (OSI) reference model for the WLAN communication.

The original IEEE 802.11 standard provided three initial specifications for the PHY. First, a less important infra-red (IR) PHY, operating at baseband was described, then another two radio-based PHYs running on 2.4 GHz carrier were also described. Initially, those described PHYs were all designed to support only 1 Mbps to 2 Mbps of transfer rates.

Both the radio-based PHY were specified for operating in 2.4 GHz of frequency channel, which is part of the Industrial, Scientific, and Medical (ISM) band. This is an unlicensed frequency channel, and is originally reserved internationally for the use of RF electromagnetic fields for Industrial, Scientific, and Medical purposes. Due to its unlicensed nature, this 2.4 GHz frequency channel has attracted different kind of development of technology to be built under this band. But, any communication devices that would like to operate in this ISM band must tolerate to any interference emitted from any ISM equipments. Some common electrical and electronic appliances that work under this frequency band like, microwave oven, cordless phone, etc are all sort of source of interference to our communication devices.

<table>
<tr><td>7. Application</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6. Presentation</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>5. Session</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>4. Transport</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3. Network</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">2. Data Link</td><td>LLC</td><td colspan="4">802.2 Logical Link Control</td></tr>
<tr><td>MAC</td><td rowspan="2">802.3
Ethernet</td><td rowspan="2">802.4
Token Bus</td><td rowspan="2">802.5
Token Ring</td><td rowspan="2">802.11
WLAN</td></tr>
<tr><td>1. Physical</td><td>PHY</td></tr>
</table>

Figure 1 OSI reference model

2.2 Evolution of IEEE 802.11 – the IEEE 802.11a/b/g

The original PHY layer of the 802.11 (1997) for the use in 2.4 GHz, are Frequency Hopped Spread Spectrum, and Direct Sequence Spread Spectrum (DSSS), but these two PHYs can only support a slow transfer rate up to 2 Mbps (Mega bits per second) as mentioned before.

In the year of 1999, two new amendments were defined to the original IEEE 802.11 standard. One of it is the IEEE 802.11b, this standard as usual work in the 2.4 GHz spectrum, also it uses the DSSS PHY but with an enhanced modulation scheme known as Complementary Code Keying (CCK), providing a theoretical transfer rate up to 11 Mbps. With a 5 times boost of higher data rates (2 Mbps to 11 Mbps), the IEEE 802.11b development achieved a significant market success in the WLAN industry.

Another amendment made during the same year in 1999 was the IEEE 802.11a, the main different between this standard compare to the IEEE 802.11b/g is that it operate in a special 5 GHz spectrum, it has both pros and cons with this, the advantage is that this is a cleaner spectrum if compare to the noisy 2.4 GHz ISM band, but the disadvantage is that the operation in 5 GHz could be quite limited, because under rules and regulation by different country, especially in Europe, the use of 5 GHz is generally restricted, resulting a slow adoption worldwide. Other than the operation in different frequency channel, the IEEE 802.11a created a new PHY using a transmission scheme known as Orthogonal Frequency Division Multiplexing (OFDM), which is an important breakthrough feature that increases the theoretical data rate up to 54 Mbps.

	802.11	802.11a	802.11b	802.11g	802.11n
PHY technology	DSSS	OFDM	DSSS w/CCK	OFDM & DSSS w/CCK	MIMO & OFDM
Throughput	1 or 2 Mbps	6 – 54 Mbps	5.5 or 11 Mbps	1 – 54 Mbps	6 – 600 Mbps
Frequency channel	2.4 GHz	5 GHz	2.4 GHz	2.4 GHz	2.4 GHz & 5 GHz
Channel spacing	25 MHz	20 MHz	25 MHz	25 MHz	20 GHz & 40 GHz

Table 1 PHY comparisons of 802.11 and its amendments

Because of the new data rates of up to 54 Mbps that can achieve by using OFDM, and the wireless communication in the friendly 2.4 GHz ISM band are more welcomed, the 802.11 working group started on developing the 802.11g amendment once the Federal Communications Commission (FCC) permitted the use of OFDM in the 2.4 GHz frequency spectrum in the year of 2001. In addition, the operation in 2.4 GHz ensures that backward compatibility and interoperability can be maintained between 802.11g and older 802.11b devices. The operation in legacy mode allows communication between 802.11b station and 802.11g clients, and vice versa. The advantages of 802.11g mentioned above, with another 5 times gain in data rate compare to 802.11b has made the 802.11g a major market success in the WLAN industry nowadays, we can easily find any modern portable devices that have this technology installed in it. A summary of PHY features implemented in different 802.11 amendments is given table above.

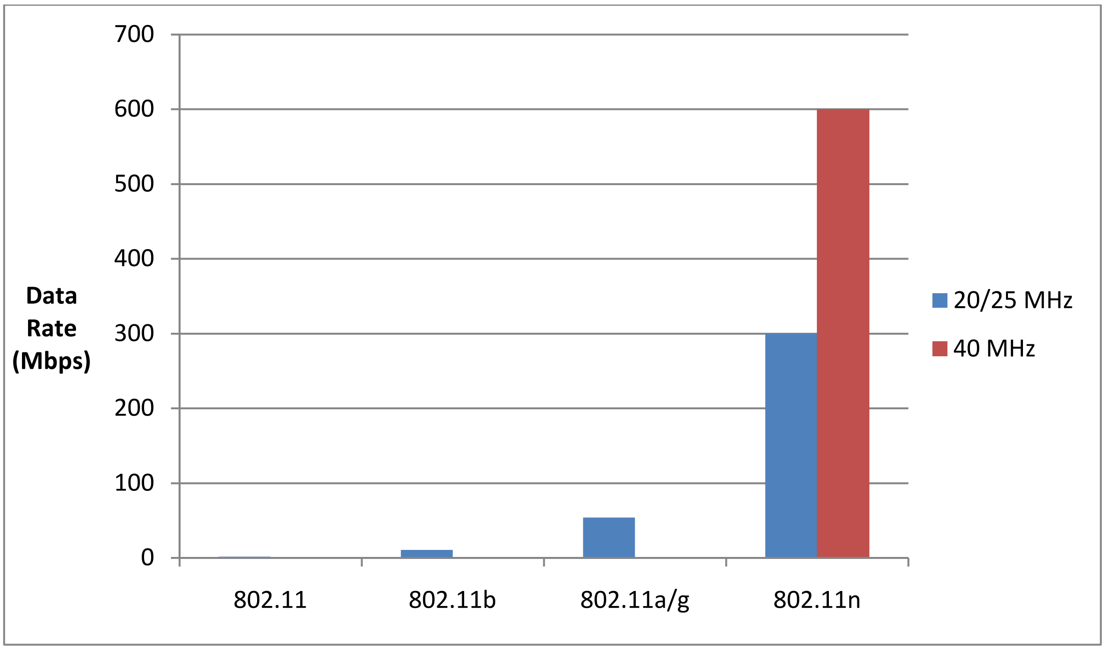


Figure 2 PHY data rate increment of 802.11

The IEEE 802.11 has experienced a five-fold increment in the data rate in every 'generation' of its amendments by the adoption of each new PHY technology. This rate continue to increase in 802.11n with a theoretical data rate of 300 Mbps in 20 MHz operation, as well as a data rate of up to 600 Mbps in the 40 MHz operation. This has shown an exponential increase which can be observed in Figure 2 above.

2.3 High Throughput and IEEE 802.11n Development

According to Thomas and Tokunbo, the development of the IEEE 802.11n amendment to the original standard began in late 2003, when the Institute of Electrical and Electronics Engineers (IEEE) forms the High Throughput Task Group, or the Task Group n (TGn), to start working on the specification, with the Project Authorisation Request (PAR) form, submitted and agreed by the 802 working group, the PAR defined the following scope and purpose by the task group:

The scope of this project is to define an amendment that shall define standardized modifications to both the 802.11 physical layers (PHY) and the 802.11 medium access control (MAC) layer so that modes of operation can be enabled that are capable of much higher throughputs, with a maximum throughput of at least 100 Mbps, as measured at the MAC data Service Access Point (SAP). IEEE, 2006.

Bounded by this statement, the TGn must develop and contain the WLAN operation mode of up to 100 Mbps, which is the very first targeted achievable throughput that doubles the maximum rate of 54 Mbps in the 802.11a/g specifications. There is also another five criteria form that must demonstrate the necessity of creating an amendment to the standard by the study group (Rosdahl, 2003):

- Broad market potential.
- Compatibility with existing IEEE 802.1 architecture.

- Distinct identity from other IEEE 802 standard.
- Technical feasibility.
- Economic feasibility.

The purpose of having these criteria in mind when developing the next generation WLAN standard is to make sure that the result forms a marketable product that can be easily identified between different but similar technologies that exist nowadays.

In year 2004, TGn's call for proposal has attracted many hardware and networking company companies to submit their suggestion to the new amendment, which then resulted a number of 61 proposals submitted to IEEE (J. Farwell, 2006). But then it has been short-listed down into two main proposal teams in early 2005, which is the TGn Sync and World Wide Spectral Efficiency (WWiSE). Both groups are formed by several companies, the TGn Sync group founded by Agere, Atheros Communications, Cisco, Intel, Philips, Sony, to name a few. They proposed the use of 40 MHz bandwidth, MIMO, and other more advanced signal processing method to achieve a transfer rate of up to 315 Mbps. Meanwhile, the WWiSE, was formed by Airgo Networks, Broadcom, Conexant, Motorola, and Texas Instrument, they suggested a simpler upgrade which can be introduced to the market quickly, which was based on the 802.11a standard, using the 20 MHz bandwidth and MIMO to achieve a practical throughput of 135 Mbps.

Since neither proposal gain a majority of vote after a few months of competition between TGn Sync and WWiSE, and due to significant supportive of companies in both the proposals, as a result, a joint proposal group, which is also known as TGn Joint Proposal was formed. This joint proposal group is formed by members from both proposal groups, and they submitted a new proposal by combining benefits from both proposal groups that previously defined, and this resulted the very first draft version of the IEEE 802.11n, or popularly known as draft version 1.0.

Chipmakers and Original Equipment Manufacturers (OEMs) started building the draft-n or pre-n version devices based on the draft v1.0 and introduced to the business and consumer market which has been widely welcomed because of the remarkable improvement offer by the new standard, in the area for example, throughput, link reliability, and also the backward compatibility with 802.11b/g which is widely installed nowadays. It then quickly followed by a new draft version 2.0 introduced in year 2007 that implemented more new optional upgraded features that again increase the speed and coverage dramatically. During the write up of this dissertation, the final confirmation details of 802.11n are still under discussion and the final ratified version are expected to be revealed in January 2010.

2.4 Features overview of IEEE 802.11n

To improve the throughput of 802.11n over what can be achieved by existing 802.11a/g, we can primarily identify the improvement over 2 major part, first is to improve the effectiveness and robustness of the wireless transmission via the PHY layer, second is to implement a better efficiency of MAC format.

Table 2 below shows the major and optional features that can be found in the new IEEE 802.11n amendment. As can be observed, the major mandatory enhancements mostly apply to the PHY layer with extensive implementation of OFDM and MIMO, or Spatial Division Multiplexing (SDM) technology to achieve a higher data rate.

	Features	Major	Optional
PHY	PHY operation	OFDM and MIMO (SDM)	TxBF and STBC
	Spatial Streams	1, 2	3, 4
	Channel Bandwidth	Standard 20 MHz	Channel Bonding 40 MHz
	OFDM pattern	OFDM with standard GI	Short GI
	Wireless coding	Convolution coding	LDPC coding
MAC	MAC format	Mixed	Aggregated

Table 2 Major and optional features of IEEE 802.11n standard

Optional features are also important by increasing the robustness in the operation of wireless transmission. For example using Space Time Block Coding (STBC) and Transmit Beam Forming (TxBF) in multiple transmit antenna to improve the received signal strength as well as increasing the effectiveness of reconstruction of data in the receiver. Low Density Parity Check (LDPC) is also adopted to produce a more robust channel code. More spatial streams can be generated by the use of more antenna, at least as many antennas as spatial stream. The use of shorter Guard Interval (GI) can also increase the PHY performance. Channel bonding is used to increase the conventional 20 MHz channel bandwidth by doubling it to a 40 MHz wide spectrum.

The first draft version (draft v1.0) comprises the use of MIMO up to two spatial streams, in 2.4 GHz operation, with optional 40 MHz bandwidth. In draft v2.0, products with dual-band functionality was introduced, they can operate in both 2.4 GHz and optional 5 GHz spectrum, other features implemented such as short GI, increased antenna chain are used to increase the PHY data rate up to 300 Mbps. Improved MAC format are also introduced such as MSDU and MPDU which also dramatically increasing the overall throughput.

2.5 Chapter Summary

This chapter give some introduction idea on how the overall 802.11 standard has evolved since the very first standardisation in 1997, it has 12 years of history in development until now, and is a well established standard with millions of users worldwide. 802.11n has became one of the major enhancement in the development of WLAN, in a nutshell it provide higher speed and wider coverage, the next chapter will provide more detail on how these enhancements can be achieved.

Chapter 3 Features detail of IEEE 802.11n

This chapter will consider some of the major features implemented in the PHY layer of 802.11n in more detail, and briefly discuss about the advancement in MAC.

3.1 Orthogonal Frequency Division Multiplexing (OFDM)

3.1.1 OFDM background

The enhancement in PHY layer of 802.11n is based upon the 802.11a Orthogonal Frequency Division Multiplexing (OFDM) structure. OFDM is a proven robust technique which has been widely implemented in many wideband, high speed digital communication systems, including transmission over coppers wires, or wireless transmission in a frequency selective fading environment, such as ADSL, digital television broadcasting, and etc.

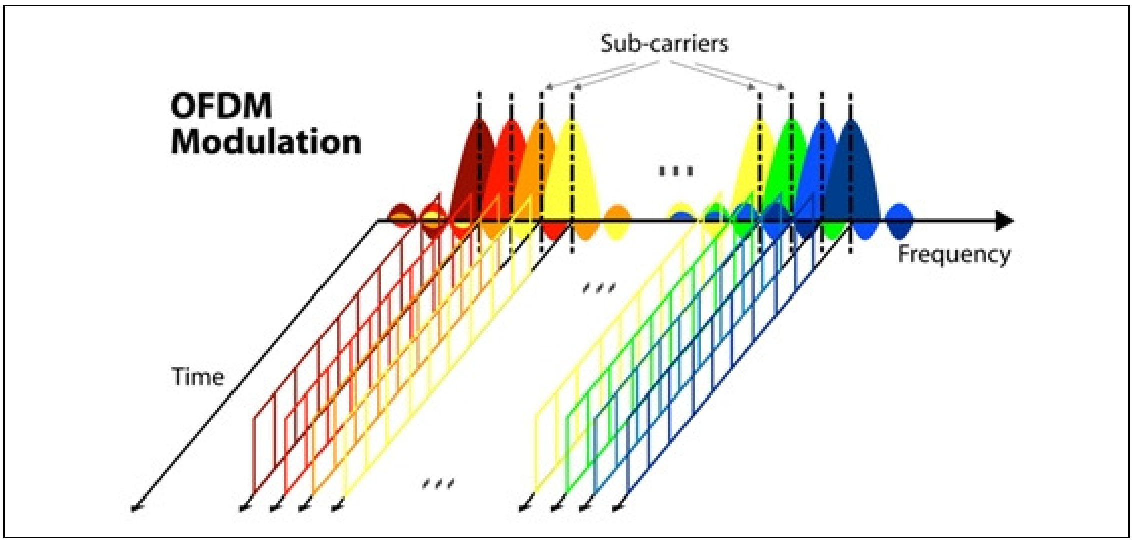


Figure 3 Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a multicarrier transmission technique, as we can observe in Figure 3 above, it shows the operation of a OFDM modulation, instead of transmitting a signal by modulating it into a single carrier, OFDM subdivide the available frequency bandwidth into many subcarrier, each subcarrier is precisely divided so that they do not overlap each other, therefore creating a mathematically orthogonal channel transmission scheme, each one is then being modulated by a lower rate of data stream simultaneously.

Another example of operation of OFDM with four subcarriers is plotted in the time domain waveform shown in figure 4 below. Each subcarrier shown below is then being modulated by an intelligence signal, and then forming and transmitting the composite waveform by summing each of the modulated signals.

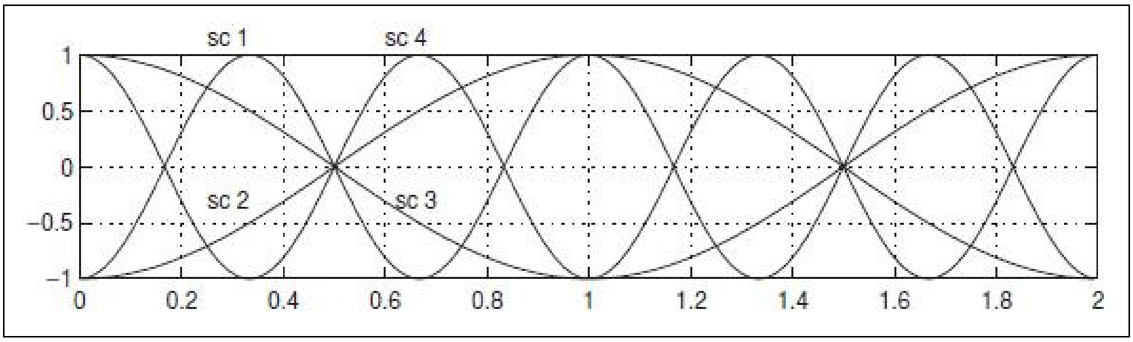


Figure 4 Frequency w.r.t normalised time (Time domain waveform)

One of the advantages of using OFDM is that it has the ability to cope with severe channel conditions, for example only a few subcarriers are impacted by a deep fade or narrow band interference, smaller sizes of information data are carried by each individual subcarriers, if error occur in either one or two of the subcarriers, they can be protected and recovered by forward error correction codes more easily, only the error occurred subcarrier need to be retransmitted, unlike the single carrier modulation, the whole bunch of data need to be retransmitted if any of the packet contains error detected in the receiver. OFDM can tolerant with time synchronisation errors, which is one of the important factors in high data rate system. In addition, OFDM is more bandwidth efficient, because nearly a square power spectrum can be created with narrow subcarriers with each subcarrier supporting a constellation with many bits per symbol (Eldad and Robert, 2008).

3.1.2 OFDM compare to Single Carrier Modulation

As discussed before, for single carrier transmission, data bits are modulated onto a single frequency, and then the modulated signal is transmitted sequentially with time. For example, let the single carrier modulation has a symbol rate of T_S which occupied a bandwidth B, where B is typically the inverse of T_S, and if the transmission has a transmission delay spread of τ_m, in order to have a reception without any Inter Symbol Interference (ISI), the following condition must be satisfied.

$$\tau_m \ll T_S$$

As a result, the possible bit rate

$$R_b = log_2(M){T_S}^{-1}$$

for a given single carrier modulation scheme is limited by the delay spread of the channel (Henrik and Christian, 2005).

Meanwhile, multicarrier transmission like OFDM can overcome this problem by modulating smaller data bits into smaller N subcarrier across the available bandwidth. A comparison between single carrier modulation and OFDM with $N = 4$ modulation is shown in figure 5 below.

The single carrier modulation can be seen as a serial transmission scheme, while the OFDM can be regarded as a parallel transmission which it will not directly affect the total bandwidth that is needed. Each subcarrier has a bandwidth of B/N, while T_S, the symbol duration can be increased by a factor of N, which has a N times higher data rate for a given delay spread, but the factor N cannot be increased arbitrarily because longer symbol will make the transmission very sensitive to the time incoherence of the channel (Henrik and Christian, 2005).

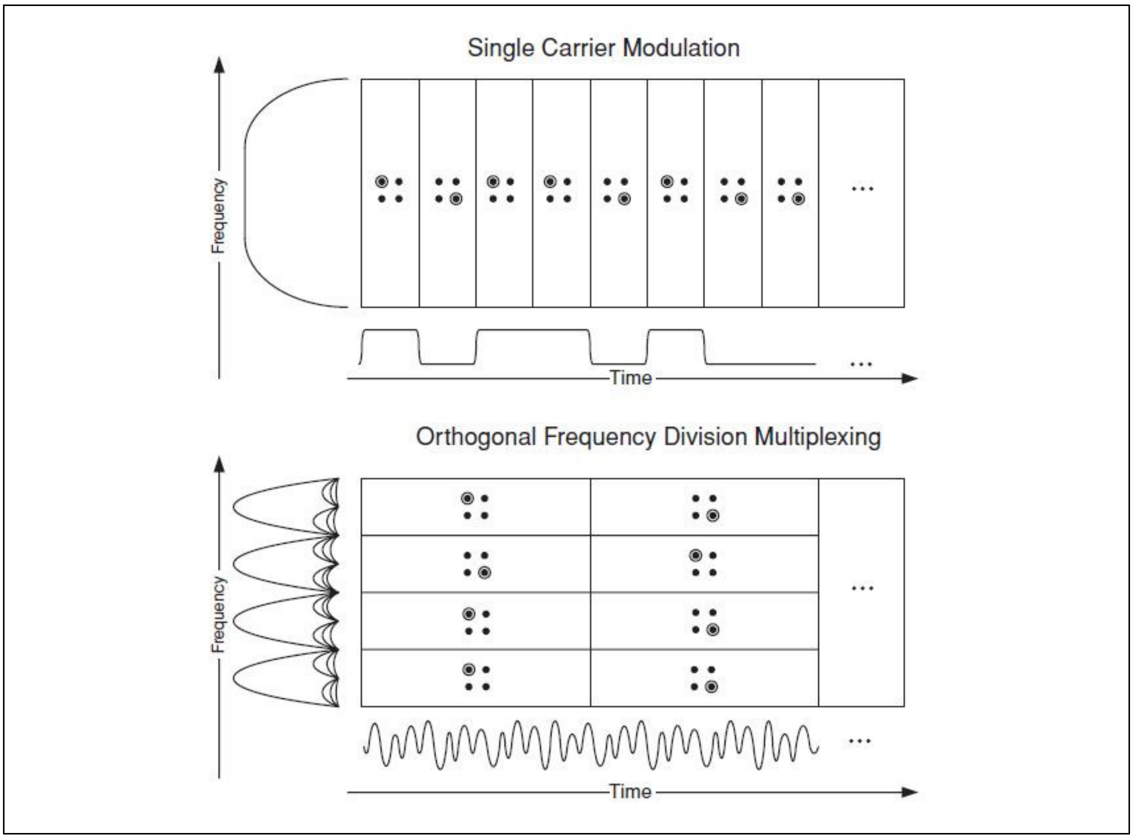

Single Carrier Modulation
Frequency
...
Time
...
Orthogonal Frequency Division Multiplexing
Frequency
...
...
Time

Figure 5 Single carrier compare to Multicarrier OFDM

From Figure 5 above, we can observe that the time varying frequency of the OFDM signal has a larger variation compare to the single carrier modulation, resulting a higher peak-to-average ratio, and the signal of OFDM does not show any form of similarity like the single carrier modulation having a theoretically constant envelope, these are basically the limitation of OFDM compare to single carrier modulation. But, in order for the single carrier modulation to transmit a higher data rate with the limited bandwidth, it needs to utilise a higher order modulation in the amplitude and phase, this results in the peak-to-average ratio of the signal approaching the same like OFDM waveform. So, the difference between single carrier modulation and OFDM in the peak-to-average ratio is very small (Eldad and Robert, 2008).

3.1.3 Operation of OFDM

In order to generate a OFDM signal successfully, the spacing between the subcarriers must be carefully separated to maintain the orthogonality of each subcarriers. Each subcarrier is then assigned the required data bits to transmit, different number of data bits can be modulated onto each subcarrier depend on the modulation, or constellation scheme, for example, Binary Phase Shift Keying (BPSK), only binary number '0' and '1' can be transmit on each modulation, Quadrature Phase Shift Keying (QPSK), combination of two digit binary number can be transmitted, Quadrature Amplitude Modulation (QAM), or etc, the usage of different constellation schemes are based on different implementation purposes. Figure 6 below shows the 3 important components in the OFDM transmission for the transmitter and receiver.

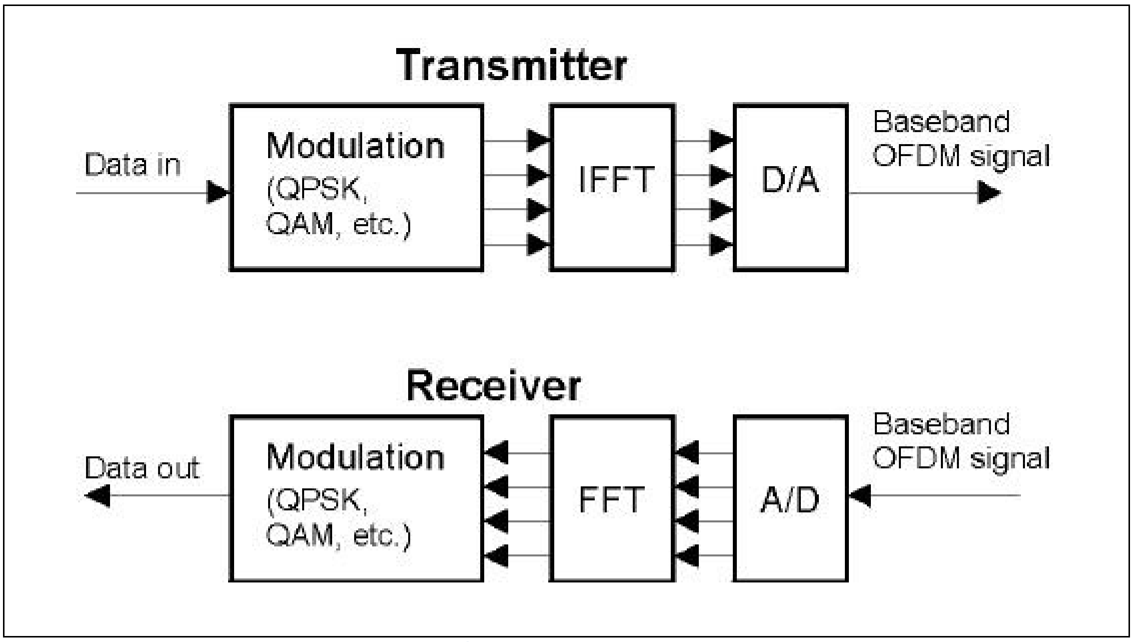


Figure 6 Basic flow chart showing OFDM transmission

After each subcarrier are modulated by the assigned data bits, and since the modulation of data bits were done in the frequency domain, so it is necessary to transform this spectrum back into its time domain representation for the practical transmission of signal, by using Inverse Fourier Transform (IFT), or more typically in many communication systems, this can be easily done by using Inverse Fast Fourier Transform (IFFT). The transformation by IFFT is a very simple method of IFT that is very efficient and it can ensure the orthogonality between each subcarrier.

A baseband OFDM waveform can be constructed as a IFT of a set of coefficients X_k (Eldad and Robert, 2008),

$$r(t) = \frac{1}{N}\sum_{k} X_k \, e^{j2\pi k \Delta_F t} \quad 0 \leq t < T$$

where, X_k is the modulated symbols,

Δ_F is the subcarrier frequency spacing,

T is the IFT symbol period, where $\Delta_F = {}^{1}\!/_{T}$,

N is the number of samples in the IFT,

After the IFFT generated a signal to be transmitted in digital discrete time format, it is then converted to its analogue representation by a digital to analogue, or D/A converter, which is then feed to the antenna for transmission.

At the receiver, it is just a reverse operation of the transmitter as shown in Figure 6 above, the antenna at the receiver feed the received signal to a analogue to digital, or A/D converter, to reconstruct the received analogue signal to its digital representation, then instead of the IFFT operation, a Fast Fourier Transform (FFT) is used to find the equivalent frequency spectrum from the time domain signal, the FFT equation below is then use to extract the modulated symbol from the received waveform (Eldad and Robert, 2008),

$$X_k = \sum_{n} r(n) e^{-j2\pi k \cdot n / N}$$

After the FFT transform, the data being modulated on each subcarrier can be recovered, the next step is to feed these raw data to the next demodulator block to demodulate it into the data that we needed. And basically this concludes the fundamental OFDM transmission.

3.1.4 Guard Interval of OFDM

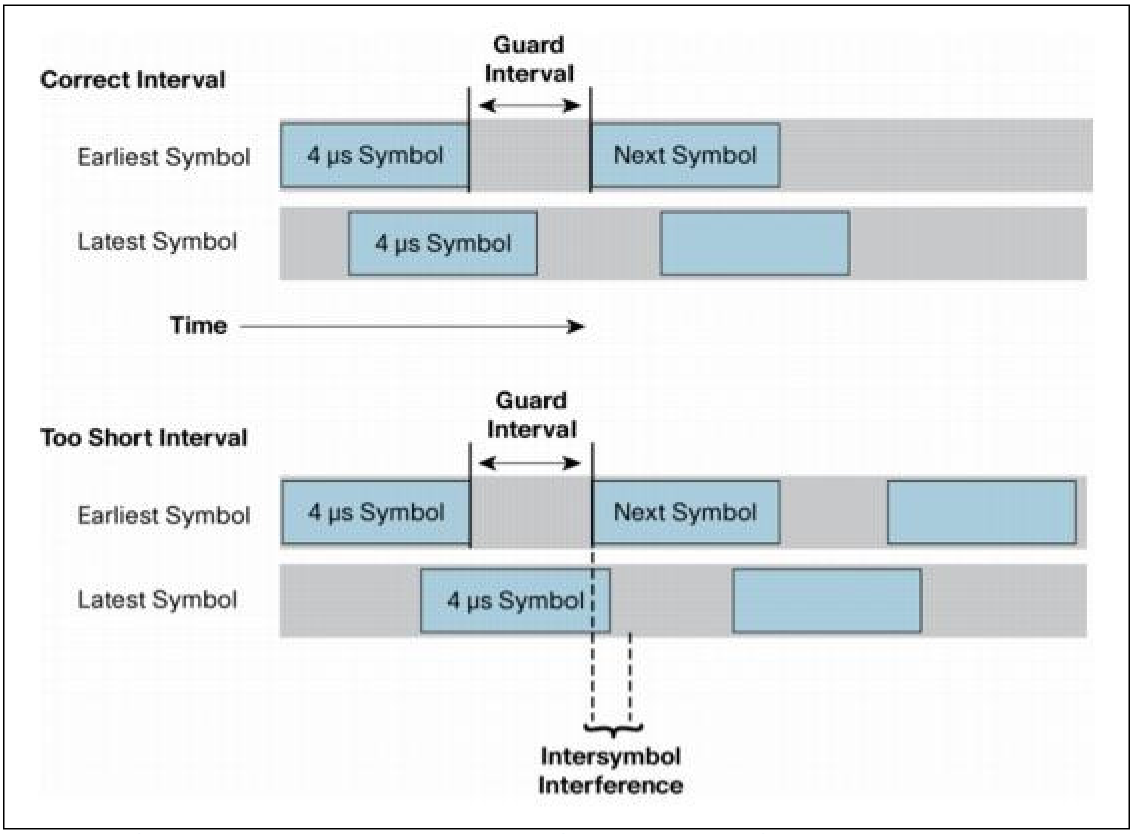


Figure 7 Guard Interval implementation in OFDM (Picture, Cisco.com)

One of the advantages of using OFDM is its ability to combat with multipath delay fading, basically this is a nature of multicarrier modulation, where they usually have a longer symbol period, which can effectively reduces the ISI problem. The robustness against multipath delay can be further improved by introducing an additional period between two adjacent symbols with the use of a Guard Interval (GI) as shown in Figure 7 above.

The GI allows the multiple copies of the same signal that arrived at the receiver at a later time through different path of travelling to be died out before the next data symbol is gathered. From Figure 7 above, we can noticed if a correct time interval is chosen, the latest symbol arrived later can still stay in the 'safe zone' of GI, meanwhile for a shorter GI, the latest symbol do not have enough time to fade out before the next data symbol is gathered, therefore both of the interfering each other causing ISI that greatly reduce the SNR rate.

The GI must be properly allocated to make sure the echoes of the signal due to multipath delay stay within the GI time interval. A shorter GI causes ISI to happen more rapidly, a longer GI take more time resources, reducing transmit efficiency. Legacy 802.11a/g equipment uses 800 ns as the GI, which gives an allowance of distance for 800 ft in between transmitter and receiver. For a smaller environment, 802.11n provides an optional GI which reduced to 400 ns, greatly reducing the symbol time from 4 ms to 3.6 ms, effectively increasing the data rates by 10%.

3.2 Multiple-Input Multiple-Output (MIMO)

3.2.1 MIMO background

Spatial Division Multiplexing (SDM), or more commonly known as Multiple-Input Multiple-Output (MIMO), or sometimes also refer to as multiple antenna, or smart antenna technology, is consider as one of the most important advancement in 802.11n. MIMO generally mean two or more than two channel inputs are being used simultaneously with two channel outputs. The use of multiple antennas has been introduce for quite some time, but a system with multiple antennas does not always mean that SDM or MIMO is implemented, it could probably be multi-channel that switch between antennas in the transmitter or receiver, which normally could be seen in 802.11a/g devices. Figure 8 below shows the different between types of antenna combination to create different communication channel.

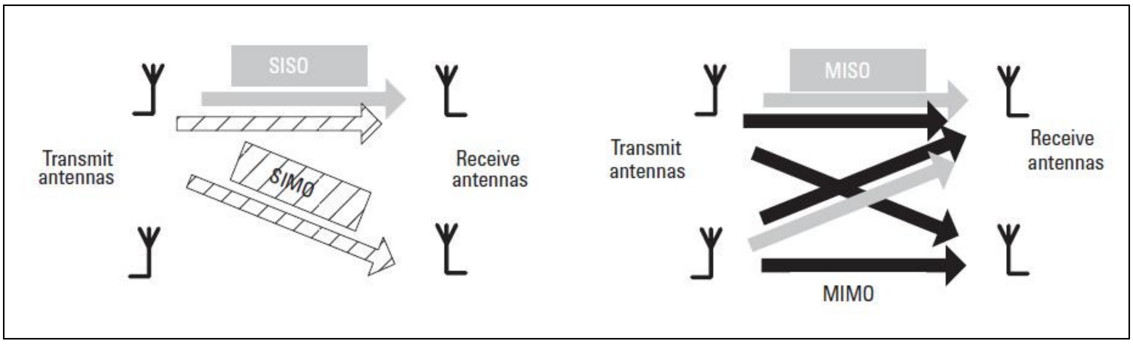


Figure 8 Channel variation with the use of different radio combination

One analogy can be used to understand MIMO better is that we can imagine the transmission from one end to another end is connected with a pipe, from Figure 8 above, a SISO contains only one pipe connecting the transmitter and receiver, while a MIMO system can have four pipes connecting each other. If a one-pipe connection means a dedicated path for a transmission, we need also to take into account the effect of any types of interference affecting the pipe during transmission, if the interference greatly degrade the transmission, the pipe make 'leak' the data packet without any other pipe as a backup, therefore, retransmission is needed. Meanwhile, the MIMO channel has four pipes for transmission, if either pipe is affected by interference, the other pipe will take over making sure the data can be safely arriving at the destination. In another word, each pipe increases the Signal to Noise Ratio (SNR), as a result, the better the SNR, the more information can be transmitted and received.

For a standard single antenna transmission scheme, the channel capacity can be estimated as follow,

$$C = B \cdot log_2(1 + \rho)$$

where, C is the channel capacity (bits/second)

B is the occupied bandwidth

$\rho\ is\ the\ SNR\ (in\ linear\ ratio)$

as we can see, the channel capacity is proportional to the occupied bandwidth and SNR, which mean if we would like to increase the channel capacity, we can either increase the bandwidth or SNR. But the use of spectrum normally assigned and fixed by some regulatory authorities, which mean altering the bandwidth is prohibited. Increasing SNR can be done by using a more complex modulation scheme, for example 256 QAM, but this result in compensation in the need of process power, increasing the manufacturing cost, also it only increase logarithmically.

Meanwhile, the MIMO capacity can be express as,

$$C = \sum_{i\,=\,1\ to\ N} B \cdot log_2\left[1 + \frac{\rho}{N}\sigma_i{}^2(H)\right]$$

where, N is the number of pairs of transmit-receive antenna

$\sigma_i{}^2$ is the singular values of the radio channel matrix, H

basically the equation above is derive from the previous equation for the single antenna condition which shows that the channel capacity is proportional to the number of pairs of transmit-receive antenna chain which increase linearly. Therefore, number of effective antenna can be added to increase the channel capacity and transmit data rate without changing the channel bandwidth or using a complex modulation, which is a more practical and economical way to improve the wireless transmission.

MIMO transmission can be observed as, for example a MIMO system has j transmit antennas and i receive antennas, the channel response from any j to any i antennas is the sum of the multiple reflection of path which has a different propagation delay, also the total received signal can be assumed as the sum of the signals from all transmit antennas (Thomas and Tokunbo, 2008).

A simple way to represent a MIMO channel as described above can be written with a matrix $\boldsymbol{H}$ with i by j elements, each element $h_{i,j}$ representing the channel gain from transmit antenna j to receive antenna i. The received signal $\boldsymbol{y}$ with i by 1 elements can be written as the multiplication of $\boldsymbol{H}$ with the transmitted signal $\boldsymbol{x}$ which has a j by 1 elements as below,

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n}$$

where, $\boldsymbol{n}$ is Additive White Gaussian Noise (AWGN)

The above equation can be further expanded, in the time domain MIMO channel which is written as (J. Barry, E. Lee, and D. Messerschmitt, 2004),

$$r(t) = \sum_{k=-\infty}^{\infty} H(t - kT)a(k) + n(t)$$

with,

$$H(\tau) = \begin{bmatrix} h_{1,1}(\tau) & \cdots & h_{1,M}(\tau) \\ \vdots & \ddots & \vdots \\ h_{N,1}(\tau) & \cdots & h_{N,M}(\tau) \end{bmatrix}$$

$$a(k) = \begin{bmatrix} a_1(k) \\ \vdots \\ a_M(k) \end{bmatrix} \quad n(t) = \begin{bmatrix} n_1(t) \\ \vdots \\ n_N(t) \end{bmatrix}$$

where, $H(\tau)$ is the time-domain MIMO channel,

$\boldsymbol{a}$ is the transmitted sequence,

$a(k)$ is vector elements across transmit streams in $\boldsymbol{a}$,

T is the time interval between each element $a(k)$,

$n(t)$ is noise vector

M is the maximum assumed transmit antenna number

N is the maximum assumed receive antenna number also $H(\tau)$ is a matrix with a unique channel impulse response for every element $h_{i,j}(\tau)$.

Typically, every element $h_{i,j}(\tau)$ in matrix $H(\tau)$ can be written as (J. Terry and J. Heiskala, 2002),

$$h(\tau, t) = \sum_{n} \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t))$$

where, τ is the delay index of channel response,

n is the index of the reflection path,

$\alpha_n(t)$ is the channel attenuation over path n at time t,

$e^{-j2\pi f_c \tau_n(t)}$ is the carrier phase offset due to path delay τ_n

$\delta(\tau - \tau_n(t)$ is the propagation delay over path n.

the indices i and j are removed for better notation, and t is added to show the time-varying nature of the channel.

3.2.2 Channel Diversity – Space Time Block Coding (STBC)

Channel diversity is not a new technique which can be seen in some of the currently available 802.11a/g products, it is widely use to reduce error in single antenna transmission scheme, increasing the signal path robustness by the use of channel path diversity.

In some conventional 802.11a/g devices with two antennas, switched diversity is used either in the transmitter antennas or receiver antennas, under this condition, it is still consider as a Single-Input Single-Output (SISO) system because only one transmission chain is established between one transmitter and one receiver. An Automatic Gain Control (AGC) is connected in between the antennas at the receiver to choose and switch which antenna to use that gives a better signal strength. Meanwhile, provided the transmitter has sufficient knowledge about the channel, it can also switch between two transmitter and use the antenna that gives a better path.

Similarly, but added with more complexity with the use of two or more transmit antennas simultaneously instead of switching, as shown in Figure 9 below, a 'modified' signal is transmitted together with the original signal. This involves some modification to the transmitter hardware to double the transmit chain. Based on *Alamouti*, the inventor of STBC, he demonstrated the possibility to increase the performance using STBC in the multiple transmit antennas as if combining the maximum ratio of multiple receive antennas. Given two inputs x_1 and x_2 which is sequentially in time, the output can be written as (Alamouti, 1998),

$$y_1 = \begin{bmatrix} x_1 \\ -{x_2}^* \end{bmatrix} \qquad y_2 = \begin{bmatrix} x_2 \\ {x_1}^* \end{bmatrix}$$

where, x_1 and x_2 are the symbols for first spatial transmit stream

$-{x_2}^*$ and ${x_1}^*$ are the second symbols transmit stream

the equation shows that the first transmit stream is remain unchanged, with the second transmit stream generating the space-time code.

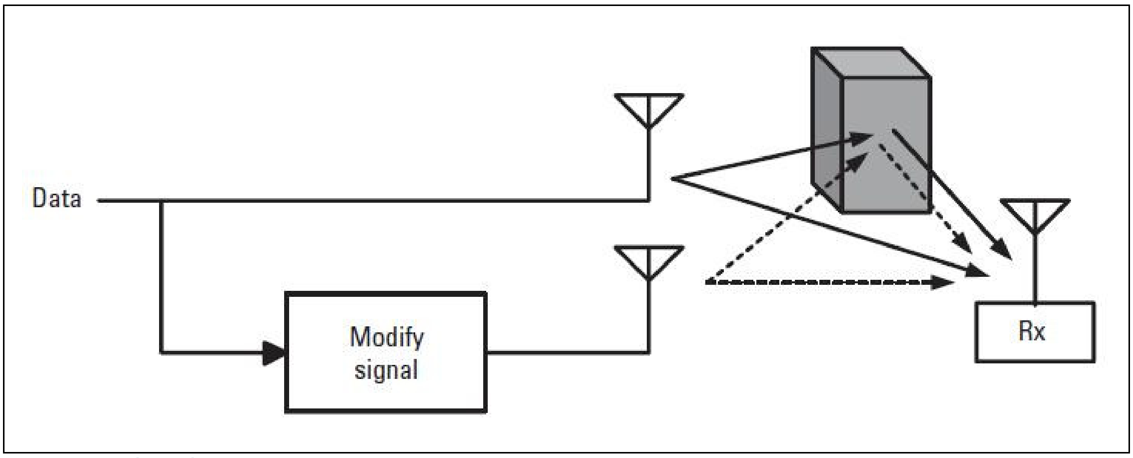


Figure 9 STBC operation with two transmit antennas

Symbol receive by the receiver are (Alamouti, 1998),

$$r_1 = [h_1 \quad h_2] \cdot \begin{bmatrix} x_1 \\ -{x_2}^* \end{bmatrix} + n_1$$

$$r_2 = [h_1 \quad h_2] \cdot \begin{bmatrix} x_2 \\ {x_1}^* \end{bmatrix} + n_2$$

by combining the vector $[r_1, {r_2}^*]$ the transmitted symbol can be recovered using only one receive antenna, as a result, the use of STBC is actually a Multiple-Input Single-Output system (MISO) system, which is use to provide a transmission path with increase robustness. Where 'Space' means the separation of different antennas, and 'Time' here refers to the OFDM symbols pairs in the reversal of the transmission time (Alamouti, 1998).

3.2.3 Spatial Division Multiplexing (SDM)

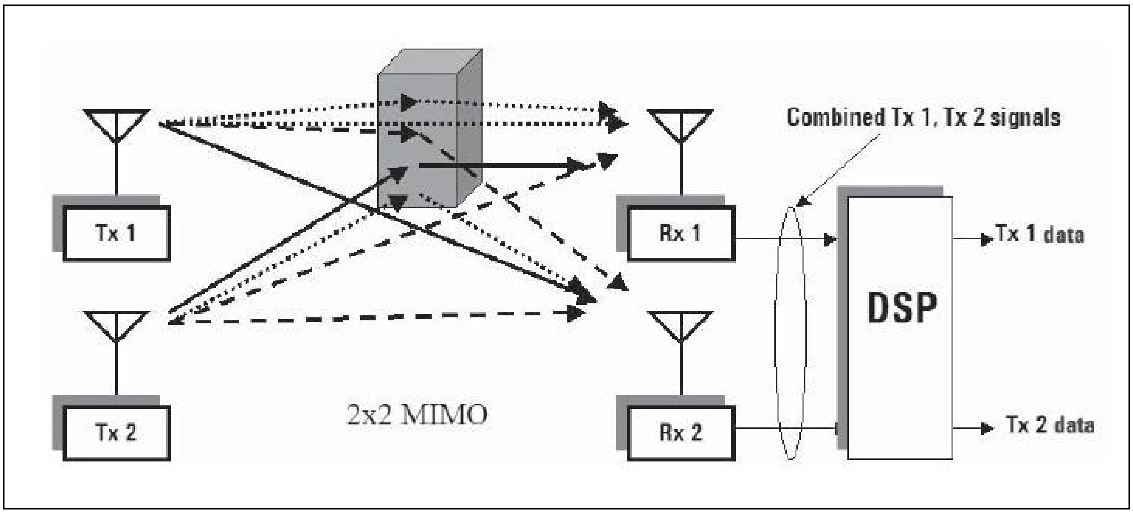


Figure 10 Basic 2 by 2 MIMO SDM system (Picture, Agilent.com)

One of the key techniques in MIMO is Spatial Division Multiplexing (SDM), it can achieve a higher throughput with the utilisation of multiple antennas, transmitting multiple data streams within the operational bandwidth, as its name suggest, the spatial stream transmission can be achieve by separating the transmit antenna in different location, or more generally we can call this a space different, and then the different antennas transmit multiple data streams, creating a multiplexing of transmission in a spatial dimension, each receive antenna can individually decode the signal at the receiver, this technique involves both at the transmitter and receiver as shown in Figure 10 above. The gain in SDM can also increase the channel capacity.

The spacing between antennas can be important and therefore must be chosen carefully, a change in one wavelength, λ, cause 2π phase shift of the carrier, as a result, it could change the signal fading, also known as space loss differently, so, as a rules of thumb, a common way to separate the antenna can be,

$$antenna\ spacing = \frac{\lambda}{2}$$

where, λ differs with different carrier used (i.e. 5 GHz or 2.4 GHz)
a common selection for antenna spacing for 5 GHz and 2.4 GHz are 2.7 cm and 6.0 cm respectively, both for the transmitter and receiver.

Based on the time-domain MIMO channel mentioned before, MIMO SDM can be modelled again for each subcarrier in the frequency domain as,

$$[R(k) = H(k)S(k) + N(k)]$$

with $H(k)$ as a MIMO channel with complex scalar value for each subcarrier k,

$$H(k) = \begin{bmatrix} h_{1,1}(k) & \cdots & h_{1,M}(k) \\ \vdots & \ddots & \vdots \\ h_{N,1}(k) & \cdots & h_{N,M}(k) \end{bmatrix}$$

every elements $h_{i,j}$ in matrix $H(k)$ is a element of the channel gain and phase for each subcarrier k, in complex scalar form.

The receiver must be able to detect every channel response (every row in the matrix $H(k)$ represented the receive stream) as a unique stream to be able to recover the original transmitted data stream, $S(k)$. This involve an understanding on the channel knowledge, as well as an appropriate spacing between antenna which affect the quality of signal seen on the receive antennas.

3.2.4 Transmit Beamforming

Transmit Beamforming is another extension from MIMO, its main purpose is to achieve an improvement in signal strength on the receiver with the use of channel estimation to find the transmission signal's spatial matrix. In another word, if two signals are being sent simultaneously from two different antennas, the signal can be seen by the receiver as a destructive interference because a delay signal that arrived later at the receiver can be out of phase compare to the previous signal received. The signal received in different phase can cause a degradation of the overall signal receive strength.

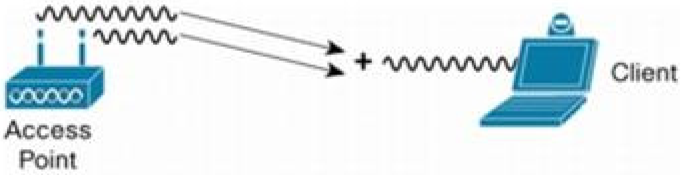


Figure 11 Constructive Transmit Beamforming

Transmit Beamforming can be used to improve the situation by using the two different chains of data transmission in a constructive way, provided the transmitter know about the channel knowledge by feedback information from the receiver, then radio's phase is adjusted carefully for the following transmission. As a result, this technique improves the SNR at the receiver which then effectively increase the transmission quality.

To be able to understand the channel optimally, a Singular Value Decomposition (SVD) can be used to decompose the MIMO channel which simplifies the MIMO detection as below (S. Haykin, 1996), for any i by j channel matrix H_k,

$$H_k = U_k \sigma_k {V_k}^H$$

where, H_k is the estimated channel matrix for subcarrier k,

σ_k is a diagonal matrix of singular values,

U_k is an i by N orthonormal matrix,

V_k is an j by N orthonormal matrix,

N is a minimum number of $\{i, j\}$

from the equation above, the optimal optimal spatial matrices V_k can be obtained, therefore, the transmitter can use it to form the transmit beamforming, so the signal receive at the receiver can now be expressed as (Thomas and Tokunbo, 2008),

$$\begin{aligned} r_k &= H_k V_k x_k + n_k \\ &= U_k \sigma_k {V_k}^H V_k x_k + n_k \\ &= U_k \sigma_k x_k + n_k \end{aligned}$$

From the above equation, the singular values on the diagonal matrix σ_k can be represented as the received signal strength for each element, x_k, at the receiver.

3.3 Other optional enhancements

3.3.1 Channel bonding

The original 802.11 and the earlier amendment 802.11b which uses the DSSS radio, they occupied a spectrum which is 22 MHz wide. While the newer amendments 802.11a/g improved by shrinking the spectrum smaller which uses only 20 MHz wide. Spectral efficiency is used to measure the performance of the radio which is calculated in bits per second per Hertz (bit/s/Hz),

$$Spectral\ Efficiency * Spectrum\ wide = Effective\ Througput$$

802.11b has a spectral efficiency of 0.5 bit/s/Hz, and since 802.11b utilises 22 MHz wide spectrum, therefore it has an effective throughput of 11Mbps. While 802.11a/g has a higher spectral efficiency of 2.7 bit/s/Hz which give a effective throughput of 54 Mbps.

As we have already seen some of the existing WLAN product, especially 802.11g which claim that they can run at a rate which doubling the standard 54 Mbps which is effectively up to 108 Mbps, these are some proprietary WLAN system which was implemented by different WLAN chips or OEM manufacturer, for example, Atheros' 'Super G', these WLAN systems do not based on the IEEE standard, so in order to benefit from the high data rate that provided by such system, same product developed by the same manufacturer must be used.

One of the key techniques to allow such system to achieve two times higher data rate is with the use of channel bonding, occupying two channels at the same time, as a result the spectral efficiency can also be double from 2.7 to 5.4 for 802.11a/g standard, therefore achieving an effective throughput of 108 Mbps.

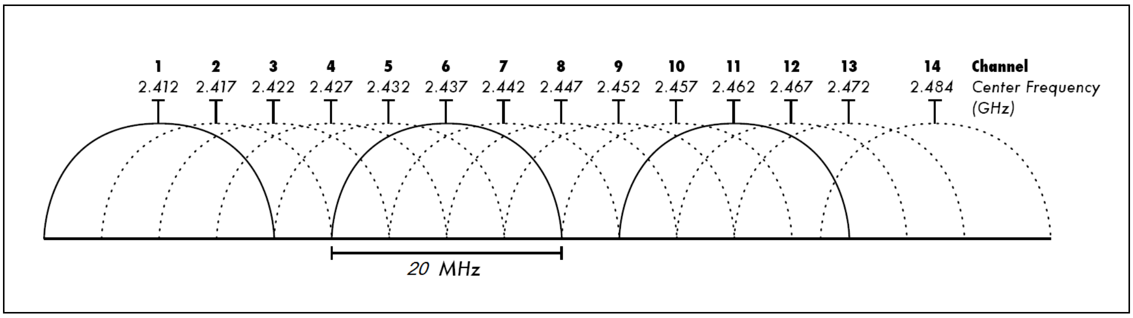


Figure 12 Spectrum of a 2.4 GHz WLAN system

As can be seen in Figure 12 above, a standard 2.4 GHz WLAN system have 11 channels, ranging from channel 1 to 11, and either one can be selected by user for the WLAN transmission in the base station (router). Again, we can observe that there are three non-overlapping channels which are channel 1, 6, and 11, by taking such property of these three non-overlapping channels, two adjacent 20 MHz channels can be bonded and form a single 40 MHz channel as in Figure 13 below.

Instead of implementing channel bonding as a proprietary system discussed above, 802.11n bring this simple technique to increase data rate as an optional standard. Also, as in Figure 13 below, in a 20 MHz wide spectrum, the top and bottom channel bound need to be reserved in order to avoid inter channel interference, by bonding two 20 MHz channels together, the bottom bound of the lower frequency spectrum and the top bound of the upper frequency spectrum can now be use to modulate and sent data, effectively increasing the data rate slightly more than just double.

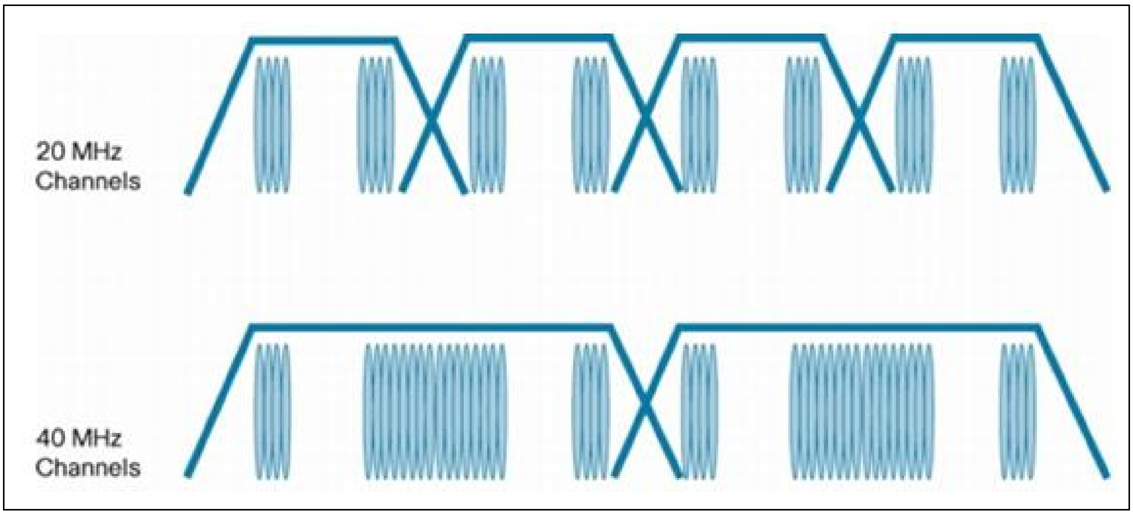


Figure 13 Channel bonding

3.3.2 MAC aggregation

Radio Preamble	Radio Header	MAC Header	Data	FCS

Radio Preamble	Radio Header	ACK

← Overhead →

← Overhead →

Figure 14 802.11 frame overhead

The improvements of 802.11n discussed above are all mainly based on the radio, or more specifically, the PHY layer of the WLAN system. And since the development of enhanced 802.11 WLAN system, for example the amendments 802.11a/b/g, most of the improvements are generally based on the PHY layer, but something important that has brought notice is the need to also improve the Medium Access Control (MAC) which is seems to be aging and not up to date to adapt with modern high speed communication.

One of the problems with existing 802.11 MAC is that the MAC layer protocol, spaces between frames, and transmitted frame acknowledgements are designed to have a fixed overhead. Effectively, these fixed overhead can sometimes longer than the overall data frame, in addition collisions and contention for the air could also reduce the maximum throughput for the 802.11 WLAN systems (Cisco whitepaper, 2007). One of the techniques used by 802.11n to overcome this problem is by implementing frame aggregation by changing the MAC frame which put more frame into a single transmission, effectively fixing the fixed overhead, collisions, and contentions problems.

Radio Preamble	Radio Header	MAC Header	Data	FCS

Figure 15 802.11 frame aggregation

MAC Service Data Units (MSDU) Aggregation

This is a more deployable technique among the two variation of aggregation. For an 802.11 system, every Ethernet frame need to be transformed to an 802.11 format then only can be transmitted to a mobile station.

MSDU = Ethernet Frame

Radio Preamble	Radio Header	MAC Header	MSDU 1	MSDU 2	MSDU 3	...	MSDU N	FCS

Figure 16 MSDU Aggregation

Instead of doing this transformation of packet type between Ethernet and 802.11 format one by one prior transmission, MAC Service Data Units (MSDU) can do a bulk process of aggregating several Ethernet frame of a same destination, then wrap all these Ethernet frame and transform into a single 802.11 frame for transmission, as shown in Figure 16 above. MSDU is more efficient compare to MPDU (which will be discussed in next section) because an Ethernet frame is much smaller than 802.11 frame (MPDU aggregation based on 802.11 frames).

MSDU can be used only to aggregate all frames that are destined to a same location, for example a mobile client to a base station, or vice versa, broadcasting to several devices from the base station is not supported by MSDU. The encryption for MSDU is done by another 'wrapping' to the 802.11 transmission frame, one restriction to MSDU is that different type of Quality-of-Service (QoS) frame must not be mixed for aggregation.

MAC Protocol Data Units (MPDU) Aggregation

RP = Radio Preamble

RH = Rapid Header

MH = MAC Header

RP	RH	MH	MH1	Data1	MH2	Data2	...	MH N	Data N	FCS

Figure 17 MPDU Aggregation

The different of MAC Protocol Data Units (MPDU) compare to MSDU is that instead of combining many Ethernet frame and transforming them into a single 802.11 frame for transmission, MPDU transform every single Ethernet frame into an individual 802.11 frame of the same destination without 'wrapping' them again into another 802.11 frame, because the aggregated frame is already started with an 802.11 header therefore the aggregated frame is ready for transmission.

Some properties of MPDU are the same with MSDU, for example the frames destined to the same location must be aggregated together, and therefore do not support broadcasting to several destinations. Also, the same type of QoS frame must be aggregated together, cross over between different types of QoS frame are not supported.

The performance of MPDU is lower because of the header of each individual 802.11 is longer than the individual Ethernet header which is being used in the MSDU. The MPDU gone worse if encryption is apply to it, since the encryption need to be applied to each 802.11 frame within the MPDU aggregation, while MSDU only need to apply an single encryption 'wrapping' to the overall MSDU aggregated frame, this further increase the length of the MPDU effective transmit MAC frame, therefore reducing the overall performance.

3.4 Chapter Summary

This concludes all the important main features implemented in the 802.11n amendment, including the 'hardware' part, for example the radio system or the PHY layer of the 802.11n, and the 'software' part, for example the MAC format improvement. A more detail of examination to the 802.11n PHY layer of 802.11n, or more specifically the OFDM and MIMO is simulated and studied in the following chapter.

Chapter 4 Methodology and Simulation

As discussed in previous chapters, most of the advanced techniques have been implemented into the 802.11n were through the enhancement in the PHY layer of the wireless radio. In this chapter, a simulation platform will be set to model this improved PHY, or more specifically the OFDM, and MIMO techniques.

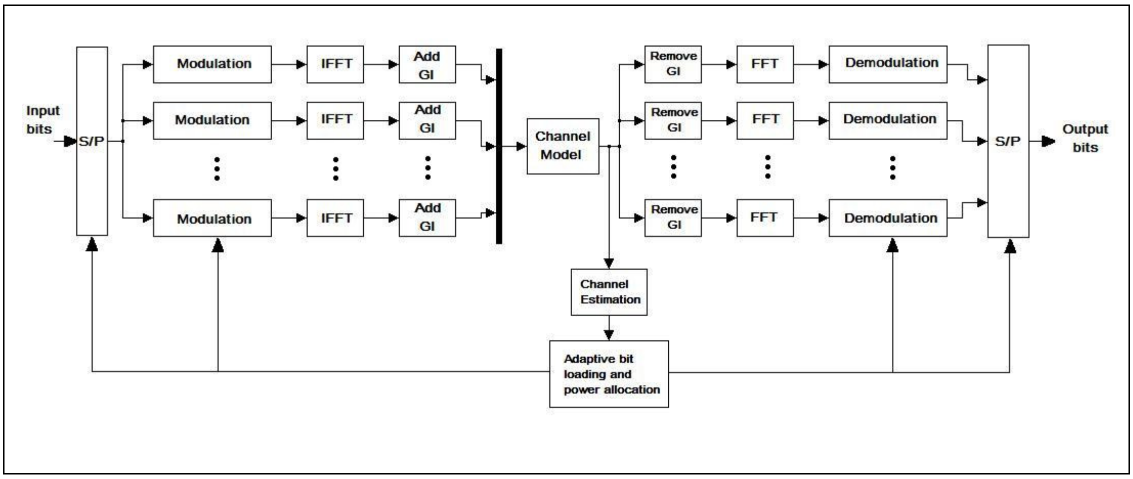


Figure 18 High level simulation block diagram

Software package MATLAB has been chosen for the simulation because of personal familiarity with it and also its ease of use and has many in-built functions that further increase our coding efficiency. The studies and development of this simulation are generally based on the code developed and published by Prateek Bansal and Andrew Brzezinski (December, 2001). This piece of code basically simulates the essential part of 802.11n, which is the OFDM and MIMO system intuitively, also with added ability to load the data bits adaptively, a high level simulation block diagram is shown above in Figure 18. Modification has been done to the code to add functionality and usability, for example an added Graphical User Interface (GUI), different delay profile selection, option for simulating with different modulation scheme, and etc., which will be discussed in more detail in the following subchapters.

4.1 MIMO and the channel model

The propagation of signal to the free air can have many characteristics, for example attenuation, multipath effects, and etc., this is normally caused by different transmission path that the signal travelled and are mainly due to reflection or diffraction of the signal, these are sometimes known as disadvantages, because it bring some serious interference to our communication system like delay spread, or more typically ISI interference as discussed before. Since the delayed signal which travelled a longer path reach the last to the destination, the different in time and power cause the reconstruction of the original signal to be interfered as in Figure 19 below.

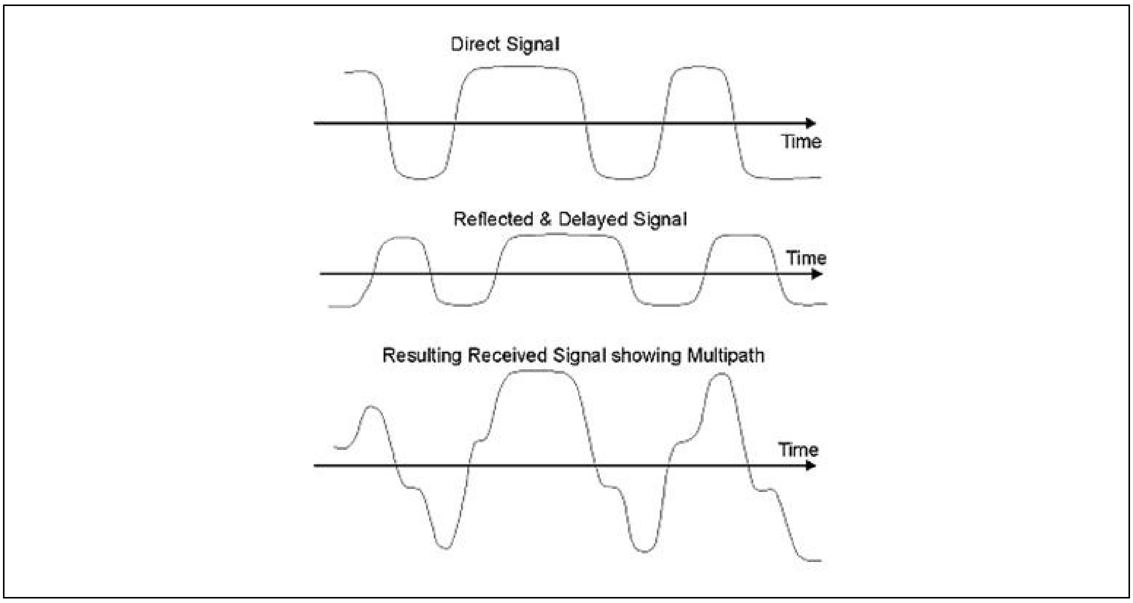

Direct Signal
Time
Reflected & Delayed Signal
Time
Resulting Received Signal showing Multipath
Time

Figure 19 Received signal interference caused by delay spread

But, if without those bouncing signal, obstacles or partitions in an indoor environment will cause no direct Line Of Sight (LOS) path for a particular signal to arrive to the destination, therefore, bouncing of signal comes into handy here for the signal to arrive to the destination which can be seen in the Figure 20 below. Figure 20 shows the reflection or fading effect of an environment using a MIMO system, which only shows some basic reflections by a few objects in a typical indoor environment, readers may convince yourself with a more exotic way of scattering, for example reflecting from ceiling, floor, and etc.

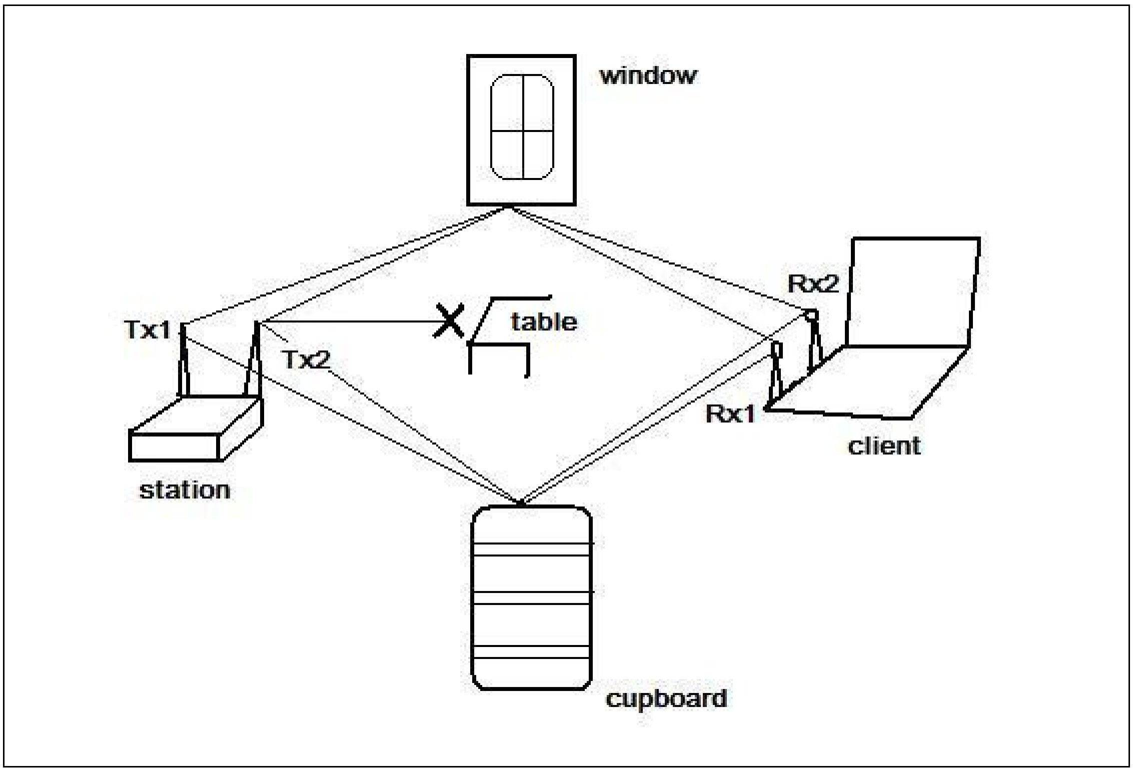


Figure 20 Multipath effects propagations

Also there is a limitation of delay spread, with a maximum delay spread of 20 μs in an outdoor environment, by keeping the delay spread in a range outline as in table 3 below, the error bits rate can be kept as low as 25 kbps.

Environment	Delay Spread	Maximum Allowable Path Length
Indoor	40 ns – 200 ns	12 m – 60 m
Outdoor	1 μs – 20 μs	300 m – 6 km

Table 3 Typical indoor and outdoor delay spread

For an 802.11 WLAN system that generally implemented in an indoor environment, we are more concerned about the variety multipath effects that the signal travelled by bouncing around items or structures in the environment. This characteristic can be modelled by Rayleigh frequency selective fading, with its ability to simulate a rich scattering environment which varied with time. Rayleigh fading model assume that the magnitude of the power of a signal will vary randomly depending on the distance pass through by the signal in the transmission medium or the communications channel, which is distributed as Gaussian random variables, this effect is illustrated in Figure 21 below.

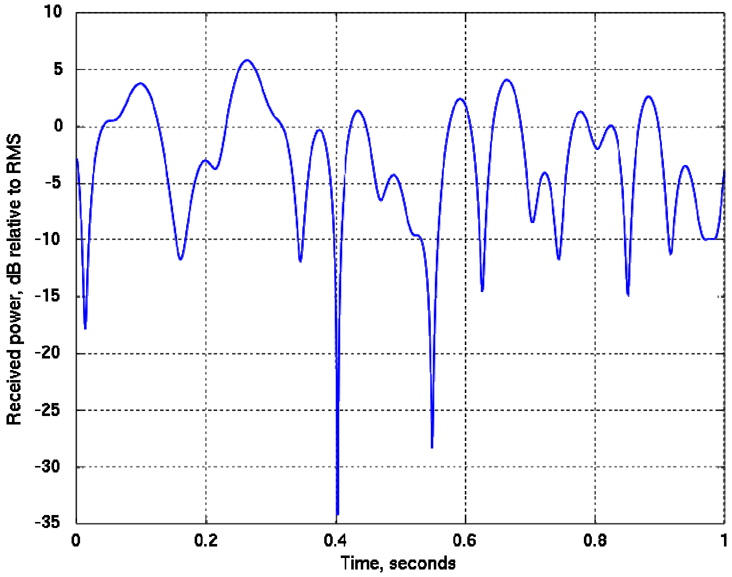


Figure 21 Typical Rayleigh frequency selective fading

In the simulation, the MIMO channel is modelled based on the equation below,

$$\boldsymbol{y}_n = \sum_{l=0}^{L-1} \boldsymbol{H}_l \boldsymbol{x}_{n-1} + \boldsymbol{n}_n$$

where, $\boldsymbol{y}_n$, $\boldsymbol{x}_n$, $\boldsymbol{n}_n$, are output, input, and additive noise respectively, and they can be vectors

$\boldsymbol{H}_l$ is the channel gain, it can be a matrix

L is the symbol periods of the delay spread

Also a power delay profile which has an exponential decaying nature is also defined based on the TGn model proposed by Vinko Erceg et al. 2004, based on the delay profile, a channel tap, matrix H_l, can be modelled and is written as the sum of a constant with LOS component matrix, and a variable with Non Light Of Sight (NLOS) component matrix, and Rayleigh matrix. A 4 by 4 antennas MIMO configuration with its channel matrices H_l can be written as (Thomas and Tokunbo, 2008):

$$H_l = \sqrt{P_l}\left(\sqrt{\frac{K}{K+1}}\begin{bmatrix} e^{j\emptyset_{11}} & e^{j\emptyset_{12}} & e^{j\emptyset_{13}} & e^{j\emptyset_{14}} \\ e^{j\emptyset_{21}} & e^{j\emptyset_{22}} & e^{j\emptyset_{23}} & e^{j\emptyset_{24}} \\ e^{j\emptyset_{31}} & e^{j\emptyset_{32}} & e^{j\emptyset_{33}} & e^{j\emptyset_{34}} \\ e^{j\emptyset_{41}} & e^{j\emptyset_{42}} & e^{j\emptyset_{43}} & e^{j\emptyset_{44}} \end{bmatrix} + \sqrt{\frac{1}{K+1}}\begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix}\right)$$

where, P_l is the overall power of channel tap l,

K is the Ricean K-factor,

$e^{j\emptyset_{ij}}$ are the elements of the fixed LOS matrix,

X_{ij} are correlated zero-mean, unit variance, complex Gaussian random variable coefficients of the NLOS, Rayleigh Matrix.

From now on, a bracketed coding extracted from a particular function file will be shown to explain some of the key functions of the coding that we are interested in, with their respective M file indicate in the caption. The full coding for the whole simulation can be found in the Appendix section attached in the end of this document.

```
H_int = 1/sqrt(2)*(randn(Mr*length(A),Mt) + j*randn(Mr*length(A),Mt));
```

Coding 1 <create_channel.m>

The model for the previous equation can be seen in Coding 1, which assume K to be zero, since K is the Ricean K-factor which is another modelling method that follows the Ricean distribution to estimate the wireless channel with the exist of LOS component, for the simplicity of the simulation, we can generally assume the transmission without the LOS component, this assumption can be practical because in a real indoor environment, signal normally do not have the LOS path as discussed before, consider the situation as in Figure 20.

By assuming $K = 0$, which will make the coefficient of the LOS component, $\sqrt{\frac{K}{K+1}} = 0$, therefore the previous equation can be written again without the LOS component as follow (assume 4 by 4 antennas combination):

$$H_l = \sqrt{P_l}\begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix}$$

Coding 1 first define the size of the channel based on the number of transmit and receive antenna, which are Mt and Mr respectively, which also depend on the length of the power delay profile, $length(A)$. And each component X_{ij} is a correlated zero-mean, unit variance, AWGN random variable coefficients which is use to represent the NLOS Rayleigh matrix between i-th receive to j-th transmit antenna. This AWGN random number can be easily generated independently for both the real and complex component with the $randn$ in-build function of MATLAB. Each channel element is assume to be independent and identically distributed (iid). The power $\sqrt{P_l}$ in Coding 1 is set to a constant because the power will be defined by the power delay profile, A as in Coding 2 below.

```
H_int2=[];
for i = 1:length(A)
    H_int2 = [H_int2;sqrt(A(i))*H_int((i-1)*Mr+1:i*Mr,:)];
end
```

Coding 2 <create_channel.m>

As discussed before, a GI is introduced to a OFDM symbol to combat with the effect of multipath that causes ISI interference. A GI allows the desired OFDM symbol's signal to be died out before the next OFDM symbol's signal could be transmitted, it is also a useful technique to be implemented to reduce the signal complexity receive in the receiver, GI extension is implemented as in Coding 3 below to be a prefix of each channel transmission block, the result shows the linearly convolved signal which is also the response of the system.

```
H_int2 = [H_int2;zeros((N-length(A))*Mr,Mt)];
```

Coding 3 <create_channel.m>

Then each channel above could be transformed into a circular convolution from their linear convolution of the signal, this operation could be done by using FFT function in MATLAB as shown in Coding 4 below, the reason for this is to make the convolved channel or signal, to be represented by their product with the linear representation (Prateek and Andrew, 2001). The result H_f could be also known as the MIMO impulse response of the channel between transmit and receive antennas.

```
H_f = zeros(Mr,Mt*(N-16));
for i = 1:Mt
    for jj = 1:Mr
        h_f = fft(H_int2(jj:Mr:(N-16-1)*Mr+jj,i));
        for k = 1:(N-16)
            H_f(jj,i+(k-1)*Mt) = h_f(k);
        end
    end
end
```

Coding 4 <create_channel.m>

Recall the MIMO channel response matrix, H_l defined in previous equation, with every elements h_{ij} represent the channel gain from receive antenna i to tranmsit antenna j. Also, throughout this simulation, we assumed that the perfect channel knowledge, which mean H_l is available to both the transmitter and receiver, therefore each of the tone of MIMO channel can be decomposed into non-interfering parallel SISO channels using SVD decomposition. As discussed before, the SVD decomposition of the instantaneous channel matrix can be represented by:

$$\boldsymbol{H}_i = \boldsymbol{U}_i\boldsymbol{S}_i\boldsymbol{V}_i^*$$

where, $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are the unitary matrics, and $\boldsymbol{V}_i^*$ is the transpose conjugate

$\boldsymbol{S}_i$ is the diagonal matrix of singular values $\boldsymbol{H}_i$

i is the i-th tone of the instantaneous channel matrix

the operation is shown below in Coding 5, where the in-build MATLAB function *svd* is used. Each value U, S, and V is saved individually in their respective vector or matrix after the decomposition. The results of this decomposition is to overcome the interference in the subchannels, and also it need to be used to optimally allocate data bits and energy by the adaptive loading and also the precoding and shaping for the transmit and receive signal before and after the IFFT and FFT process respectively, which will be discussed later. *svd* can be implemented directly to the MIMO channel provided the perfect knowledge is known at the transmitter side, which can also be used to simplifies the MIMO detection at the receiver side.

```
for i = 1:N
    [Utmp Stmp Vtmp] = svd(h_f(:,(i-1)*Mt+1:i*Mt));
    U=[U Utmp];
    V=[V Vtmp];
    S=[S Stmp];
end
```

Coding 5 <svd_decompose_channel.m>

4.2 Adaptive loading and the modulation

4.2.1 Adaptive loading background

As mentioned earlier, the code developed by Prateek and Andrew come with the ability of adaptive loading. For a conventional system, the number of bits and power allocation to every available subcarrier used a method called 'water-filling'. And since the data transmission may be varied from time to time depend on the application that we are using or the channel condition, therefore the different number of data bits may also occur in different transmission. By using 'water-filling' method, the different number of data bits and power will be allocated sequentially independently with the channel condition.

'Water-filling' may be a good technique if our transmission channel is perfect and free of interference, but in a real world condition, this is not true, because a practical wirelesss transmission channel may contains a lot of noise that cause interference to different carrier. In a OFDM system, data bits are modulated into different subcarrier for transmission, since OFDM divide the available bandwidth into smaller band, its advantage of having a narrowband and flat-fading can be assumed. Interference and some channel condition may affect some subcarrier to have a lower gain resulting a bad BER if data are carried in these subcarrier for transmission, while the other unaffected subcarrier having a better performance compare to those degraded subcarrier. Therefore, 'Water-filling' is not an optimal solution here, filling every data bits to every available subcarrier, causing the data reaching the destination to be corrupted, and need to be retransmitted, as a result increasing the BER.

Provided perfect channel knowledge is available to both the transmitter and receiver. This is where adaptive loading can be useful, taking advantage of the subcarrier with better performance, dynamically allocating different number of data bits and energy based on the different performance of different subcarrier, subcarrier with higher gain can be allocated with more data bits with the use of lower energy and vice versa for carrier with lower gain. Since our channel is time-varying, different frequency-selective channel contains a different decorrelation time, therefore the adaptation must be updated in every channel decorrelation. The use of adaptive loading is to achieve data rate and power optimisation, based on the knowledge of each subchannel gains, and this technique implemented here in the simulation is based on the algorithm developed by P. Chow et al, 1995, and J. Campello de Souza, 1999.

4.2.2 Adaptive Modulation and Demodulation

In digital communication, modulation and demodulation is done by transforming individual, or groups of data bits together into a complex number representation of an element in a particular signal constellation. Since our simulation can load data adaptively, Multiple Quadrature Amplitude Modulation (MQAM) is selected because of its ability to modulate differently depend on the number of bits to be modulated, therefore it is use for the process of modulation and demodulation throughout this simulation. During the adaptive loading part, only six level of MQAM signal constellation will be considered, or more specifically they are 0, 2, 4, 16, 64, 256 QAMs. An example is shown below in Figure 22 for a 16-QAM constellation diagram, showing also are the in-phase and quadrature-phase, which represent the real and imaginary part respectively. The use of this six level MQAM modulation scheme is expected to perform as efficient as the unrestricted constellation as discussed in A. Goldsmith, 2005.

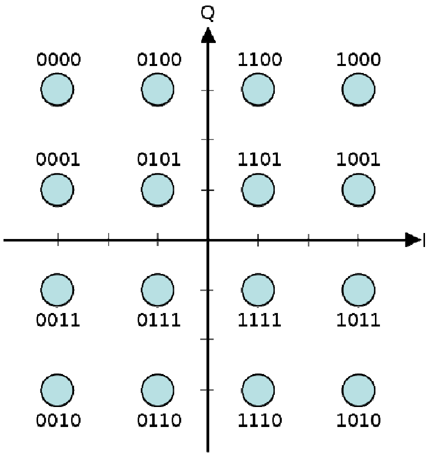


Figure 22 16-QAM constellation diagram

At the initial state of the simulation, all these MQAM data will be loaded for the use later in the process of modulation and demodulation, Coding 6 below shows the loading of the MQAM data to be used.

```
load ENC2.mat
load ENC4.mat
load ENC16.mat
load ENC64.mat
load ENC256.mat
```

Coding 6 <ofdm_main.m>

Coding 6 above just load the data MAT file of each constellation into the MATLAB workspace, Table 4 below outline all the data bits and complex constellation index number contain within each component in the MAT file above, with which for the ENC16, 64, and 256 showing only the first five components to reserve space. In adaptive modulation, the modulator takes the set of input bits and energy values, based on the number of bits, and then by make a reference to the constellation table, the modulator output the constellation symbol based on the number of bits from the input, and also scaled the energy appropriately for every subcarrier.

MAT file	Data bits codewords	Complex constellation index
ENC2.mat	0 1	-0.7071 0.7071

ENC4.mat	0 0
	0 1
	1 0
	1 1
		-0.7071 + 0.7071i
		0.7071 + 0.7071i
		-0.7071 - 0.7071i
		0.7071 - 0.7071i

ENC16.mat	0 0 0 0
	0 0 0 1
	0 0 1 0
	0 0 1 1
	0 1 0 0
	...
		-0.9487 + 0.9487i
		-0.3162 + 0.9487i
		0.9487 + 0.9487i
		0.3162 + 0.9487i
		-0.9487 + 0.3162i
		...

ENC64.mat	0 0 0 0 0 0
	0 0 0 0 0 1
	0 0 0 0 1 0
	0 0 0 0 1 1
	0 0 0 1 0 0
	...
		-1.0801 + 1.0801i
		-0.7715 + 1.0801i
		-0.1543 + 1.0801i
		-0.4629 + 1.0801i
		1.0801 + 1.0801i
		...

ENC256.mat	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 1
	0 0 0 0 0 0 1 0
	0 0 0 0 0 0 1 1
	0 0 0 0 0 1 0 0
	...
		-1.1504 + 1.1504i
		-0.9971 + 1.1504i
		-0.6903 + 1.1504i
		-0.8437 + 1.1504i
		-0.0767 + 1.1504i
		...

Table 4 MQAM constellation numerical representation

To increase the robustness at the receiver during demodulation in order to reduce the bit errors, Gray coded bit assignment constellation is also implemented. Gray coding make sure that an adjacent symbol can be selected even if a particular symbol contains error or corrupted at the decoder, this resulting only a single bit error instead of treating the whole symbol as an error where retransmission is required.

After the adaptive loading algorithm (which will be discussed later in the next subchapter) has assigned the number of data bits that each individual subcarrier can carry, the next thing need to do is to modulate these data and then prepare for transmission. The random data bits are generated with the *randn* function, and then normalising it to contains only binary digits 1 and 0 as shown in Coding 7 below.

```
x = (randn(1,Btot)>0);
```

Coding 7 <ofdm_main.m>

Coding 8 below show essentially the modulation on each subcarrier based on the number of bits assigned, using the in-build MATLAB *switch* function to select six different cases of modulation scheme as discussed before by assigning the complex constellation index for every subcarrier. Also the energy allocated to each subcarrier will be multiplied together with the complex constellation index during the modulation as determined before by the adaptive loading algorithm. Zero bits can also be allocated if a particular subcarrier is heavily interfered.

```
for i = 1:length(b)
    switch b(i)
    case {1}
        y = [y s2(x(b2(i))+1)*sqrt(e(i))];
    case {2}
        y = [y s4(x(b2(i):(b2(i+1)-1))*[2;1]+1)*sqrt(e(i))];
    case {4}
        y = [y s16(x(b2(i):(b2(i+1)-1))*[8;4;2;1]+1)*sqrt(e(i))];
    case {6}
        y = [y s64(x(b2(i):(b2(i+1)-1))*[32;16;8;4;2;1]+1)*sqrt(e(i))];
    case {8}
        y = [y s256(x(b2(i):(b2(i+1)-
1))*[128;64;32;16;8;4;2;1]+1)*sqrt(e(i))];
    otherwise
        y = [y 0];
    end
end
```

Coding 8 <modulate.m>

The demodulation process is also similar as the modulation process mentioned above, but it is done in a 'reverse' manner. As discussed before, since the perfect channel knowledge is assume to be known to both the transmitter and receiver, therefore the receiver also have the information for the bits and energy allocation, which will then ease our demodulation process by applying zero-forcing method as shown in Coding 9 below. After recovering the constellation index by finding the nearest distance between each estimated symbol, their corresponding binary data bits could also be recovered by comparing to the codeword index as shown in Table 4 above.

```
for i = 1:length(b)
    switch b(i)
    case {1}
        [tmp, index] = min(abs(s2-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c2(index,:)];
    case {2}
        [tmp, index] = min(abs(s4-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c4(index,:)];
    case {4}
        [tmp, index] = min(abs(s16-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c16(index,:)];
    case {6}
        [tmp, index] = min(abs(s64-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c64(index,:)];
   case {8}
        [tmp, index] = min(abs(s256-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c256(index,:)];
   otherwise
        index = 0;
   end
end
```

Coding 9 <demodulate.m>

After the demodulate process block, and since the transmitted data is recovered now, the data can then be sent to the next block for the wireless transmission performance calculation.

4.2.3 Adaptive loading

Our time varying fading channel statistically affecting the subcarrier of a OFDM signal, causing some subcarrier to have a better energy gain compare to the other, therefore a constant power water-filling allocation method could not be ideal as mentioned before. With perfect channel knowledge, adaptive loading can identify the performance different of these subcarriers, higher gain subcarrier can be allocated with more data bits while using less transmission energy, and vice versa to the lower gain subcarrier.

The adaptive bit loading is initialise based on the method suggested by Peter S. Chow, 1995. The algorithm procedure can be summarised step by step as follows (Prateek and Andrew, 2001):

1. Compute the subchannel SNR

   Or more typically, the simulation implemented here computes the GNR with which the SNR is divided by gap, where gap is a tuning parameter, different gap values can yields a different normalised SNR for every subchannel as shown in Coding 10 below. Where gap can affect the computation of the value of the energy table, also tuning the gap can be useful to characterised the BER performance of the system. The normalised SNR is the ratio of Energy per Bit (E_b) to the Spectral Noise Density (N_o), or more typically E_b/N_0, while the value gap is based on the approximation suggested G.D Forney Jr. et al., 1991.

```
SNR = abs((subcar_gains.^2)./(noise*gap));
```

Coding 10 <ComputeSNR.m>

2. Iterating through every i-th subcarrier to calculate the number of bits to be assigned on every i-th subcarrier according to the following formula:

$$b(i) = log_2(1 + GNR(i))$$

3. Round down the value $b(i)$ to $b(i)'$ by using the MATLAB in-build function $round$.
4. Restrict $b(i)'$ to the modulation MQAM order, which is the values 0, 1, 2, 4, 6, 8.
5. Calculate the energy for every i-th channel based on the bits assigned before in step 2, by using the following formula:

$$e_i\big(b(i)\big) = (2^{b(i)} - 1)/GNR(i)$$

6. Then generate a energy increment table for every subchannel. For every i-th channel,

$$\Delta e_i(b) = e_i(b) - e_i(b-1) = \frac{2^{b-1}}{GNR}$$

Step 2 to 6 above can be simulated as in Coding 11 below, the full coding can be found in the Appendix section.

```
for i = 1:num_subc
  tempbits         = log2(1 + abs(SNR(i)));
  roundtempbits    = round(tempbits);

  if (roundtempbits > M)
     roundtempbits = M;
  end

  if (mod(roundtempbits,2)== 1 & roundtempbits ~= 1)
      roundtempbits = roundtempbits -1;
  end

  if roundtempbits > 0
     energy_alloc(i)   = (2^roundtempbits-1)/SNR(i) ;
  else
     energy_alloc(i) = 0;
  end

  bits_alloc(i) = roundtempbits;
end
```

Coding 11 <chow_algo.m>

Then an energy table is constructed, it is used to store the energy required by every i-th subchannel transitioning from bits to bits, for example from 1 to 2 bits, from 2 to 3 bits, and etc, this is to make sure the correct minimum number of additional energy can be assigned on the next iteration to support, for example an additional bit. And since our MQAM modulation only support a maximum constellation of 8 bits for every subcarrier, transitioning from 8 to 9 bits will be set to a very large value. Also, since the allowable MQAM modulation order only support even number of 0, 1, 2, 4, 6, 8 bits (0, and 1 are special case), therefore odd number such as 3, 5, 7 are not supported, but a method is use to address this problem, that is by averaging the value of each adjacent even number, this process can be seen in Coding 12 below.

```
energytable = abs((1./SNR)'*(2.^([1:M+1]-1)));

energytable(:,M+1) = Inf*ones(size(energytable(:,M+1)));

for i = 3:2:M
    energytable(:,i) = (energytable(:,i) +energytable(:,i+1))/2;
    energytable(:,i+1) = energytable(:,i);
end
```

Coding 12 <EnergyTableInit.m>

The Chow's algorithm above initialise the loading process, and it need to be further optimise based on the increment of additional data bits or information to be carried by each subcarrier that uses the least incremental energy for its transmission. Given the initial bit allocation, and the energy incremental data, the B-tightening method suggested by J. Campello, 1999 can be used to optimised the bit allocation, the algorithm is summarised as follow (Prateek and Andrew, 2001):

```
1. $Input$:
$b, initial\ bit\ allocation$
$B, the\ total\ number\ of\ bits\ to\ be\ allocated$
2. $Output$:
$b, replace\ by\ the\ optimised\ bit\ allocation$
3. $Algorithm$
$B' \leftarrow 0$
$for\ n = 1\ to\ N$
    $B' \leftarrow B' + b(n)$
```

```
while $(B' \neq B)$
    if $(B' > B)$
        $n = arg\{max_{1 \leq j \leq N} \Delta e_j(b_j)\}$
        $B \leftarrow B - 1$
        $b(n) \leftarrow b(n) - 1$
    else
        $m = \arg\{min_{1 \leq i \leq N} \Delta e_i(b_i + 1)\}$
        $B \leftarrow B + 1$
        $b(n) \leftarrow b(n) + 1$
```

The above algorithm can be call as in Coding 13 below, the full coding is attached in the Appendix section.

```
[bits_alloc,energy_alloc] =
campello_algo(bits_alloc,energy_alloc,energytable,total_bits,num_subc,M);
```

Coding 13 <BitLoad.m>

The subcarrier is now fully optimised and characterised with the number of bits and energy allocation based on the frequency selective channel condition, as a result the performance of the wireless transmission can be increased.

4.3 OFDM's IFFT and FFT operation

As its name suggest, this is the OFDM operation, where IFFT is implemented in the transmitter to generate the equivalent time domain waveform for transmission, meanwhile FFT is implemented in the receiver to recover the frequency spectrum from the time varying signal. For the transmission process, the system can run the IFFT once the data have been modulated, but since the MIMO channel $\boldsymbol{H_i}$ is decomposed into its *svd* components $\boldsymbol{U_i S_i V_i}^*$ as discussed before, we can apply a transmit precoding filter of $\boldsymbol{V_i}$ just before the IFFT operation, the precoder cancels the right unitary matrix in SVD of the channel by left-multiplies the channel of each tone as in Coding 14 below.

```
x_pre =[];
for i = 1:N
    x_pre = [x_pre; [V(:,(i-1)*Mt+1:i*Mt)]*transpose(x_mod((i-
1)*Mt+1:i*Mt))];
end
x_pre = transpose(x_pre);
```

Coding 14 <precode.m>

Similarly a shaping filter of $\boldsymbol{U}_i^*$ is apply just after the FFT operation at the receiver, this will shapes the received symbol after FFT conversion but just before the demodulation by left multiplying it with the transpose of the received symbol, the channel will then be decomposed back into parallel subchannel as in Coding 15 below.

```
shaped_vals =[];
for i = 1:N
    shaped_vals = [shaped_vals; [U(:,(i-
1)*Mr+1:i*Mr)]'*transpose(rec_symbol((i-1)*Mr+1:i*Mr))];
end
shaped_vals = transpose(shaped_vals);
```

Coding 15 <shape.m>

The process of applying the unitary pre-filtering (precoding) and post-filtering (shaping) matrix to the transmitted and received data symbol in between the IFFT and FFT operation is to decompose the MIMO channel into a couple of conventional SISO channels, which also decompose the subcarriers into parallel form for the transmission process, as a result we can easily allocate the number of bits and energy required based on the parallel subcarrier SNR (J. H. Sung and J. R. Barry, 2001).

Next, the IFFT can be operated after the precoding process, by using the in-build MATLAB IFFT function and applying GI to the signal, and then it is ready to be transmitted as shown in Coding 16 below.

```
ofdm_symbol = ifft(inp_symbol,num_subc);
guard_symbol = ofdm_symbol(end-guard_interval+1:end);
ofdm_symbol = [guard_symbol ofdm_symbol];
```

Coding 16 <ifft_cp_tx_blk.m>

The signal generated by IFFT above is then modelled to be transmitted into the wireless transmission channel, as mentioned earlier, the MIMO channel estimation is formulated as $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, since $\boldsymbol{H}$ is know, which is the MIMO frequency selective fading channel, and $\boldsymbol{x}$ is the vector of input symbol, and it is a matrix in MIMO transmission. While $\boldsymbol{n}$ is the AWGN, which is a spatial noise and it is temporary white. The process of creating a noise vector by a given noise variance, and then applying all these known value together $\boldsymbol{H}$, $\boldsymbol{x}$, and $\boldsymbol{n}$, we can find the channel output which will be detected by the receiver as in the Coding 17 below.

```
noise = sqrt(sig2)*1/sqrt(2)*(randn(Mt*N,1) + j*randn(Mt*N,1));

y = H*x + noise;
```

Coding 17 <channel.m>

The receiver receives the channel vector output $\boldsymbol{y}$ (matrix in the case of MIMO), MATLAB in-build function FFT can be used to extract the frequency spectrum encoded within the time varying signal waveform to retrieve the transmitted modulated symbol, GI need to be remove before the FFT as shown in Coding 18 below. As mentioned earlier, the OFDM symbol output from FFT need to be shaped before demodulation process can begin to recover the original transmitted data on each subcarrier.

```
rec_time_symbol = ofdm_symbol(guard_interval+1:end);
rec_symbol = fft(rec_time_symbol,num_subc);
```

Coding 18 <fft_cp_rx_blk.m>

The error for each transmit iteration can now be calculated by using MATLAB in-build function *xor* to check the if the received data is correct by comparing to the original transmitted data as in Coding 19 below.

```
totalErrors = totalErrors + sum(xor(y_demod,x));
```

Coding 19 <ofdm_main.m>

The normalised SNR, E_b/N_0 can be calculated after each channel iteration by summing all energy allocated on each subcarrier to get the total energy allocated, and then dividing by the total number of transmitted data bits and noise variance. Lastly, the final BER data can be generated by dividing the total error on each channel iteration by the total number of data bits transmitted and channel and transmit iteration to get an average value as can be seen in Coding 20 below.

```
EbNo = [EbNo sum(energy_alloc)/Btot/sig2];

Errors = [Errors totalErrors/Btot/ChannelIter/TransmitIter]
TotEbNo = [TotEbNo mean(EbNo)]
```

Coding 20 <ofdm_main.m>

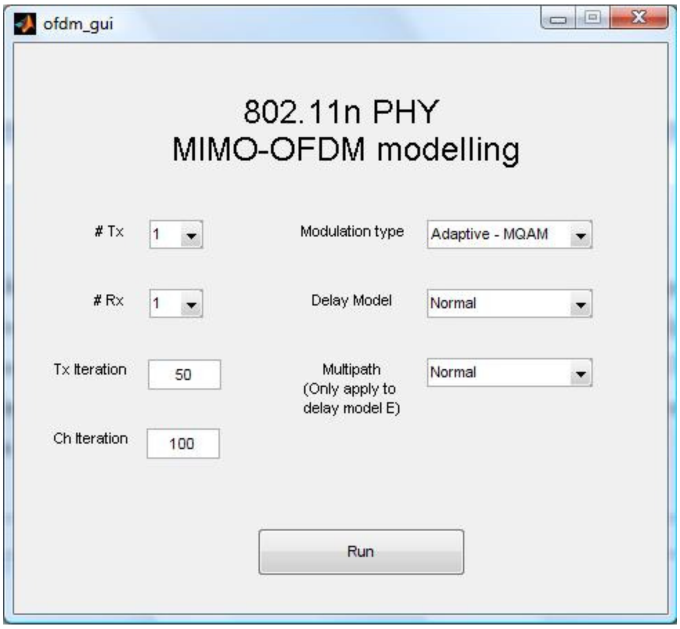
4.4 Data input and user interface

Input variable	Changeable conditions
Number of Transmit antenna	Selectable from 1 to 4
Number of Receive antenna	
Transmission iteration	User input
Channel iteration	
Modulation type	Adaptive or fixed rate
Delay model	Selectable from 4 different channel condition
Multipath effect	Increase multipath effect linearly

Table 5 Simulation changeable input variables

Some input data need to be feed initially to the simulation process before it can run and do its job, this include some variable that we would like to change from time to time before the simulation starts, in order to model the wireless transmission under different condition, for example number of transmit antenna, and etc, some changeable variable are summarised in Table 5 above.

A simple Graphical User Interface (GUI) has been designed for user to change the different input of various setting as mentioned in Table 5 above easily. User can also call and run the simulation straight away from the GUI after the desire input variable is set. The interface of the GUI is shown below in Figure 23.



Tx 1

Rx 1

Tx Iteration 50

Ch Iteration 100

Modulation type Adaptive - MQAM

Delay Model Normal

Multipath (Only apply to delay model E) Normal

Figure 23 The GUI of the simulation <main_gui.m>

The #Tx and #Rx are the selectable inputs, ranging from 1 to 4, to change the number of transmit and receive antenna respectively. Note, currently only the same number of antenna MIMO system at both the transmitter and receiver are supported. Tx iteration and Ch iteration are the Monte Carlo iteration, user can vary this independently, it can be increased in order to produce results with smoother curve, but the bigger the number, the more time it will take for the simulation to run. Next is the modulation type, users are able to select in between adaptive or fixed rate modulation that to be used for the simulation. The delay model is based on the suggestion by TGn in Vinko Erceg et al. 2004, it specified different WLAN models for different environment, in this simulation, user are able to change between channel B, D, E or standard, which will be discussed in next chapter. Lastly, is the multipath effect, more multipath can be added to the simulation for a specified channel model by incrementing in a fixed ratio.

4.5 Chapter Summary

This chapter give the detail for the simulation modelling, from the input to the overall process, which make a good start for the next chapter to analyse and discuss the results of the simulation based on the various input variable mentioned earlier.

Chapter 5 Results and Discussion

It is assumed that the entire simulation is a discrete time system, also as mentioned before in previous chapter, perfect channel response information is assumed to be available for both the transmitter and receiver, which generally simplifies our modelling. Throughout the simulation, some of the parameters listed below in Table 6 are held constant, unless otherwise stated. Other values such as number of transmit and receive antenna to test the different combination of a MIMO system will be discussed later in this chapter.

Parameters	Value
Number of subcarrier	64
OFDM symbol	64 symbol periods
Guard interval (GI)	16 symbol periods
Available MQAM modulation	0, 1, 2, 4, 6, 8
Power delay profile (general)	$1, {}^{1}/_{e^{1}}, {}^{1}/_{e^{2}}$
Noise variance	1×10^{-3}
Transmit Monte Carlo iteration	50
Channel Monte Carlo iteration	100

Table 6 Fixed Simulation Parameters

5.1 Adaptive data bits and energy allocation

As discussed before, this simulation can allocate number of data bits and energy required on each subcarrier dynamically, based on the transmission's channel response which can be illustrated as Figure 24 below.

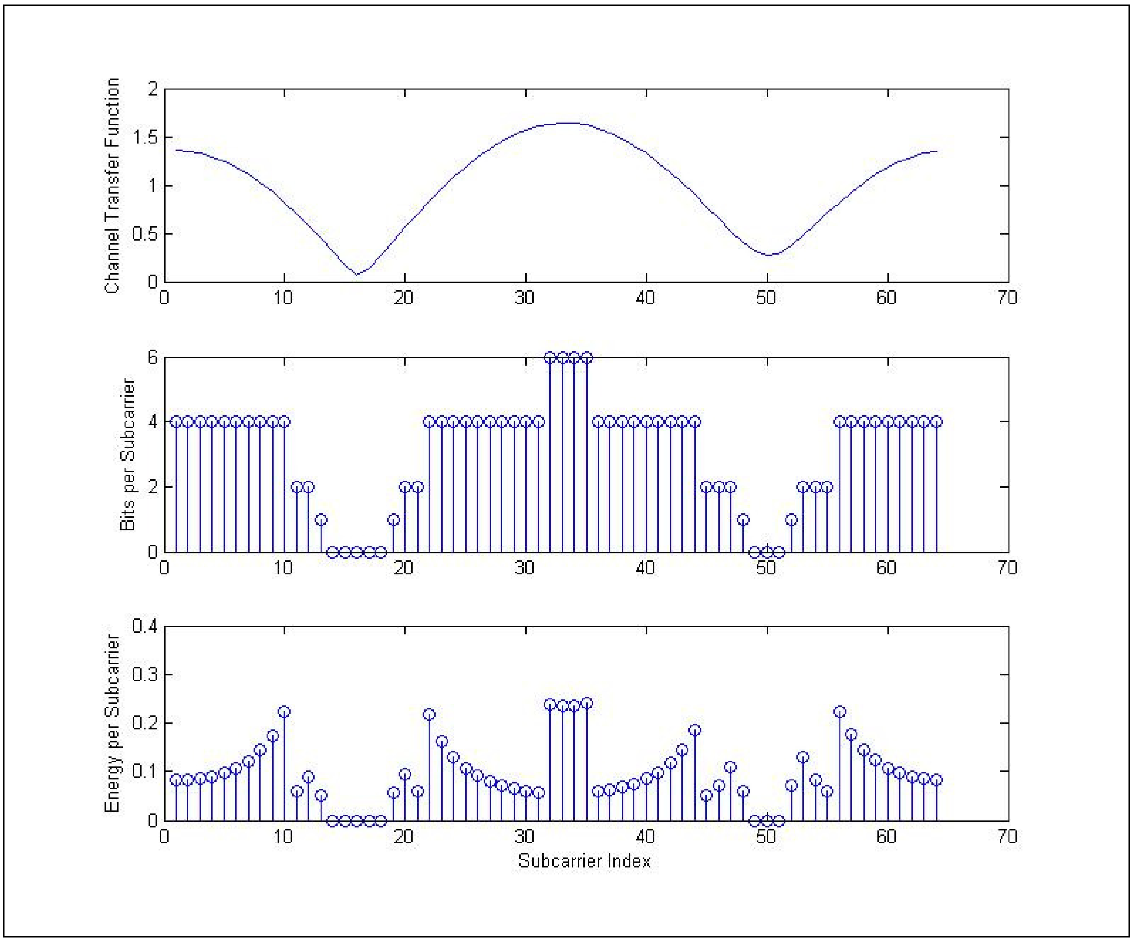


Figure 24 Adaptive Bits and Energy allocation on each subcarrier

Figure 24 above shows a transmit instance for a particular transmission. The first subplot above shows the channel transfer function of that instance with respect to the subcarrier index, we can observe that the channel response vary from carrier to carrier, therefore, by using adaptive allocation, data bits and energy will be allocated based on this channel condition in order to fully optimised the quality of wireless transmission, as expected, the subcarrier with very poor channel gain would be assigned with lesser bits or even zero bit allocated to each of them. For example, as we can see above, some worse channel condition happened in between subcarrier index 10 to 20 and also somewhere around subcarrier index 50. As a result, we can notice on the second subplot, which is the bits per subcarrier subplot, the subcarrier index 14 to 18 and 49 to 51 experiences no bits allocated, meanwhile subcarrier with higher channel gain can therefore use to regulate this situation by allocating more data bits on them. We can also notice that some subcarriers with higher gain are allocated with lesser energy compare to some of the lower gain subcarrier which needs more energy to transmit a particular number of data bits.

5.2 PHY performance

This simulation emphasise on modelling the performance of the PHY layer of the 802.11n, or more typically the OFDM and MIMO system. One of the most common techniques to analyse the performance of a telecommunication system is by computing the value of Bit Error Rate, or Bit Error Ratio (BER). BER evaluate the system performance by calculating the total number of error bits received during a transmission and comparing it to the total number of original transmitted data bits, a simple mathematics representation is shown below:

$$BER = \frac{Total\ Number\ of\ Errorneous\ Data\ Bits\ Received}{Total\ Number\ of\ Original\ Data\ Bits\ Transmitted}$$

BER is an ideal solution to be used for determining and analysing the performance of the wireless transmission and will be implemented here.

The smaller the BER, the better the performance, and since BER is evaluated as a ratio, so a value of $BER = 1$ is therefore the worst condition where all the transmitted data appears to be an error at the receiver. In order to achieve an infinity small BER, we need also an infinity large E_b/N_0, but this could never be achieved practically. As discussed before, E_b/N_0 is the normalised SNR, or more generally it is the measure of SNR for digital communication. E_b/N_0 is a measurement at the receiver of the amount of data which can be carried by the transmit signal depends on how the received signal strength exceeds the noise at the receiver, or more generally it is a measurement of the relative on the desired signal compare to the noise power (Gary M. Miller, 1999). E_b/N_0 is normally expressed in decibels (dB), the higher it is, more data can be carried on the signal and received successfully at the receiver.

First of all, a comparison has been done to verify some of the performance differences by varying the basic settings of the wireless transmission. Figure 25 below shows the comparison between 3 different conditions, which are a 2 by 2 MIMO link with adaptive modulation, a SISO link with adaptive modulation, and a SISO link with fixed OFDM by assigning a fix number of data bits on each subcarrier and optimised variable energy allocation. For a fair comparison, the total number of data bits on every OFDM symbol is set to a fixed constant throughout the simulation.

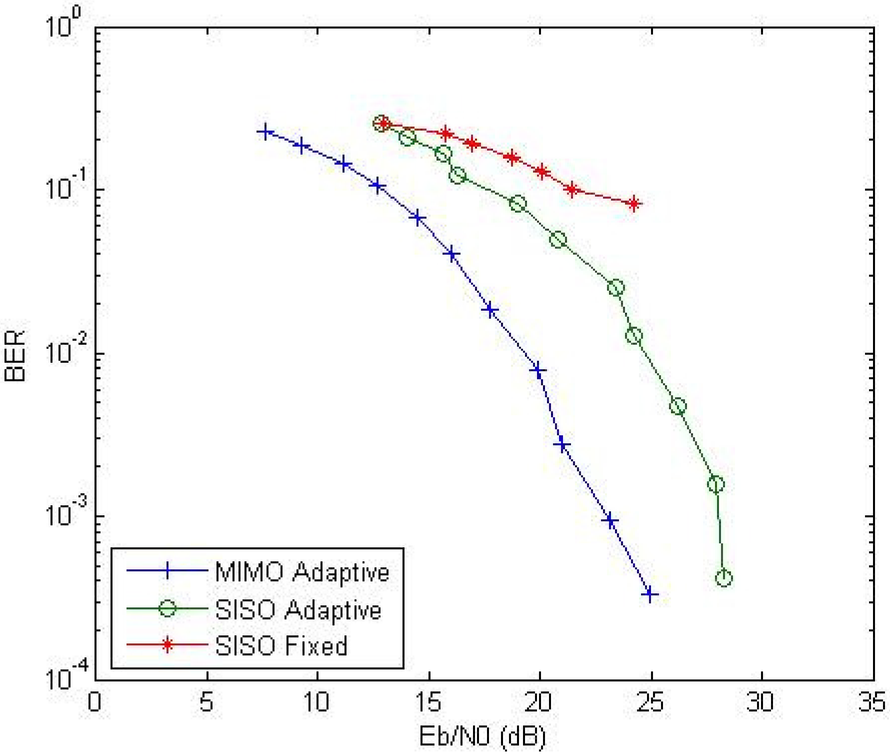


Figure 25 BER performance for different schemes

By observing the figure above, we can easily tell that the MIMO system has the best performance simply by comparing the same value of BER, the MIMO system only required a very small value of E_b/N_0, while for the SISO system, they required a higher value of E_b/N_0 in order to achieve the same BER value like the MIMO system. The SISO fixed-rate scheme performs the worse among 3 of them, in order to achieve a BER value of around 10^{-1}, it require a E_b/N_0 value of around 25 dB, which for the MIMO system, it can achieve the BER of more than 10^{-3} for the same E_b/N_0. Also, the last few sample of the SISO fixed-rate scheme has been neglected because the value varied by a large margin that generally do not shows any important information here, that is mainly because it could not exceed anymore for a higher E_b/N_0 where the SISO fixed-rate system is not capable with. By comparing both the SISO systems, it shows that the SISO adaptive system outperformed the SISO fixed-rate system, the figure shows a 2 to 3 dB improve in E_b/N_0 in order to achieve the same BER performance, where this is always welcome because if implementing a hardware upgrade is resource consuming.

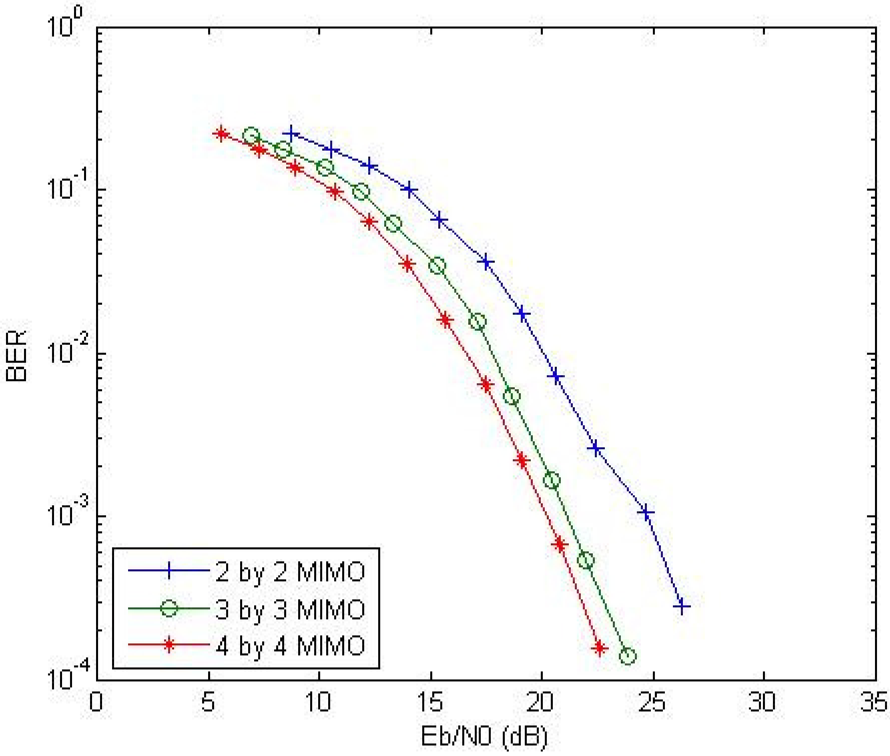


Figure 26 BER performance for different MIMO antenna combination

Next in Figure 26 above, we analyse the MIMO system with different number of antenna combination. Only the same number of antenna at both transmitter and receiver are being compared here. Figure 26 above shows that the 4 by 4 MIMO system has the best performance, following with 3 by 3 then 2 by 2 MIMO system. As discussed before, the more antenna combination of a MIMO system can increase the array gain, in another word, array gain is the increment of SNR in the receiver through spatial processing by coherently combining the transmit antenna array. Every bits of SNR improvement means more data bits can be allocated for successful transmission, once a minimum SNR is achieved to exchange information between transmitter and receiver at a desired rate, any additional SNR can increase the effective information data rate, Figure 27 below shows how number of antenna is related to the SNR, as well as the proportional increment of data rate with SNR. Array gain also improves the resistance to noise of the MIMO transmission, and hence increasing the coverage area of the wireless network (Ezio Biglieri et al., 2007).

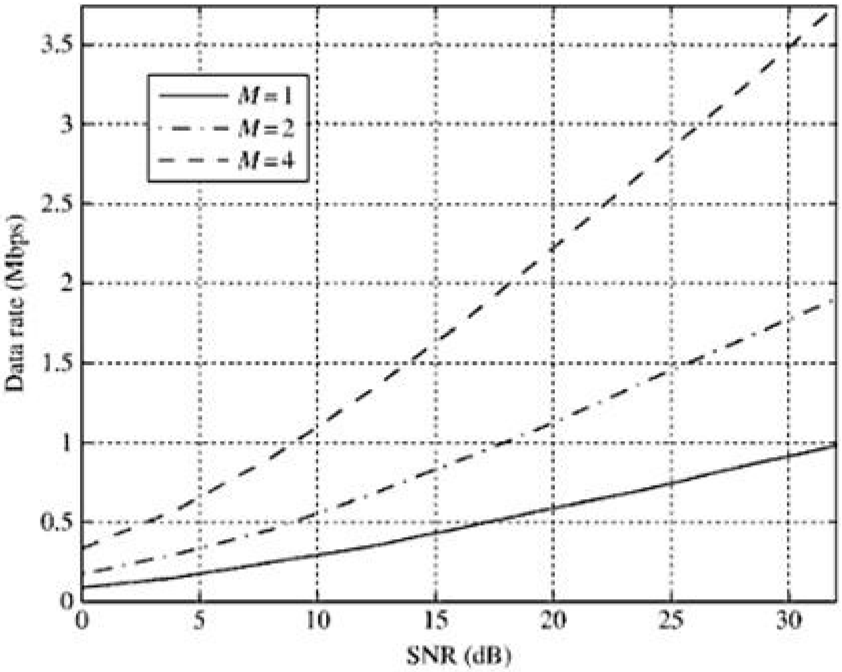


Figure 27 Data rate vs SNR with different antenna combinations

As an experimental purpose, more MIMO antenna combination has been simulated and the results are shown below in Figure 28, note the significant improvement in performance when increasing the combination of antenna number. An obvious different in performance can be seen by increasing a 2 by 2 MIMO system to a 3 by 3 MIMO system, it gives an approximately 3 dB improvement in order to achieve a same BER value. In order to better compare the various amounts of MIMO antenna configurations, combinations such as 5 by 5, 7 by 7, and 10 by 10 has been included, and we can see that the benefits in performance enhancement, but note that the gain improvement is only modest, which is around 1 dB better to each other, for example from 4 to 5, 5 to 7 and 7 to 10. Of course the more antenna the better the performance, but this may not be practically feasible, and the draft version of the 802.11n amendment also suggested that a maximum number of 4 by 4 combination which performs relatively well.

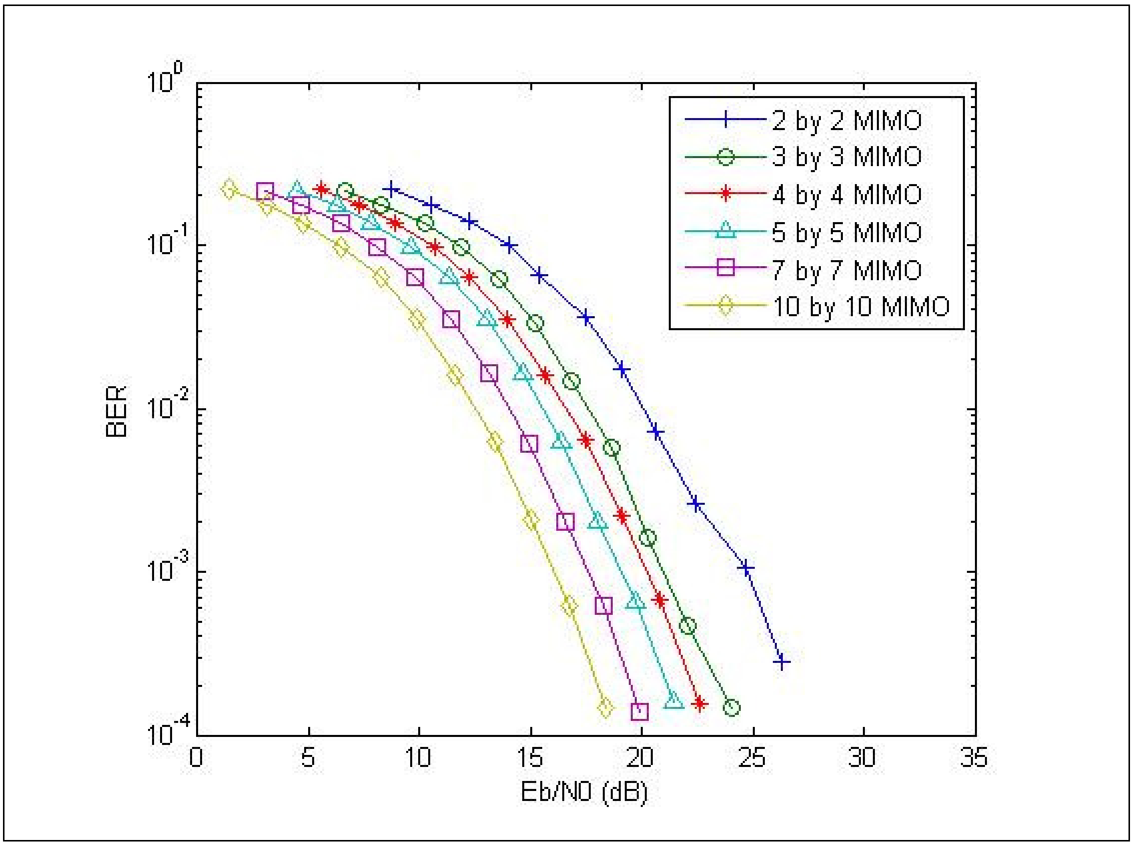


Figure 28 Various MIMO scheme with more antenna combinations

The IEEE 802.11n workgroup TGn proposed a few channel models and path loss with different power delay profile, they were defined for different environments, and due to the general purpose implementation of WLAN, these channel models were mostly developed based on indoor measurement. Here we will examined some of the different using these channel models, during the simulation, a self-defined and only TGn channel model B, D, E, with 3 power taps delay from each of them were considered, Table 7 below summaries and differentiate some of the characteristic of these channel models, and giving some typical examples of the environment with the respective channel model (Eldad and Robert, 2008).

Model	Environment	Delay Profile
Self-defined	General purpose For comparison	$\frac{1}{e^{0}}, \quad \frac{1}{e^{1}}, \quad \frac{1}{e^{2}}$
B	Residential e.g. intra-room, room-to-room	$\frac{1}{e^{0}}, \quad \frac{1}{e^{1.25}}, \quad \frac{1}{e^{2.5}}$
D	Typical office e.g. sea of cubes, large conference room	$\frac{1}{e^{0}}, \quad \frac{1}{e^{1.4}}, \quad \frac{1}{e^{4.45}}$
E	Large office e.g. multi-story office, campus small hotspot	$\frac{1}{e^{0.6}}, \quad \frac{1}{e^{1.58}}, \quad \frac{1}{e^{5.2}}$

Table 7 Channel Models and the delay profile

A 2 by 2 MIMO system with adaptive modulation were simulated here by applying different channel power delay profile as given in Table 7 above. The result can be seen in Figure 29 below, for the first 3 channel models (the self-defined, B, and D model) for a small environment, the results were very closed to each other, the difference is very small, this may due to the small instances of delay spread which did not varied too much in a typically small environment, and their BER performance also very close to each other. Meanwhile, we can also notice that in the channel model E for a large environment, it has a significant performance drop due to a large delay spread of the signal.

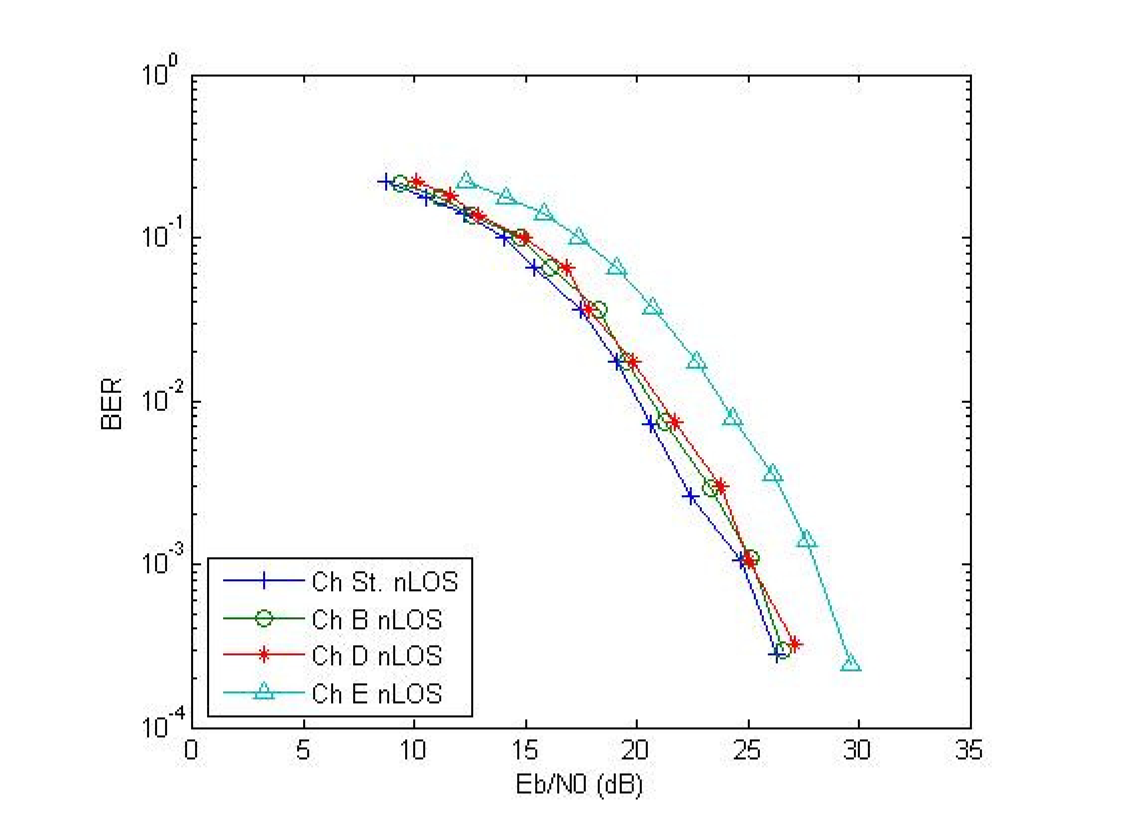


Figure 29 Comparison between different channel model

As discussed before, MIMO will take the multipath effect as a benefit to increase the transmission performance instead of destructive construction like a SISO system. A rich scattering environment with more resolvable multipath in an indoor environment increase the achievable MIMO capacity at a particular SNR value, and multipath fading effect tends to produce correlated channels, which increase the MIMO capacity linearly (Zhongwei and Ananda, 2006). In this investigation, a 2 by 2 MIMO antenna configuration is kept constant, running at the channel model E described before, and by applying an increase number of channel taps to stimulate the rich multipath effects, and the results can be illustrated in Figure 30 below. '+0' is the normal 3 taps original configuration, meanwhile each increment in number brings additional 3 channel taps to increase the multipath effect. As we can see below, the results shows increased performance for a MIMO system with increase multipath effects.

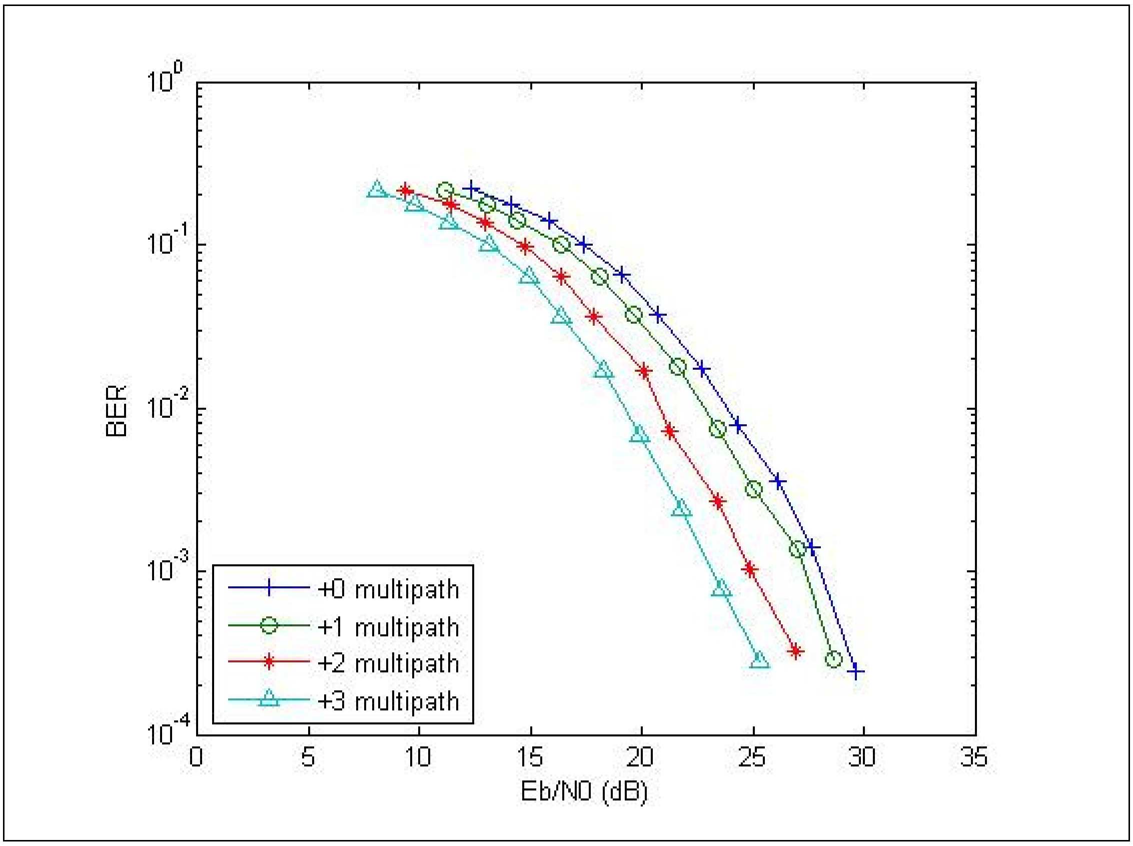


Figure 30 Multipath effects

5.3 Chapter summary

From the results above, first, we know how the adaptive modulation increase the performance by allocating data bits and energy dynamically depends on channel condition, and we also basically convinced that MIMO really plays an important role in order to dramatically improve wireless transmission, as well as increasing the number of effective antenna, constructively utilise the multiple delay signal spread to produce a better SNR that reduce the BER value, which can greatly increase the overall throughput and reliability for the wireless transmission.

Chapter 6 Conclusions

This dissertation aims to investigate the new family member of IEEE 802.11, which is the IEEE 802.11n amendment that brings numerous enhancements to the original IEEE 802.11 standard. The purpose of this new amendment is to address some of the problems that were still suffering in the previous generations. The changes in IEEE 802.11n are mostly focused on increasing the overall throughput and expanding the range of coverage, some of the important techniques implemented in the IEEE 802.11n are summaries below:

- In the Physical (PHY) layer
  Orthogonal Frequency Division Multiplexing (OFDM) is the key feature in high throughput communication system, with added Multiple-Input Multiple-Output (MIMO) technique to further increase the transmission performance, for example with the use of Spatial-Division Multiplexing (SDM), Space-Time Block Coding (STBC), Transmit Beamforming, and etc., together with optional channel bonding operation (40 MHz mode).
- In the Media Access Control (MAC) layer
  Aggregating MAC frame, for example with the use of MAC Service Data Units (MSDU) Aggregation, and the MAC Protocol Data Units (MPDU) Aggregation. These techniques improve the performance of wireless transmission for example, better Quality of Service (QoS), which is an important feature in real time communication, where delay during communication cannot be compromised, such as Voice over Internet Protocol (VoIP) or online internet gaming.

An elaboration of discussion were focused on the PHY layer, which include the description of operation of OFDM, how data symbols can be transmitted by Inverse Fast Fourier Transform (IFFT) and recovered by Fast Fourier Transform (FFT), and the advantages of using OFDM such as the ability to combat with Inter Symbol Interference (ISI) with the use of added Guard Interval (GI) symbol, and etc. Further on, MIMO, the key factor in higher throughput were discussed, some features were describe, for example the matrix channel modelling, combine signal strength constructively with the use of multipath effect, and etc. Also, some brief discussion of the MAC aggregation, for example the Ethernet packet based MSDU and 802.11 packet based MPDU.

Later, the simulation platform was described, the procedure of the simulation was discussed, with detail description on some extracted coding. Adaptive modulation was also introduced with its ability to allocate data bits and energy dynamically, which increased the performance by utilising the subcarrier with better channel gain, also MQAM modulation was chosen as the digital modulation scheme since adaptive modulation was implemented. Then MIMO modelling was describe, with the use of a set of channel tap based on different channel model defined by TGn, as well as the nLOS Rayleigh frequency selective fading channel.

Finally, the BER versus E_b/N_0 performance was used to illustrate the advantages of using MIMO, together with some additional modelling like testing the effect of different channel, as well as the multipath effect, the obtained results further increase our confident in MIMO as an important modifications to the PHY layer in high throughput wireless communications such as the new IEEE 802.11n high performance WLAN system.

Some further work can be done for example, modifying or improving the simulation more sophisticatedly into the IEEE 802.11n specifications, also the modelling can be extended to simulate other improved features or functions in 802.11n such as the MAC aggregation, channel boding, and etc., since they also play an important roles in improving the 802.11n WLAN system. Comparison between 802.11n with other available wireless communication can be made if possible to determine the co-existence issues such as the similarity and the applications. For example a new implementation of 802.11 for peer-to-peer low range high speed communication, or also known as IEEE 802.11z specification, or Wi-Fi™ Direct, this could be an interesting addition in functionality to 802.11 which normally require a station in between devices, and since most of the mobile devices nowadays have 802.11 WLAN adapter build-in, therefore it can be studied on how this new functionality can co-exist with the similar wireless technology such as Bluetooth.

References

Agilent Technologies, Inc, “MIMO Wireless LAN PHY Layer [RF] Operation & Measurement,” Application Note 1509, 2008.

Andrea Goldsmith, “Wireless Communications,” Cambridge University Press, 2005.

Atheros communications, “Getting the Most out of MIMO: Boosting Wireless LAN Performance with Full Compatibility,” White paper, 2005.

Broadcom Corporation, “802.11n: Next-Generation Wireless LAN Technology,” White paper, 2006.

Cisco System, Inc., “802.11n: The Next Generation of Wireless Performance,” Cisco Public Information, White paper, 2007.

Cisco Systems, Inc., “Channel Deployment Issues for 2.4-GHz 802.11 WLANs,” 2004.

Eldad Perahia and Robert Stacey, “Next Generation Wireless LANs Throughput, Robustness, and Reliability in 802.11n,” Cambridge University Press, The Edinburgh Building, Cambridge, UK, 2008.

Enis Akay, Ersin Sengul, and Ender Ayanoglu, “Performance Analysis of Beamforming for MIMO OFDM with BICM,” IEEE International Conference on Communications, 2005. ICC 2005. Volume 1, pp. 613-617, 16-20 May 2005.

Eric Lawrey, “The suitability of OFDM as a modulation technique for wireless telecommunications, with a CDMA comparison,” Thesis, 1997.

Ezio Biglieri et al., “MIMO Wireless Communications,” Cambridge University Press, The Edinburgh Building, Cambridge, UK, 2007.

F. Sarabchi and M. E. Kalantari, "On the Use of Adaptive Loading algorithms in MIMO-OFDM Systems."

G. D. Forney Jr. And M. V. Eyuboglu. Combined Equalization and Coding Using Precoding. IEEE Communications Magazine, 29(12):25-34, December, 1991.

Gary J. Mullett, "Wireless Telecommunications Systems and Networks, Thomson Delmer Learning, Clifton Park, NY, 2006.

Henrik Schulze and Christian Luders, " Theory and Applications of OFDM and CDMA for Wideband Wireless Communications," John Wiley and Sons, Ltd, West Sussex, England, 2005.

IEEE 802.11-03/940r2: TGn Channel Models. IEEE.

J. Barry, E. Lee, and D. Messerschmitt, "Digital Communications," 3rd ed. New York: Springer, 2004.

J. Campello: "Practical bit loading for DMT," IEEE International Conference on Communications ICC99, pp. 801-805, 1999.

J. H. Sung and J. R. Barry, "Space-Time Processing with Channel Knowledge at the Transmitter," European Conference on Communications, Bratislava, vol. 1, pp. 26-29, July 5-7, 2001.

J. Terry and J. Heiskala, "OFDM Wireless LANs: A Theoretical and Practical Guide," Indianapolis: Sam, 2002.

Karkhanechi, H. M., and Levy, B. C., " Spatial multiplexing and diversity gain in OFDM-based MIMO systems," IEEE Topical Conference on Wireless Communication Technology, 2003.

Kaveh Pahlavan, Prashant Krishnamurthy, "Principle of Wireless Networks: A unified Approach, Prentice Hall, Upper Saddle River, New Jersey, 2002.

Louie, R. H. Y, Yonghui Li, and Vucetic, B., “Performance Analysis of Beamforming in Two Hop Amplify and Forward Relays Networks,” IEEE International Conference on Communications, 2008, ICC ’08, pp. 19-23, 2008.

Mischa Schwartz, “Mobile Wireless Communications,” Cambridge University Press, The Edinburgh Building, Cambridge, UK, 2005.

Ozcelik. M., Czink. N., and Bonek. E., “What makes a good MIMO channel model?” IEEE 61st Vehicular Technology Conference, 2005. VTC 2005-Spring. Volume 1, pp. 156-160, 30 May-1 June 2005.

P. Chow et al, “A Practical Discrete Multitone Transceiver Loading Algorithm for Data Transmission over Spectrally Shaped Channels,” IEEE Trans. Comm, Vol. 43, No.2, February 1995, Page 773-775.

Prateek Bansal and Andrew Brzezinski, “Adaptive Loading in MIMO / OFDM Systems,” Stanford University, 2001.

RF signal processing, “The principles of OFDM,” 2001. Available: http://www.rfdesign.com

S. Haykin, “Adaptive Filter Theory,” Englewood Cliffs, NJ: Prentice Hall, 1996.

S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” IEEE Journal on Select Areas in Communications, Vol. 16, October 1998.

Steve Rackley, “Wireless Networking Technology From Principles To Successful Implementation, Newnes, Jordan Hill, Oxford, 2007.

Tao Cui, and Chintha Tellambura, “Power Delay Profile and Noise Variance Estimation for OFDM,” IEEE Communications Letters, Vol. 10, No. 1, January 2006.

TGn Sync Proposal Technical Specification document: IEEE 802.11-04/0889r7.

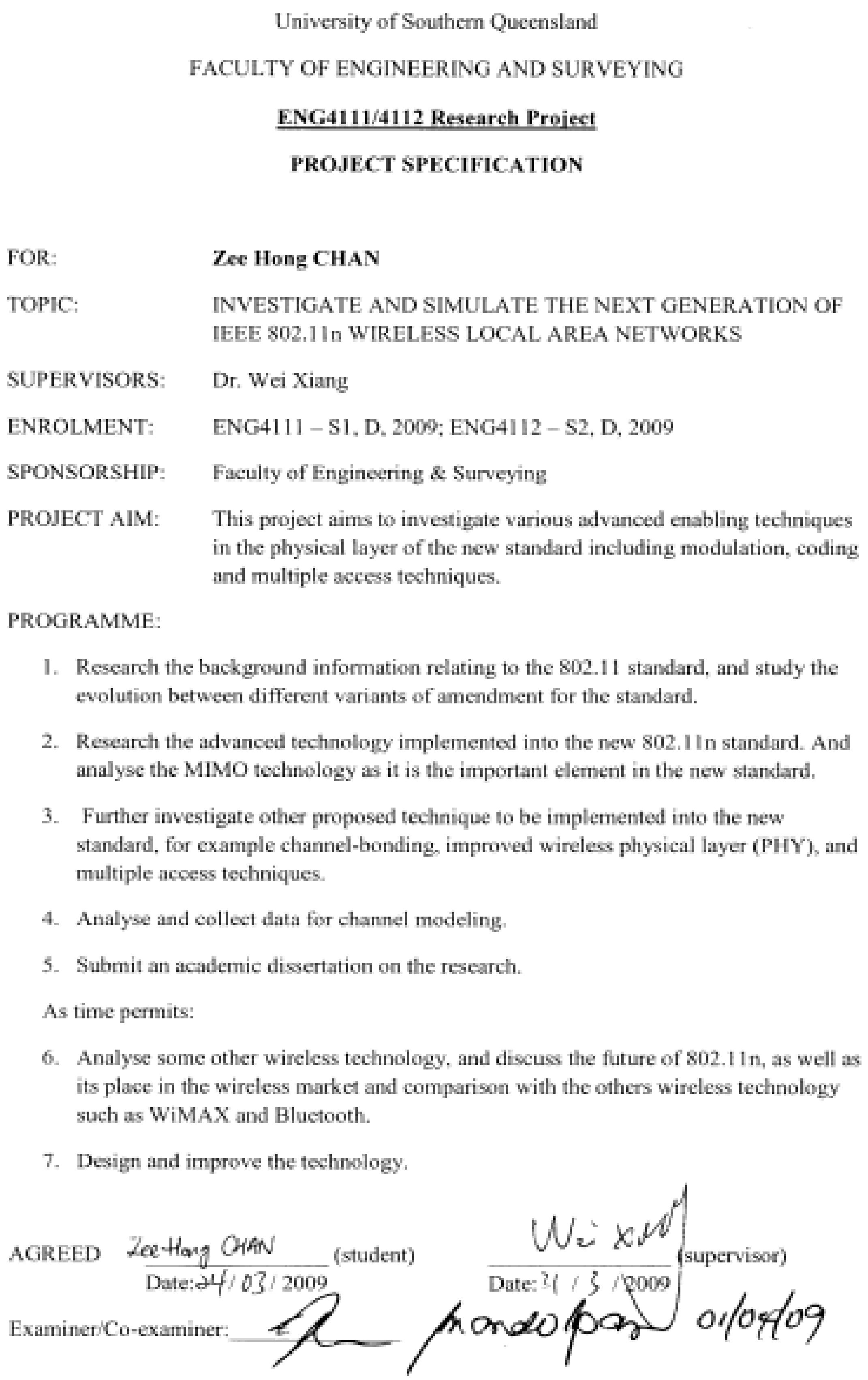
Thomas Paul and Tokunbo Ogunfunmi, "Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment," Circuits and System Magazine, IEEE, First Quater 2008, 28-54.

Vinko Erceg et al., "IEEE P802.11 Wireless LANs TGn Channel Models" IEEE 802.11-03/940r4, 2004.

Wi-Fi CERTIFIED™ 802.11n draft 2.0: Longer-Range, Faster-Throughput, Multimedia-Grade Wi-Fi® Networks, Wi-Fi Alliance, 2007.

Zhongwei Tang and Ananda S. Mohan, "Indoor Multipath Characterization for MIMO Wireless Communications," University of Technology, Sydney.

Appendix A – Project Specification



University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Research Project

PROJECT SPECIFICATION

FOR: **Zee Hong CHAN**

TOPIC: INVESTIGATE AND SIMULATE THE NEXT GENERATION OF IEEE 802.11n WIRELESS LOCAL AREA NETWORKS

SUPERVISORS: Dr. Wei Xiang

ENROLMENT: ENG4111 – S1, D, 2009; ENG4112 – S2, D, 2009

SPONSORSHIP: Faculty of Engineering & Surveying

PROJECT AIM: This project aims to investigate various advanced enabling techniques in the physical layer of the new standard including modulation, coding and multiple access techniques.

PROGRAMME:

1. Research the background information relating to the 802.11 standard, and study the evolution between different variants of amendment for the standard.
2. Research the advanced technology implemented into the new 802.11n standard. And analyse the MIMO technology as it is the important element in the new standard.
3. Further investigate other proposed technique to be implemented into the new standard, for example channel-bonding, improved wireless physical layer (PHY), and multiple access techniques.
4. Analyse and collect data for channel modeling.
5. Submit an academic dissertation on the research.

As time permits:

6. Analyse some other wireless technology, and discuss the future of 802.11n, as well as its place in the wireless market and comparison with the others wireless technology such as WiMAX and Bluetooth.
7. Design and improve the technology.

AGREED Zee-Hong CHAN (student) (supervisor)

Date: 24/03/2009 Date: 31 / 3 / 2009

Examiner/Co-examiner: 01/04/09

Appendix B – MATLAB coding

<main_gui.m>

```
% Name of function file main_gui.m
% Research Project Matlab Script for 802.11n PHY simulation
% SPECIAL NOTE: this is based on MATLAB code that were previously developed
% by Prateek Bansal and Andrew Brzezinski in 2001.
% This function is the main function to be run to call the GUI.
% then simulation can be run by varying different settings in the GUI.

function varargout = main_gui(varargin)
% MAIN_GUI M-file for main_gui.fig
%      MAIN_GUI, by itself, creates a new MAIN_GUI or raises the existing
%      singleton*.
%
%      H = MAIN_GUI returns the handle to a new MAIN_GUI or the handle to
%      the existing singleton*.
%
%      MAIN_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MAIN_GUI.M with the given input arguments.
%
%      MAIN_GUI('Property','Value',...) creates a new MAIN_GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before ofdm_gui_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to main_gui_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help main_gui

% Last Modified by GUIDE v2.5 25-Oct-2009 08:25:35

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_gui_OpeningFcn, ...
                   'gui_OutputFcn',  @main_gui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before main_gui is made visible.
function main_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main_gui (see VARARGIN)

% Choose default command line output for main_gui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main_gui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = main_gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

Mt = get(handles.txpopup,'Value');
Mr = get(handles.rxpopup,'Value');
Adapt = get(handles.modulatepopup,'Value');
Model = get(handles.modelpopup,'Value');
Multi = get(handles.multipathpopup,'Value');
Titer = str2num(get(handles.Tedit,'String'));
Citer = str2num(get(handles.Cedit,'String'));

[Errors, TotEbNo]=ofdm_main(Mt,Mr,Adapt,Model,Multi,Titer,Citer);

% --- Executes on selection change in txpopup.
function txpopup_Callback(hObject, eventdata, handles)
% hObject    handle to txpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns txpopup contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from txpopup

input = get(hObject,'value');

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function txpopup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in rxpopup.
function rxpopup_Callback(hObject, eventdata, handles)
% hObject    handle to rxpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns rxpopup contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from rxpopup

input = get(hObject,'Value');

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function rxpopup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rxpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in modulatepopup.
function modulatepopup_Callback(hObject, eventdata, handles)
% hObject    handle to modulatepopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns modulatepopup contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from
modulatepopup

input = get(hObject, 'Value');

guidata(hObject, handles);


% --- Executes during object creation, after setting all properties.
function modulatepopup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to modulatepopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in modelpopup.
function modelpopup_Callback(hObject, eventdata, handles)
% hObject    handle to modelpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns modelpopup contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from
modelpopup

tempM = get(handles.modelpopup,'Value');
tempP = get(handles.multipathpopup,'Value');

if tempM ~= 4
    if tempP ~= 1
        msgbox('Multipath only apply for delay model E');
    end
end

input = get(hObject, 'Value');

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function modelpopup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to modelpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in multipathpopup.
function multipathpopup_Callback(hObject, eventdata, handles)
% hObject    handle to multipathpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns multipathpopup contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from
multipathpopup

tempM = get(handles.modelpopup,'Value');
tempP = get(handles.multipathpopup,'Value');

if tempM ~= 4
    if tempP ~= 1
        msgbox('Change this for delay model E only');
    end
end

input = get(hObject, 'Value');

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function multipathpopup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to multipathpopup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end



function Tedit_Callback(hObject, eventdata, handles)
% hObject    handle to Tedit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Tedit as text
%        str2double(get(hObject,'String')) returns contents of Tedit as a
double


% --- Executes during object creation, after setting all properties.
function Tedit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Tedit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Cedit_Callback(hObject, eventdata, handles)
% hObject    handle to Cedit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Cedit as text
%        str2double(get(hObject,'String')) returns contents of Cedit as a
double


% --- Executes during object creation, after setting all properties.
function Cedit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Cedit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

<ofdm_main.m>

```
function [Errors, TotEbNo] = ofdm_main(Mt,Mr,Adapt,Model,Multi,Titer,Citer)
% function [Errors, TotEbNo] = ofdm_main(Mt,Mr,Adapt,Model,Multi,Titer,Citer)
%
% this is the main function that call other function during the simulation
% this function get the data input from the GUI and output the results by
% drawing a BER plot versus EbNo

switch Model
```

```
    case 1
        A = [1 1/exp(1) 1/exp(2)] % channel 'normal'
    case 2
        A = [1 1/exp(1.25) 1/exp(2.5)] % channel B
    case 3
        A = [1 1/exp(1.4) 1/exp(4.45)] % channel D
    case 4
        A = [1./exp(0.6:0.1:0.9), 1/exp(1.03), 1./exp(1.28:0.3:1.88), ...
        1/exp(2.25), 1/exp(2.69), 1./exp(3.2:0.5:5.2)]; % channel E
        switch Multi
            case 1
                A = A(1:7:15)
            case 2
                A = [A(1:5:15),A(15)]
            case 3
                A = [A(1:3:15),A(15)]
            case 4
                A = A(1:2:15)
        end
end
N = 64;  % number of symbols in a single OFDM symbol
GI = 16;  % guard interval
sig2 = 1e-3;  % noise variance
M = 8;  % max constellation bit number
Btot = 128*Mt;  % total # bits per OFDM symbol
Mgap = 10.^(1:(1.7/10):2.7);  % gap
if Adapt == 2
    Mgap = Mgap(1:7);
end
TransmitIter = Titer % # iterations of symbol transmissions for each channel
instance
ChannelIter = Citer % # iterations of independent identically distributed
channel instances
GapIter = length(Mgap);

load ENC2.mat
load ENC4.mat
load ENC16.mat
load ENC64.mat
load ENC256.mat

TotEbNo = [];
Errors =[];
EbNo = [];
for lGap = 1:GapIter
    lGap
    gap = Mgap(lGap);
    totalErrors = 0;

    for lChan = 1:ChannelIter

        % create channel
        [H h_f]=create_channel(Mt, Mr, A, N+GI);

        % decompose each subchannel in the frequency domain
        [U S V] = svd_decompose_channel(Mt, Mr, h_f, N);

        % bitloading
        [bits_alloc,energy_alloc] = BitLoad(S,Btot,Mt*N,gap,sig2,M);
        if Adapt == 2
            bits_alloc = 2*ones(size(bits_alloc));
        end


        for lTrans = 1:TransmitIter

            % bits to transmit
            x = (randn(1,Btot)>0);
            % modulate
            x_mod = modulate(x,bits_alloc,energy_alloc, s2,s4,s16,s64,s256);

            % precode modulated signal
            x_pre = precode(Mt, x_mod, V, N);

            % ifft, with cyclic prefix for each antenna
            ofdm_symbol =[];
            for i=1:Mt
                ofdm_symbol = [ofdm_symbol; ifft_cp_tx_blk(x_pre(i:Mt:Mt*(N-
1)+i),N,GI)];
            end

            ofdm_symbol2 = reshape(ofdm_symbol,Mt*(N+GI),1);

            % channel
            y = transpose(channel(sig2, Mt, Mr, ofdm_symbol2, H, N+GI));

            % fft
            rec_symbol =[];
            for i=1:Mt
                rec_symbol = [rec_symbol; fft_cp_rx_blk(y(i:Mt:Mt*(N+GI-
1)+i),N,GI)];
            end

            rec_symbol2 = reshape(rec_symbol,1,Mt*N);

            % shape received signal
            shaped_vals = shape(rec_symbol2, Mr, U, N);

            % demodulate
            y_demod = demodulate(shaped_vals, bits_alloc, energy_alloc, S,
s2,s4,s16,s64,s256, c2,c4,c16,c64,c256);

            % comparison
            totalErrors = totalErrors + sum(xor(y_demod,x));
        end
        EbNo = [EbNo sum(energy_alloc)/Btot/sig2];

    end

    Errors = [Errors totalErrors/Btot/ChannelIter/TransmitIter]
    TotEbNo = [TotEbNo mean(EbNo)]
    EbNo = [];
end

TotEbNo1 = 10*log10(TotEbNo);

figure(2)
semilogy(TotEbNo1, Errors, '+-');
xlabel('Eb/No');
ylabel('BER');
set(gca, 'xlim', [0 35]);

% save ch_3E_mm3_2x2_adapt_128_new Errors TotEbNo1
```

<create_channel.m>

```
function [H, H_f]=create_channel(Mt, Mr, A, N);
% function [H, H_f]=create_channel(Mt, Mr, A, N);
%
%   A - vector containing the power-delay profile (real values)
%   Mt - number of Tx antennas
%   Mr - number of Rx antennas
%   N - number of vector symbols to be sent in a single OFDM symbol Tx
%       ie: N MIMO transmissions in one OFDM symbol

% This is for Rayleigh frequency-selective fading, which assumes complex
% Gaussian matrix elements with in-phase and quadrature components
independent.
% Assume iid matrix channel elements, and further, independent channel taps

% define the channel taps
H_int = 1/sqrt(2)*(randn(Mr*length(A),Mt) + j*randn(Mr*length(A),Mt));

H_int2=[];
for i = 1:length(A)
    H_int2 = [H_int2;sqrt(A(i))*H_int((i-1)*Mr+1:i*Mr,:)];
end

H_int2 = [H_int2;zeros((N-length(A))*Mr,Mt)];

H_f = zeros(Mr,Mt*(N-16));
for i = 1:Mt
    for jj = 1:Mr
        h_f = fft(H_int2(jj:Mr:(N-16-1)*Mr+jj,i));
        for k = 1:(N-16)
            H_f(jj,i+(k-1)*Mt) = h_f(k);
        end
    end
end

H=[H_int2];
for i = 1:N-1
    H=[H,[zeros(Mr*i,Mt);H_int2(1:(N-i)*Mr,:)]];
end
```

<svd_decompose_channel.m>

```
function [U, S, V] = svd_decompose_channel(Mt, Mr, h_f, N);
% [U S V] = svd_decompose_channel(Mt, Mr, h_f, N);
%
% Function decomposes the channel at each subcarrier into its SVD components
%
%   Mt - # Tx antennas
%   Mr - # Rx antennas
%   h_f - MIMO impulse response - Mr rows, Mt*L columns, where L is the
number of
%         channel taps
%   N - # subcarriers

U = [];
S = [];
V = [];
for i = 1:N
    [Utmp Stmp Vtmp] = svd(h_f(:,(i-1)*Mt+1:i*Mt));
    U=[U Utmp];
    V=[V Vtmp];
    S=[S Stmp];
end

S = sum(S,1);
```

<BitLoad.m>

```
function [bits_alloc,energy_alloc]  =
BitLoad(subchan_gains,total_bits,num_subc,gap,noise,M)

% Bit Loading Algorithm
% ---------------------
%
% Inputs :
%          subchan_gains     : SubCarrier Gains
%           total_bits      : Total Number of bits
%            num_subc       : Number of Subcarriers
%            gap            : Gap of the system
%            noise          : Noise Power
%               M           : Max Constellation Size
% Outputs:
%          bits_alloc       : Bits allocation for each subchannel
%          power_alloc      : Total Power allocation
% ---------------------------------------------------------------

% Compute SNR's for each channel
SNR = ComputeSNR(subchan_gains,noise,gap);

% This function just initializes the system with a particular bit
% allocation and energy allocation using Chow's Algorithm. This is
% further efficientize using Campello's Algorithm

[bits_alloc, energy_alloc] = chow_algo(SNR,num_subc,M);

% Form the Energy Increment Table based on the present channel
% gains for all the subchannels in order to be used by Campello
% Algorithm

energytable = EnergyTableInit(SNR,M);

% Efficientize the algorithm using the Campello's algorithm

[bits_alloc,energy_alloc] =
campello_algo(bits_alloc,energy_alloc,energytable,total_bits,num_subc,M);
```

<ComputeSNR.m>

```
function SNR = ComputeSNR(subcar_gains,noise,gap)
% function SNR = ComputSNR(subcar_gains,noise,gap)
%
% this function calculate the SNR, or more typically the GNR of each
% subcarrier

SNR = abs((subcar_gains.^2)./(noise*gap));
```

<chow_algo.m>

```
% Chow's Algorithm
% ----------------
% This is based on the paper by Chow et al titled
%
```

```
% A Practical Discrete Multitone Transceiver Loading Algorithm
% for Data Transmission over Spectrally Shaped Channels.IEEE Trans
% on Communications. Vol. 43, No 2/3/4, pp. 773-775, Feb/Mar/Apr 1995

function [bits_alloc, energy_alloc] = chow_algo(SNR,num_subc,M)

for i = 1:num_subc
% Assuming each of the subchannels has a flat fading, we get initial
estimate
% of the bits for each subchannel
  tempbits        = log2(1 + abs(SNR(i))); % bits per two dimension.
  roundtempbits   = round(tempbits); % round the bits

  if (roundtempbits > M) % Limit them between 2 and 15
     roundtempbits = M;
  end

  if (mod(roundtempbits,2)== 1 & roundtempbits ~= 1)
      roundtempbits = roundtempbits -1;
  end

  if roundtempbits > 0 % Calculate the Energy required for the subchannel
     energy_alloc(i)  = (2^roundtempbits-1)/SNR(i) ;
  else
     energy_alloc(i) = 0;
  end

  bits_alloc(i) = roundtempbits; % Update the BitSubChan
end
```

<EnergyTableInit.m>

```
function energytable = EnergyTableInit(SNR,M);

% Inputs:
%   subcar_gains : Subcarrier Gains
%              M : max Constellation Size
%            Gap : Gap of the system
%          Noise : Noise Power
%  Outputs:
%    energytable : Energytable
%
% Based on the Subcarrier Gains, we calculate the energy
% increment required by each subcarrier for transmitting
% 1,2 ,4 ,6,8 bits.
% Energy = 2^(i-1)/subcar_gains;
% ------------------------------------------------------

%subcar_gains = (subcar_gains.^2)/(Gap*Noise);

energytable = abs((1./SNR)'*(2.^([1:M+1]-1)));

% Increase the energy value for constellation size of more than M to
% a very high value so that it is not assigned.

energytable(:,M+1) = Inf*ones(size(energytable(:,M+1)));

for i = 3:2:M
    energytable(:,i) = (energytable(:,i) +energytable(:,i+1))/2;
    energytable(:,i+1) = energytable(:,i);
end
```

<campello_algo.m>

```
% campello_algo.m
% --------------
% This function is used by Campello's algorithm to allocate bits and energy
for
% each subchannel optimally.

function [bits_alloc, energy_alloc] =
campello_algo(bits_alloc,energy_alloc,energytable,total_bits,num_subc,M)

bt = sum(bits_alloc);

%  We can't transmit more than M*(Number of subchannel) bits
if total_bits > M*num_subc
    total_bits = M*num_subc;
end

while (bt ~= total_bits)
    if (bt > total_bits)

        max_val = 0;
        max_ind = ceil(rand(1)*num_subc);

        for i = 1:num_subc

            if bits_alloc(i) ~= 0
                temp = energytable(i,bits_alloc(i)) ;
            else
                temp = 0;
            end

            if (temp > max_val)
                max_val = temp;
                max_ind = i;
            end

        end

        if (bits_alloc(max_ind) > 0)
            bits_alloc(max_ind) = bits_alloc(max_ind) -1;
            energy_alloc(max_ind) = energy_alloc(max_ind) - max_val;
            bt = bt-1;
        end

    else


        min_val = Inf;
        min_ind = ceil(rand(1)*num_subc);
        for i = 1:num_subc
            if bits_alloc(i) ~=0 & bits_alloc(i) < M+1
                temp = energytable(i,bits_alloc(i) + 1);
            else
                temp = Inf;
            end
            if (temp < min_val)
                min_val = temp;
                min_ind = i;
            end
        end


        if (bits_alloc(min_ind) < M)
            bits_alloc(min_ind) = bits_alloc(min_ind) +1;
            if (min_val==inf)
```

```
                min_val = energytable(min_ind,bits_alloc(min_ind));
            end

            energy_alloc(min_ind) = energy_alloc(min_ind) +min_val;
            bt = bt+1;
        end

    end

end

for i = 1:length(bits_alloc)
    if (mod(bits_alloc(i),2) == 1 & bits_alloc(i) ~=1)
        [bits_alloc,energy_alloc] =
ResolvetheLastBit(bits_alloc,energy_alloc,i,energytable,num_subc);
    end
end
```

<ResolvetheLastBit.m>

```
function  [bits_alloc, energy_alloc] =
ResolvetheLastBit(bits_alloc,energy_alloc,index,energytable,num_subc)
% function [bits_alloc, energy_alloc] =
ResolvetheLastBit(bits_alloc,energy_alloc,index,energytable,num)_subc)
%
% this function resolve the last bit to deal with the single violated bit
% constraint.

max_val = 0;

for i = 1:num_subc
    if (i ~= index & bits_alloc(i) == 1)
        if bits_alloc(i) ~= 0
            temp = energytable(i,bits_alloc(i)) ;
        end
        if (temp > max_val)
            max_val = temp;
            max_ind = i;
        end
    end
end

min_val = Inf;
for i = 1:num_subc
    if (i~= index & bits_alloc(i) == 1)
        if bits_alloc(i) ~=0
            temp = energytable(i,bits_alloc(i) + 1);
        end
        if (temp < min_val)
            min_val = temp;
            min_ind = i;
        end
    end
end

if (min_val < max_val)
    bits_alloc(min_ind) = bits_alloc(min_ind) + 1;
    bits_alloc(index) = bits_alloc(index) - 1;
    energy_alloc(index) = energy_alloc(index) - min_val;
else
    bits_alloc(max_ind) = bits_alloc(max_ind) - 1;
    bits_alloc(index) = bits_alloc(index) + 1;
    energy_alloc(index) = energy_alloc(index) + max_val;
```

```
end
```

<modulate.m>

```
function y = modulate(x,b,e, s2,s4,s16,s64,s256)
% function y = modulate(x,b,e, s2,s4,s16,s64,s256)
%
%    y - modulated output, in the form of a row vector
%    x - a vector of input bits, for all the subcarriers (row vector)
%    b - subcarrier bit allocation (64 elements in this matrix, each one
%        corresponding to the number of bits to be allocated to the
subcarrier
%        having the same index)
%    e - subcarrier energy allocation (64 elements in this matrix, each one
%        corresponding to the energy to be allocated to the subcarrier
%        having the same index)
%    s_ - the encoder for a given constellation size

y=[];

b2 = zeros(1,length(b));
b2(1) = 1;
for i = 1:length(b)
    b2(i+1) = b(i) + b2(i);
end

for i = 1:length(b)
    switch b(i)
    case {1}
        y = [y s2(x(b2(i))+1)*sqrt(e(i))];
    case {2}
        y = [y s4(x(b2(i):(b2(i+1)-1))*[2;1]+1)*sqrt(e(i))];
    case {4}
        y = [y s16(x(b2(i):(b2(i+1)-1))*[8;4;2;1]+1)*sqrt(e(i))];
    case {6}
        y = [y s64(x(b2(i):(b2(i+1)-1))*[32;16;8;4;2;1]+1)*sqrt(e(i))];
    case {8}
        y = [y s256(x(b2(i):(b2(i+1)-
1))*[128;64;32;16;8;4;2;1]+1)*sqrt(e(i))];
    otherwise
        % must be zero bits allocated
        y = [y 0];
    end
end
```

<precode.m>

```
function [x_pre] = precode(Mt, x_mod, V, N);
% [x_pre] = precode(Mt, x_mod, V, N);
%
% Function precodes the modulated symbols prior to sending through the IFFT
module.  The precoder
% left-multiplies the channel at each tone, to eliminate the right unitary
matrix in the SVD
% decomposition of the channel.
%
%   Mt - # transmit antennas
%   x_mod - modulated symbols for each tone (the symbols are grouped
together in tones, ie: for 3
%           MIMO subchannels per tone, have 3 consecutive x_mod symbols)
%   V - the matrix of right unitary matrices of the channel, in row order
%   N - # subchannels

x_pre =[];
for i = 1:N
    x_pre = [x_pre; [V(:,(i-1)*Mt+1:i*Mt)]*transpose(x_mod((i-
1)*Mt+1:i*Mt))];
end

x_pre = transpose(x_pre);
```

<ifft_cp_tx_blk.m>

```
function ofdm_symbol = ifft_cp_tx_blk(inp_symbol,num_subc,guard_interval)
%------------------------------------------------------------
% IFFT and Circular Prefix Addition Block for Transmitter
% --------------------------------------------------------
% (EE359 Project: Andrew and Prateek)
% Inputs :
%   inp_symbol     : Input Symbol from all the Subcarriers
%   num_subc       : Number of Subcarriers
%   guard_interval : Guard Interval based on OFDM Symbol
% Outputs :
%   ofdm symbol    : Output of IFFT and Guard Interval
% ------------------------------------------------------------


ofdm_symbol = ifft(inp_symbol,num_subc);

if (guard_interval > length(ofdm_symbol))
    error(' The guard interval is greater than the ofdm symbol duration ');
end

% The guard symbol is the copy of the end of the ofdm symbol to the
beginning
% of the ofdm symbol.
guard_symbol = ofdm_symbol(end-guard_interval+1:end);

% Add the cyclic prefix to the ofdm symbol
ofdm_symbol = [guard_symbol ofdm_symbol];
```

<channel.m>

```
function y = channel(sig2, Mt, Mr, x, H, N);
% function y = channel(sig2, Mt, Mr, x, H, N)
%
% Channel transmission simulator
%
% inputs:
%   sig2 - noise variance
%   Mt - number of Tx antennas
%   Mr - number of Rx antennas
%   x - vector of complex input symbols (for MIMO, this is a matrix, where
each column
%       is the value of the antenna outputs at a single time instance)
%   H - frequency selective channel - represented in block-Toeplitz form for
MIMO transmission
%   N - number of symbols transmitted in OFDM frame
%
% outputs:
%   y - vector of channel outputs (matrix for MIMO again, just like x matrix)

% create noise vector sequence (each row is a different antenna, each column
is a
% different time index)  note: noise is spatially and temporally white
noise = sqrt(sig2)*1/sqrt(2)*(randn(Mt*N,1) + j*randn(Mt*N,1));

y = H*x + noise;
```

<fft_cp_rx_blk.m>

```
function rec_symbol = fft_cp_rx_blk(ofdm_symbol,num_subc,guard_interval)
%------------------------------------------------------------
% IFFT and Circular Prefix Addition Block for Transmitter
% ----------------------------------------------------------
% (EE359 Project: Andrew and Prateek)
% Inputs :
%   ofdm_symbol    : received ofdm symbol from the channel
%   num_subc       : Number of Subcarriers
%   guard_interval : Guard Interval based on OFDM Symbol
% Outputs :
%   rec_symbol     : Received Frequency Domain Signal after
%                    removal of circular prefix and FFT.
% ------------------------------------------------------------


if (guard_interval > length(ofdm_symbol))
    error(' The guard interval is greater than the ofdm symbol duration ');
end

% The guard symbol (cyclic prefix) is removed from the ofdm symbol
rec_time_symbol = ofdm_symbol(guard_interval+1:end);

% The FFT of the time domain signal after the removal of cyclic prefix
rec_symbol = fft(rec_time_symbol,num_subc);
```

<shape.m>

```
function shaped_vals = shape(rec_symbol, Mr, U, N);
% shaped_vals = shape(rec_symbol, Mr, U, N);
%
% This function shapes the received symbols prior to demodulation.  This
completes the
% decomposition of the channel into parallel subchannels.
%
%   rec_symbol - received symbol after IFFT block
%   Mr - # receive antennas
%   U - matrix of unitary matrices premultiplying MIMO subchannels
%   N - # OFDM subchannels

shaped_vals =[];
```

```
for i = 1:N
    shaped_vals = [shaped_vals; [U(:,(i-
1)*Mr+1:i*Mr)]'*transpose(rec_symbol((i-1)*Mr+1:i*Mr))];
end

shaped_vals = transpose(shaped_vals);
```

<demodulate.m>

```
function y = demodulate(x, b, e, h, s2,s4,s16,s64,s256, c2,c4,c16,c64,c256);
% function y = demodulate(x, b, e, s2,s4,s16,s64,s256, c2,c4,c16,c64,c256);
%
% Finds minimum distance estimate of each received signal and returns the
% corresponding binary codeword.  Use a Zero-Forcing approach for
convenience
%
%    y - modulated output, in the form of a row vector
%    x - a vector of input symbols, for all the subcarriers (row vector)
%    h - channel value (in frequency) for all subcarriers (64 elements)
%    b - subcarrier bit allocation (64 elements in this matrix, each one
%        corresponding to the number of bits to be allocated to the
subcarrier
%        having the same index)
%    e - energy allocation (64 elements in this matrix)
%    s_ - the encoder for a given constellation size
%    c_ - the codewords as vectors of bits for the indices

y=[];

for i = 1:length(b)
    switch b(i)
    case {1}
        [tmp, index] = min(abs(s2-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c2(index,:)];
    case {2}
        [tmp, index] = min(abs(s4-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c4(index,:)];
    case {4}
        [tmp, index] = min(abs(s16-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c16(index,:)];
    case {6}
        [tmp, index] = min(abs(s64-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c64(index,:)];
   case {8}
        [tmp, index] = min(abs(s256-1/h(i)/sqrt(e(i))*x(i)));
        y = [y c256(index,:)];
   otherwise
        index = 0;
   end
end
```