

2011

Greater sustained throughput and contention elimination in IEEE 802.11 with DS-CDMA

Bryan Plass
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Plass, Bryan, "Greater sustained throughput and contention elimination in IEEE 802.11 with DS-CDMA" (2011). *Graduate Theses and Dissertations*. 11999.

<https://lib.dr.iastate.edu/etd/11999>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Greater sustained throughput and contention elimination in IEEE 802.11 with
DS-CDMA

by

Bryan William Plass

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Co-majors: Computer Engineering;
Information Assurance

Program of Study Committee:
Doug Jacobson, Co-major Professor
Tom Daniels, Co-major Professor
Sang Kim

Iowa State University

Ames, Iowa

2011

Copyright © Bryan William Plass, 2011. All rights reserved.

DEDICATION

To my wife Miriam and our crazy children.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	viii
CHAPTER 1. INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Description	2
CHAPTER 2. BACKGROUND	4
2.1 Other Approaches	4
2.1.1 Distributed Coordination Function (DCF)	4
2.1.2 Point Coordination Function (PCF)	5
2.1.3 Hybrid Coordination Function (HCF)	6
2.1.4 Multi-Channel CDMA (MC-CDMA)	6
2.2 Direct Sequence CDMA (DS-SS)	7
CHAPTER 3. DESIGN AND IMPLEMENTATION	9
3.1 PN Codes	9
3.2 Spreading Factor	10
3.3 Bandwidth	10
3.4 MAC Modifications	11
3.5 Network Entry	11

CHAPTER 4. THEORETICAL ANALYSIS	12
4.1 Models	12
4.1.1 IEEE 802.11g	18
4.1.2 DS-CDMA	20
4.1.3 Fragmentation	21
4.2 Results	22
CHAPTER 5. SIMULATION	27
5.1 Simulation Selection	27
5.2 Simulators Attempted	29
5.2.1 NS-2	29
5.2.2 NS-3	29
5.2.3 OPNET Modeler	30
5.2.4 Simulink	31
5.2.5 Other Simulators	31
5.2.6 Emulab	33
5.3 OPNET Modeler: The Simulator of Choice	33
5.3.1 The Best Choice	34
5.3.2 Modifying the Target Model	37
5.3.3 The Quest	45
CHAPTER 6. LIMITATIONS AND KNOWN ISSUES	47
CHAPTER 7. RELATED AND FUTURE WORK	49
7.1 Related Work	49
7.2 Future Work	49
CHAPTER 8. CONCLUSION	52
APPENDIX A. ADDITIONAL CONTACTS	53
BIBLIOGRAPHY	56

LIST OF TABLES

Table 2.1	IEEE 802.11 Timing Intervals Given in Microseconds (μs).	5
Table 3.1	Binary Form of 14 XORd with 2 Distinct PN Codes.	10
Table 4.1	Timing Related Parameters.	15
Table 4.2	Modulation Related Parameters.	16
Table 4.3	Parameters Used for Theoretical Analysis.	22
Table 5.1	Simulator Decision Matrix.	28
Table 5.2	Simulation Parameters for Initial Run.	40

LIST OF FIGURES

Figure 4.1	Non-Overlapping Channels for 2.4 GHz.	14
Figure 4.2	Transmit Spectrum Mask of OFDM.	14
Figure 4.3	Lifecycle of IEEE 802.11 DCF with RTS/CTS.	19
Figure 4.4	IEEE 802.11 DCF Backoff Procedures.	20
Figure 4.5	IEEE 802.11 DCF Fragmentation.	21
Figure 4.6	Throughput Comparison Between IEEE 802.11g and Proposed DS-CDMA Scheme.	25
Figure 4.7	Effects of Contention Window Size on Throughput.	26
Figure 5.1	Simulink IEEE 802.11b Model.	32
Figure 5.2	OPNET Modeler's 802.11b Model.	33
Figure 5.3	OPNET Modeler's GUI.	35
Figure 5.4	Options to Setup IEEE 802.11g Models.	36
Figure 5.5	OPNET wlan_mac Process Model.	38
Figure 5.6	Simulation Results Using Default and Modified Values.	41
Figure 5.7	Throughput, Offered Load, and Data Sent.	42
Figure 5.8	Proposed Packet Gormat.	42
Figure 5.9	Detection Using Rake Receiver.	44
Figure 5.10	Correlation of User Data.	45

ACKNOWLEDGEMENTS

I would like to express my thanks to those who have assisted me throughout my graduate studies.

First, my program of study committee, Dr. Doug Jacobson, Dr. Tom Daniels, and Dr. Sang Kim. Their assistance throughout this process was vital to my success. They have a willingness to help not only me but all students who seek them for guidance.

I would also like to thank all of my friends and peers. It has been said that two brains are better than one. This adage must have been written by a graduate student. For without many of you, I would not have been so successful. My only hope is that you received assistance from me in like measure.

Finally, I would like to thank my wife. For these past 18 months, she has selflessly raised our children so that I could be free to concentrate on my studies. Without this and her continual support, none of this would have been possible.

ABSTRACT

In today's networks, IEEE 802.11 Wi-Fi is commonplace. While providing great ease of use and mobility, IEEE 802.11 also has many flaws. One of these is contention. Contention is the means that wireless stations compete to determine who gets to transmit. During the contention times, the medium is sitting idle. Idle time equates to wasted bandwidth. This research presents a novel technique to eliminate contention in a wireless network while actually increasing sustained throughput by the use of Direct Sequence Spread-Spectrum Code Division Multiple Access (DS-SS-CDMA).

CHAPTER 1. INTRODUCTION

Wireless networks have pilfered our lives. Although Wi-Fi networks are commonplace, few people think about the intricacies of how they work. Unfortunately, they do not operate very efficiently. This is due to the phenomenon called contention. Because the wireless medium is loosely controlled, it is difficult to devise a scheme that addresses this contention. As such, we have been satisfied with the status quo for years. However, there is a better way to get more out of wireless networks. That is the use of DS-SS. By using DS-SS, not only will contention be eliminated, but overall throughput on the network will be dramatically increased.

1.1 Introduction

Technology is continuing to push the envelope of data transmissions. As we continue to progress, technologically speaking, we continue to find new ways to maximize the amount of data that is being pushed through various transmission mediums. Somehow, technology eventually seems to right itself to allow users to push the envelope even further. Take for example the modem. For years, researchers had thought that plain old telephone system (POTS) lines were maxed out at 9,600 bits per second. They theorized that 14,400 bits per second was possible, but they were not able to reach it. This hurdle was eventually overcome by applying trellis coded modulation schemes that were proposed previously by Gottfried Ungerboeck in 1976 [1]. Because of this, in the 1990's, researchers soon surpassed the 9,600 bits per second and pushed the limits of the Shannon-Hartley theorem,

$$C = B * \log_2\left(1 + \frac{S}{N}\right), \quad (1.1)$$

where C equals the channel capacity, B is the bandwidth in Hz, and $\frac{S}{N}$ is the signal to noise

ratio [2].

Imagine for a moment, trying to do everything today on a 9,600 bps modem. It would not be possible! Eventually, the wireless medium is going to become the 9,600 bps modem of the early 1990s. The 54 Mbps that we enjoy from 802.11g will not be enough for the demands of our lives. Because spectrum is a scarce resource, the increase in bandwidth will most likely not come from increasing the frequency range. It will have to come from different, more ingenious sources.

1.2 Problem Description

The aforementioned progression is the case with wireless communications networks as well. Because of today's dispersed networks, the need is to feed data to more distant places, at an increased data rate, and at minimal cost. Users have exchanged their 100Mbps or 1Gbps wired connections for wireless connections in the name of mobility and convenience. However, these wireless connections come at a price, contention. In the IEEE 802.3 wired world, connections use carrier sense multiple access—collision detect (CSMA-CD). Prior to sending any traffic on the wired medium, CSMA-CD is able to detect the presence of traffic already on the link. If the medium is not free, it waits until it becomes free. However, in the wireless medium, this is not possible. Instead, wireless nodes use CSMA-CA (the “A” standing for avoidance).

There are many different schemes used to implement this avoidance, but most of them continue to exercise contention — meaning nodes contend for the *chance* to transmit on the free medium. Since only one node can transmit at a time, there must be a mechanism that determines which node is allowed to transmit at what time. Contention is the mechanism that is used to determine which node will transmit. Whenever there is contention, there is always wasted time because the next transmitting terminal must be sure the medium is free prior to transmitting. The problem only gets exasperated as the number of users increase.

We propose one novel technique to eliminate the contention. That is the use of CDMA. Code division multiple access (CDMA) comes in two flavors; multi-channel (MC-CDMA) and direct sequence (DS-CDMA). Some sources [3][4] call these frequency hopping (CDMA-FH) and

pseudo-noise (CDMA-PN), respectively. In order to standardize and simplify terminology and to use more commonly and more recognizable terms, we will use MC-CDMA and DS-CDMA.

CDMA is a spread-spectrum technique, meaning the signal is spread of a greater portion of the spectrum. In MC-CDMA, the theory is to spread the information over a wider amount of bandwidth, dividing the spectrum into many sub-channels. This spreading also applies to DS-CDMA. However, pseudo-noise or code, is added to the signal so that it can be differentiated from other signals transmitting on the same larger channel. In this manner, multiple stations can transmit at the same time without interfering with each other [3].

The remainder of this paper is constructed as follows: in the next chapter, we will discuss different approaches to eliminating contention for the reduction of wasted time, chapter 3 will discuss the design and implementation of the proposed DS-CDMA scheme, in chapter 4 we will analyze this scheme and compare it to the 802.11g scheme, chapter 5 discusses the scheme simulation, chapter 6 is the current development and future directions, and the final sections conclude this paper and contain references.

CHAPTER 2. BACKGROUND

While the current IEEE 802.11 framework suits the needs of most users, there will come a time when it will not be sufficient. Because of this, many different people have devised different schemes to try to better the current technology. While small gains have been made in these proposals, few if any have proven worthwhile to try to implement on a large scale basis. Therefore, we are still stuck using technology that has significant known flaws.

2.1 Other Approaches

There are many different methods that have been suggested and implemented to attempt to eliminate and reduce contention and wasted time in IEEE 802.11. Unfortunately, not one has successfully eliminated it. In the end, all schemes will have either contention or polling. Contention in and of itself induces wasted time. Polling reduces this wasted time, but does not eliminate it. Furthermore, systems using a polling technique may incur additional wasted time if there are a number of stations that do not have any traffic to send—thereby wastefully querying stations.

2.1.1 Distributed Coordination Function (DCF)

In IEEE 802.11, the default technique deployed is the distributed coordination function (DCF). In order to understand DCF and some future techniques, one must first understand the timing intervals that are involved. There are multiple timing intervals. However, for this paper, we will concentrate on only four. Per [5], the timing intervals from shortest to longest are short interframe space (SIFS), slot time, PCF interframe space (PIFS), and DCF interframe space (DIFS). PIFS equals a SIFS plus a slot time. A DIFS equals a SIFS plus two slot times.

The actual times can be seen in Table 2.1.

Table 2.1 IEEE 802.11 Timing Intervals Given in Microseconds (μs).

	SIFS	Slot	PIFS	DIFS
IEEE 802.11a	16	9	25	24
IEEE 802.11b	10	20	30	50
IEEE 802.11g	10	9 or 20	19 or 30	28 or 50

In order for a station (STA) to determine if the medium is free or not, it must wait for a minimum of DIFS. In most cases, it will have to wait for DIFS plus any remaining time left of its backoff interval, i.e., contention window. So, at best case when the backoff timer = 0, it will wait DIFS. This time does not include the time that is added for ACKs (2 x SIFS per ACK) and the time during the request to send (RTS) / clear to send (CTS) volley (2 x SIFS per RTS/CTS).

All of this time adds up. Because in IEEE 802.11 terms are expressed in the time domain, any amount of idle time is wasted bandwidth. Due to the fact that time can never be recalled, the wasted bandwidth can never be recaptured and used elsewhere.

2.1.2 Point Coordination Function (PCF)

In order to eliminate this contention, IEEE 802.11 can use PCF or point coordination function. Although PCF was designed to eliminate or reduce the contention within the MAC operation, it still is not optimal. This is because it uses a polling mechanism to poll the STAs to see if they have traffic to send. All of this polling takes up time and may come across a station that does not have any traffic to send — thereby wasting additional time. The AP also is in control of the polling list, and therefore injects an added unfairness to the PCF scheme. The additional fact that DCF is used as the basis for PCF also means that contention still exists and cannot be eliminated.

2.1.3 Hybrid Coordination Function (HCF)

The hybrid coordination function (HCF) was introduced as a means to implement quality of service (QoS). The basic notion is that different types of traffic are marked differently, depending on their priority. When a STA is given a transmission opportunity (TXOP), the STA decides what traffic to transmit based on its queues. HCF uses enhanced distributed channel access (EDCA) and HCF controlled channel access (HCCA). EDCA is contention based and HCCA is polling based. While the notion of categorizing traffic, allowing the highest priority traffic to be sent first, sounds appealing, HCF still maintains EDCA, a contention mechanism.

2.1.4 Multi-Channel CDMA (MC-CDMA)

While not a MAC access method, multi-channel code division multiple access (MC-CDMA) is a type of multiple access that uses multiple sub-carriers or sub-channels. In MC-CDMA, STAs use multiple narrow sub-carriers to carry data. Multiple STAs put data on various sub-carriers by using different codes. These sub-carriers are then multiplexed using Orthogonal frequency-division multiplexing (OFDM). The result are codes that are “long in time and narrow in bandwidth” [4]. There are a few benefits to using MC-CDMA over previously discussed mechanisms. First, because MC-CDMA uses multiple sub-carriers that are spread across the spectrum, it is less susceptible to interference. Secondly, it uses frequency hopping, indicating that multiple STAs can access the medium simultaneously by using different hop-sets.

Reference [4] applies MC-CDMA to the IEEE 802.11a MAC protocol. In doing so, there is not much within the PHY and MAC that needs to be changed, as IEEE 802.11a is based on OFDM. However, because it is based on the same MAC protocol, it must still adhere to the same DCF rules. Therefore, this scheme will still need to wait the required DIFS and backoff intervals before transmitting on the medium. If it implements RTS/CTS, it will also need to wait the required SIFS time periods as well.

2.2 Direct Sequence CDMA (DS-CDMA)

Consumers of Wi-Fi have been satisfied with its performance for years. Many people do not question whether it can be better. That type of thinking does not sit well with the author. Therefore, he sent an email out to peers asking for some assistance. In response, a friend and colleague shared the following response when he asked about utilizing CDMA within IEEE 802.11:

We had a guest speaker yesterday in the Wireless LAN class that's in the Wi-Fi Alliance and IEEE 802.11 standards group. I asked him if there was any place for CDMA within 802.11 and he kinda chuckled. He said that there's such a mindset of contention-based solutions within 802.11 that many people can't even conceive of a shared medium solution like CDMA.

The sad part of this statement is that from this expert's vantage point, the world is stuck with status quo. We are eternally bound by the means that we now live by. His chuckle indicates that he does not necessarily conform to the status quo and is willing to think outside the box. Thinking outside the box is exactly what happens when CDMA is applied to IEEE 802.11.

When CDMA is done over direct sequence spread-spectrum, it is called DS-CDMA. The idea behind DS-CDMA is to spread the data by the use of codes over a larger bandwidth. The proposed DS-CDMA scheme spreads across the entire spectrum instead of narrow sub-carriers. The pseudo-noise (PN) codes are applied to the data before it is sent across the medium. Because each STA has a unique PN code, other STAs only accept data that is encoded with their code. All other traffic is considered noise and discarded. When traffic with the correct PN code destined for a STA, the STA quickly XORs the incoming traffic with the PN code to reveal the true data [3].

Simply stated, the application of DS-CDMA will provide the "shared medium solution" that the guest lecturer mentioned. It will enable wireless networks to accept more simultaneous users as well as provide more useable and greater sustained throughput than the current IEEE 802.11

wireless networks do. With it in place, we will be able to move past the status quo and move into a world of greater efficiency in our wireless world.

CHAPTER 3. DESIGN AND IMPLEMENTATION

When people think of CDMA, they think of either one of two things. Cell phones or satellites. This is because CDMA technology is heavily used in both of these areas. Rarely do people think of CDMA in the Wi-Fi arena.

We propose a novel scheme that allows IEEE 802.11 access without contention, thereby, providing a greater sustained throughput. This is accomplished by applying pseudo-noise (PN) codes to the data stream. With these PN codes, the STAs and the access point (AP) can differentiate data streams from other data streams.

3.1 PN Codes

The concept of using “codes” to separate data can be a little daunting. This is because we tend to think in terms of frequency separation and time separation. For simplicity, we provide the following examples. Frequency separation is witnessed by changing the station on the television or radio. Time separation is best described as a two-way conversation—when one person is done talking the other may talk. But the idea of code separation does not have to be difficult. This is best described in terms of languages. Two pair of partners can effectively communicate with their other partner at a dinner table if each set of partners speaks in a different language, say English and Chinese. The receivers are able to differentiate their language and tune out the other language and consider it noise [6].

The same is true for data communications. Because data is reduced down to 1’s and 0’s, we can manipulate the 1’s and 0’s by adding codes to them so they are distinguishable by the appropriate receiving station. For example, say two separate stations transmit the number 14 at precisely the same moment. In binary form 14 is 1110. STA1 is given a PN code of 1010 and

STA2 is given 1111. When we XOR the number with the PN codes two distinct data streams are created as seen in Table 3.1. The AP will then be able to receive these streams, apply the PN codes using XOR, and sort the data as is necessary.

Table 3.1 Binary Form of 14 XORd with 2 Distinct PN Codes.

STA ₁	STA ₂
1110	1110
⊕ 1010	⊕ 1111
0100	0001

3.2 Spreading Factor

In the preceding example, we used a spreading factor of one. Spreading factor (SF) can be explained by,

$$SF = \frac{\text{chiprate}}{\text{bitrate}}, \quad (3.1)$$

where the bit rate is the number of bits of data transmitted over time T and chip rate is the number of PN codes, or chips, transmitted over the same time T [7]. Because the goal is to spread out and utilize then entire bandwidth, we apply the codes to one bit at a time. Thus, if STA₁ wanted to transmit the number 9, 1001 in binary, the resulting transmission would be 0101101010100101. Therefore the SF would be,

$$SF = \left(\frac{16}{4} \right) = 4. \quad (3.2)$$

This spreading factor is sometimes referred to as processing gain [2][3][8]. This is because there is a gain in the signal-to-noise ratio as the bandwidth is spread. This can be seen in (1.1).

3.3 Bandwidth

In the current IEEE 802.11b/g/n schemes, the 2.4Ghz ISM band is used. Within this spectrum, there is just over 83.5Mhz available for use. Currently, in North America, it is

broken down into eleven overlapping channels of 25Mhz each. Channels 1, 6, and 11 do not overlap and have a 3Mhz guard band between each [9]. This means that in any given network, more than $\frac{2}{3}$ of the available bandwidth is being wasted because there is only one channel being used at a time.

3.4 MAC Modifications

The author is completely aware that DS-CDMA occurs at the PHY layer and the contention we are trying to mitigate occurs at the MAC level. As such, there must be modifications done to the MAC layer in order to make this scheme effective. The MAC layer protocol will need to be revised so that DCF essentially no longer functions. Stations will be able to transmit immediately when they have data to transmit without having to incur time penalties that are part of contention. In doing so, the CSMA/CA scheme may be lost. We are okay with this as it is not as important to stay within the bounds of IEEE 802.11 as it is to develop and design a scheme that eliminates contention. With this design, we will move past the long-lived status quo.

3.5 Network Entry

Under this scheme, the association and authentication of new STAs to the AP will also have to be modified. There is really no place for new STAs to contact the AP for network entry. Because the new STA has not been assigned a unique PN code by the AP, its transmission will be uncoded. As such, the AP will see this transmission as more noise on the network and rightfully discards it. In our proposal, the STAs and AP will have to be modified with a default or programmable code — similar to a WEP key or WPA passkey. With this applied, the AP will be able to identify the new STAs association and authentication requests and be able to assign a unique code to the STA. This code will remain on the STA so that in the event a disassociation occurs, the STA may reassociate with the AP.

CHAPTER 4. THEORETICAL ANALYSIS

In order to determine if this scheme is feasible, we will perform a theoretical analysis of it and analyze it against a current IEEE 802.11 scheme. This comparison will verify that there is an advantage to using this scheme.

One of the biggest reasons why we wanted to tackle this problem is the disillusion that vendors provide to the public when it comes to Wi-Fi. Products are labeled as producing 54 Mbps. Anyone who has used a IEEE 802.11g product knows that realizing this 54 Mbps is not a reality in practice. Actual usable bandwidth is much lower. Couple this with Shannon-Hartley's theorem, (1.1), and things get even more confusing.

A good example of this is this workstation. Currently, the local AP sees this STA as having -11dB of signal and -95dB of noise. By plugging the values into (1.1), the current capacity should be 141.01 Mbps given that the channel bandwidth, B , is 20000000. Rest assured, this computer is not getting this kind of performance.

4.1 Models

In our two different models, we must make some assumptions in order to perform this analysis. First, there are no hidden terminals, such that all terminals can see all other terminals in the network. Secondly, there are a fixed number of stations—each having packets ready for transmission. All packets are of fixed length. Next, for the sake of simplicity, all transmissions are received on the first time they are transmitted. Therefore there are no collisions and no retransmissions. Since we are looking at achieving the maximum throughput, we will also assume a clear channel with no bit errors so this is possible. Finally, we must determine what metric to measure. The parameter we have chosen to measure is throughput. Simply put, this

is the amount of user data that is received by the receiving station over a given time frame identified by sub-time periods and can be given by the equation,

$$S_d = \frac{L_d}{T_w + T_s + T_c}, \quad (4.1)$$

where L_d is the length of the data payload in bits, T_w , T_s , and T_c are the time spent wasted, successful, or in collision, respectfully. Since there are no collisions, $T_c = 0$ and is only given here as a placeholder [14].

Furthermore, in order to make fair comparisons, we must investigate the IEEE 802.11 standard [5] to ensure that the parameters used are correct. Since we are looking at the maximum throughput of 802.11g, we will look at that area. The big thing here is the advertisement of the 54 Mbps data rate.

It is interesting to note that some sources [9], [34], and [35] list the channel size for IEEE 802.11g to be 22 MHz wide. Because [5] uses circular referencing throughout the document, referring to various subclauses, it is difficult to ascertain what the proper channel size is. For example, in Table 15-7 of [5], all the channel IDs and their corresponding center frequencies are listed. When looking at this table, one might think that the channel size is 22 MHz. This would be a correct statement, if the network contains a mix of IEEE 802.11b and g STAs. In determining the channel bandwidth of an IEEE 802.11g system, we must assume that it is an IEEE 802.11g *only* network. With this assumption, the standard uses some more circular referencing and refers back to the IEEE 802.11a subclause. From there, we know that the channel size is 20 MHz. Graphically, this can be seen clearly in Figure 4.1 [36]. The reason for all this is because IEEE 802.11g is required to be backward compatible with IEEE 802.11b. Therefore, the DSSS section and 22 MHz channel size still applies in some instances.

Another thing to understand is the fact that spectrum just does not stop at a specified frequency. It will continue to bleed over to adjacent channels. This is why we only have three non-overlapping channels in North America. The phrase “non-overlapping” is actually misleading. The channels actually overlap, however the signal from one channel, say channel 1, is so small that it is completely negligible to users on channel 6. Figure 4.2 [5] shows that

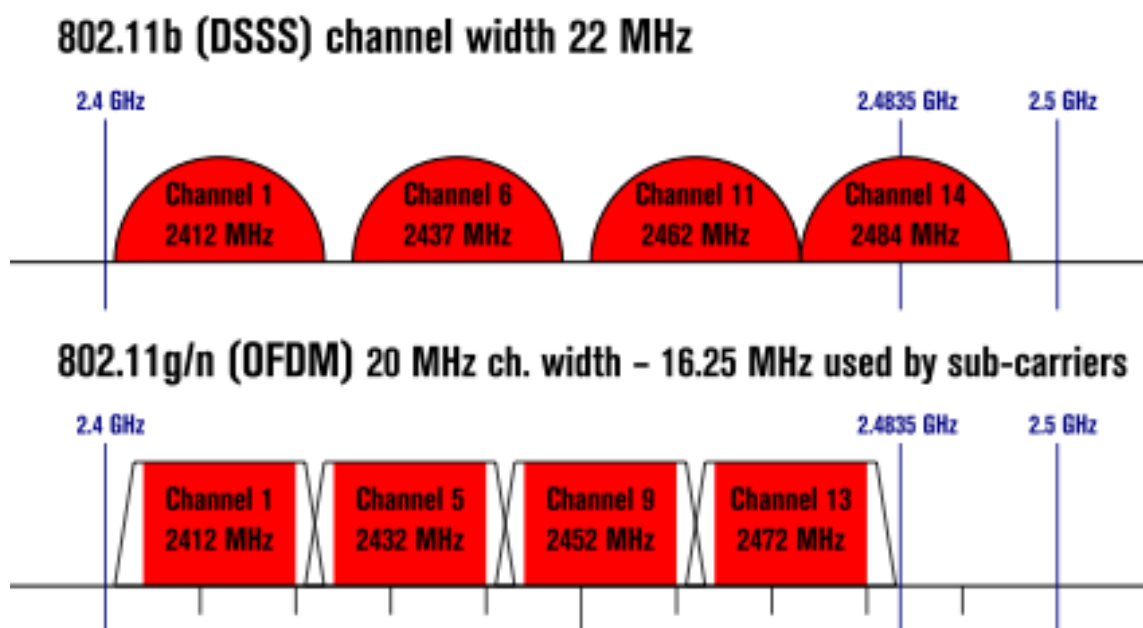


Figure 4.1 Non-Overlapping Channels for 2.4 GHz.

the “legs” of the spectrum mask continue to go further out. Keep in mind, that if this were channel 1, channel 6 would be centered 25 MHz to the right of f_c .

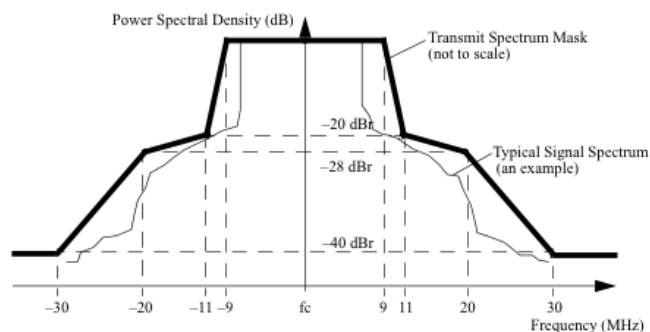


Figure 4.2 Transmit Spectrum Mask of OFDM.

Now that it has been determined that a 20 MHz channel will be used, we need to then determine the data rates that will be used. Since IEEE 802.11g advertises 54 Mbps, the question becomes how this data rate is derived. Once again, we will turn to the standard for the answer. Tables 4.1 and 4.2, extrapolated from [5], contain the information needed to derive the data rates advertised.

Table 4.1 Timing Related Parameters.

Parameter	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
N_{SD} : Number of data subcarriers	48	48	48
N_{SP} : Number of pilot subcarriers	4	4	4
N_{ST} : Number of subcarriers, total	$52 (N_{SD} + N_{SP})$	$52 (N_{SD} + N_{SP})$	$52 \cdot 52 (N_{SD} + N_{SP})$
ΔF : Subcarrier frequency spacing	0.3125 MHz (=20 MHz/64)	0.15625 MHz (=10 MHz/64)	0.078125 MHz (5 MHz/64)
T_{FFT} : Inverse Fast Fourier Transform (IFFT)/Fast Fourier Transform (FFT) Period	$3.2 \mu s (1/\Delta F)$	$6.4 \mu s (1/\Delta F)$	$12.8 \mu s (1/\Delta F)$
$T_{PREAMBLE}$: PLCP preamble duration	$16 \mu s (T_{SHORT} + T_{LONG})$	$32 \mu s (T_{SHORT} + T_{LONG})$	$64 \mu s (T_{SHORT} + T_{LONG})$
T_{SIGNAL} : Duration of the SIGNAL BPSK-OFDM symbol	$4.0 \mu s (T_{GI} + T_{FFT})$	$8.0 \mu s (T_{GI} + T_{FFT})$	$16.0 \mu s (T_{GI} + T_{FFT})$
T_{GI} : GI duration	$0.8 \mu s (T_{FFT} / 4)$	$1.6 \mu s (T_{FFT} / 4)$	$3.2 \mu s (T_{FFT} / 4)$
T_{GI2} : Training symbol GI duration	$1.6 \mu s (T_{FFT} / 2)$	$3.2 \mu s (T_{FFT} / 2)$	$6.4 \mu s (T_{FFT} / 2)$
T_{SYM} : Symbol interval	$4.0 \mu s (T_{GI} + T_{FFT})$	$8.0 \mu s (T_{GI} + T_{FFT})$	$16.0 \mu s (T_{GI} + T_{FFT})$
T_{SHORT} : Short training sequence duration	$8.0 \mu s (10 \times T_{FFT} / 4)$	$16.0 \mu s (10 \times T_{FFT} / 4)$	$32.0 \mu s (10 \times T_{FFT} / 4)$
T_{LONG} : Long training sequence duration	$8.0 \mu s (T_{GI2} + 2 \times T_{FFT})$	$16.0 \mu s (10 \times T_{FFT} / 4)$	$32.0 \mu s (10 \times T_{FFT} / 4)$

Table 4.2 Modulation Related Parameters.

Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSK})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20MHz channel spacing)	Data rate (Mb/s) (10MHz channel spacing)	Minimum sensitivity (dBm) (20MHz channel spacing)	Minimum sensitivity (dBm) (10MHz channel spacing)
BPSK	1/2	1	48	24	6	3	-82	-85
BPSK	3/4	1	48	36	9	4.5	-81	-84
QPSK	1/2	2	96	48	12	6	-79	-82
QPSK	3/4	2	96	72	18	9	-77	-80
16-QAM	1/2	4	192	96	24	12	-74	-77
16-QAM	3/4	4	192	144	36	18	-70	-73
64-QAM	2/3	6	288	192	48	24	-66	-69
64-QAM	3/4	6	288	216	54	27	-65	-68

With this information, the data rate can be found by,

$$R_d = \left(\frac{R * \log_2 M}{T_{SYM}} \right) * N_{SD}, \quad (4.2)$$

where R is the coding rate, M is the number of points on the M-ary size constellation, and N_{SD} is the number of subcarriers. So, for IEEE 802.11g, the derivation of the 54 Mbps data rate is (assuming 64-QAM modulation),

$$\begin{aligned} R_d &= \left(\frac{\frac{3}{4} * \log_2 64}{4} \right) * 48 \\ &= \left(\frac{\frac{3}{4} * (6) * 48}{4} \right) \\ &= \frac{6 * 36}{4} \\ &= \frac{216}{4} \\ &= 54Mbps. \end{aligned} \quad (4.3)$$

The purpose for going through this derivation was to find out where the 54 Mbps data rate comes from. With that in hand, we can now duplicate the process to find the appropriate data rate for the proposed scheme. Since the proposed scheme utilizes two separate channels, one for transmit and one for receive, the total bandwidth in the system must equal the total bandwidth in the IEEE 802.11g model. Therefore, the proposed scheme will have two 10 MHz channels at its disposal.

Some differences will need to be taken into consideration for the proposed scheme. First, there are no subcarriers as there are in IEEE 802.11g. All the data is carried across one large spread channel. Therefore, the Δ_F in Table 4.1 will equal,

$$\begin{aligned} \Delta_F &= \left(\frac{\text{channelsize}}{\text{subcarriers}} \right) \\ &= \left(\frac{10}{1} \right) \\ &= 10MHz. \end{aligned} \quad (4.4)$$

Next, T_{FFT} is,

$$\begin{aligned}
T_{FFT} &= \frac{1}{\Delta_F} \\
&= \frac{1}{10} \\
&= 0.1\mu s.
\end{aligned} \tag{4.5}$$

Finally, since there is only one carrier, there is no need for a guard interval between subcarriers like IEEE 802.11g. Therefore, the symbol interval can be represented by,

$$\begin{aligned}
T_{SYM} &= T_{GI} + T_{FFT} \\
&= 0 + 0.1 \\
&= 0.1\mu s.
\end{aligned} \tag{4.6}$$

Now that symbol duration is calculated, we can now take the values and plug them back into (4.2). When this is done, we get,

$$\begin{aligned}
R_{dc} &= \left(\frac{\frac{3}{4} * \log_2 64}{.1} \right) * 1 \\
&= \left(\frac{\frac{3}{4} * (6) * 1}{.1} \right) \\
&= \left(\frac{4.5}{.1} \right) \\
&= 45Mbps.
\end{aligned} \tag{4.7}$$

4.1.1 IEEE 802.11g

Since there are many different standards within the IEEE 802.11 protocol, we needed to choose one as the basis for the control group for the theoretical analysis. Although IEEE 802.11n is quickly becoming the norm, it is imperative that a standard be used that more closely emulated our scheme. For this, we ruled out IEEE 802.11n because of its MIMO technology. Therefore, we chose IEEE 802.11g. IEEE 802.11g is very widely used today and much research has been completed on it.

Let T_s equal the amount of time that it takes to send a successful data transmission. T_s is one complete cycle as is seen in Figure 4.3 [5]. All components must take place when

participating in the four-way handshake method. In the basic access method, which we will use, the RTS, CTS, and 2 x SIFS will not take place. All other items will remain. As can be seen, little of the overall time is used for the transmission of actual data. The remainder of the time (less Data and ACK) is wasted time, T_w , and considered overhead.

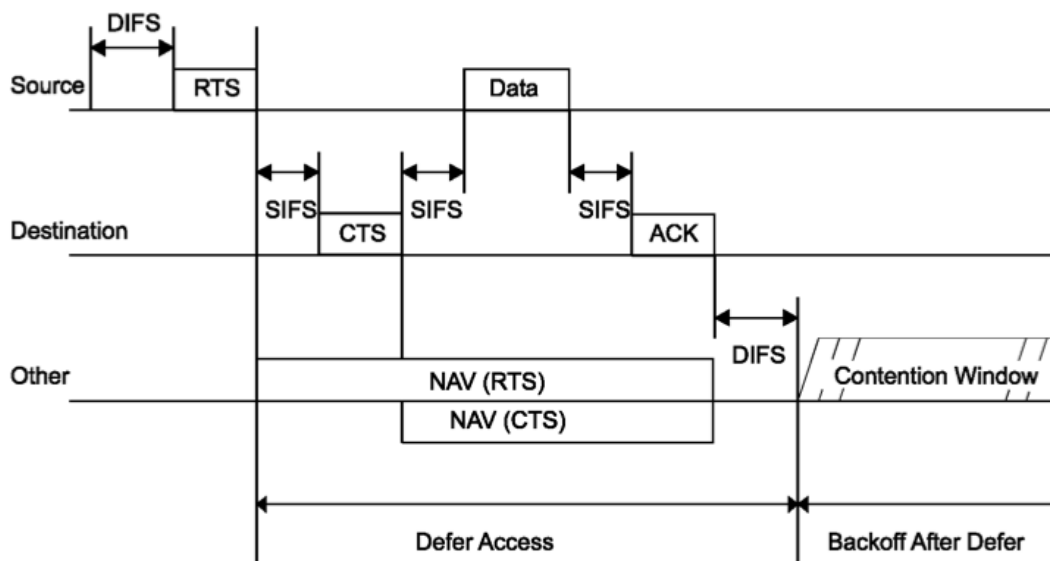


Figure 4.3 Lifecycle of IEEE 802.11 DCF with RTS/CTS.

The wasted time is calculated

$$T_{wd} = \frac{(CW_{min} - 1) * \sigma}{2}, \quad (4.8)$$

such that CW_{min} is the minimum value for the contention window and σ is equal to the slot length [14]. The contention window, CW, is a "Pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$ "... "where $CW_{min} \leq CW \leq CW_{max}$." [5]. Figure 4.4 [5] depicts how this backoff value is used within the context of multiple STAs in the network.

We also need to identify the time required to complete a successful transmission. This time is calculated by computing the MAC header (H_m), PHY header (H_p), ACK, header rate (R_h), data rate (R_d), data length (L_d), SIFS, and DIFS in the following way [14].

$$T_{sd} = \frac{H_m + H_p + ACK}{R_h} + \frac{L_d}{R_d} + SIFS + DIFS \quad (4.9)$$

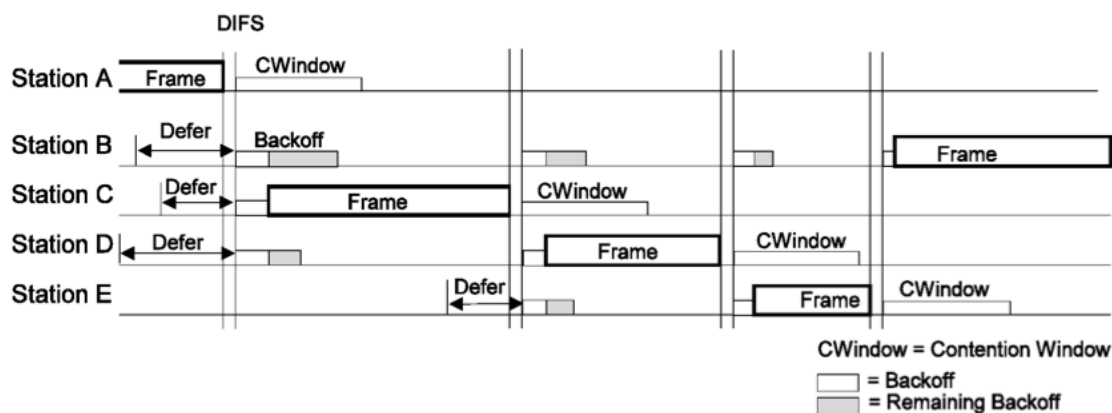


Figure 4.4 IEEE 802.11 DCF Backoff Procedures.

It is also important to note that the MAC and PHY header rates may differ, thus the different annotations, R_h and R_d respectively.

4.1.2 DS-CDMA

In the proposed scheme, DS-CDMA, the equations change only slightly. However, these changes provide a substantial boost in sustained throughput. Because DS-CDMA is not time dependent, but rather code dependent, STAs can transmit simultaneously. As a result, there is not a need to engage in an RTS/CTS scheme. There is also no need to participate in backoff procedures.

Therefore, for DS-CDMA, (4.1) can be written,

$$S_c = N * \left(\frac{L_d}{T_s} \right), \quad (4.10)$$

where N is the number of STAs wishing to transmit simultaneously. The T_w is negligible due to the only wasted time that will apply is the two-way propagation delay. We have eliminated this because it will apply to both models and will effectively cancel each other out. Once again, since there are no collisions, the T_c is also cancelled out. The only items remaining are the length of the data and the successful transmission. Due to the fact that multiple STAs can transmit simultaneously, one can easily see the advantage.

Because DS-CDMA does not participate in RTS/CTS nor backoff procedures, the successful transmission formula must also be modified. Therefore,

$$T_{sc} = N * \frac{H_m + H_p + ACK + L_d}{R_{dc}} \quad (4.11)$$

follows. Because the data rate and the header rates in the proposed scheme are equal, there is no need to separate the items. Also noteworthy to address is the inclusion of the ACK. While the destination could very well simultaneously respond with the ACK at any time, even during the next data transmission, the ACK is included in here for the completeness of the data transmission cycle.

4.1.3 Fragmentation

Also for completeness, one more item must be addressed. This is the concept of fragmentation. Since the DS-CDMA packets are going to be spread using a coding scheme, the number of bits required to be transmitted will increase significantly. Therefore, the packet

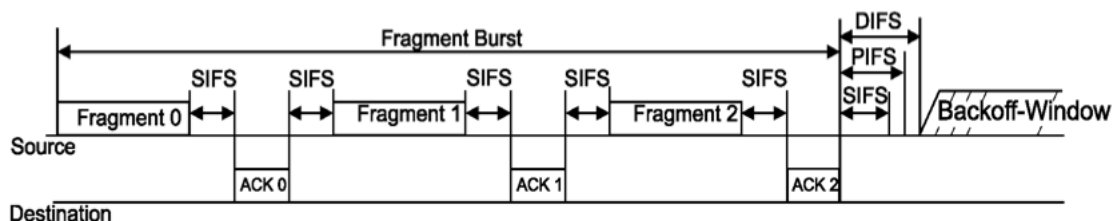


Figure 4.5 IEEE 802.11 DCF Fragmentation.

sizes will almost always exceed the maximum IEEE 802.11 MTU size of 2304 Bytes or 18432 bits. The additional overhead can be effectively captured by the following,

$$T_{sfc} = N * \frac{\left\lceil \frac{L_d}{MTU} \right\rceil * (H_m + H_p + ACK) + L_d}{R_{dc}}. \quad (4.12)$$

Of course, the IEEE 802.11g equation would be similarly crafted to capture the additional overhead. However, according to [5], fragmentation in the DCF operation adds even more

overhead as can be seen in Figure 4.5 [5].

To accurately capture this additional overhead, we modify (4.9) to get the successful transmission time of a fragmented DCF packet.

$$T_{sfd} = \frac{\left\lceil \frac{L_d}{MTU} \right\rceil * (H_m + H_p + ACK)}{R_h} + \frac{L_d}{R_d} + SIFS + DIFS + \left(\left\lceil \frac{L_d}{MTU} \right\rceil * \left((2 * SIFS) + \frac{ACK}{R_h} \right) \right) \quad (4.13)$$

From this point, we can now plug in numbers to verify the results.

4.2 Results

In order to effectively evaluate the results, we must first normalize all values. The easiest way to do this is to get all values into common time and data values. The time value that will be used is μs . The normalized data value will be bits—such as the maximum MTU size of 2304 Bytes being represented as 18432 bits. Table 4.3 gives the parameters used for the testing.

Table 4.3 Parameters Used for Theoretical Analysis.

Parameter	IEEE 802.11g	DS-CDMA	Unit
Slot Time (σ)	9	0	μs
SIFS	10	0	μs
DIFS	28	0	μs
ACK	112	112	bits
CTS	112	0	bits
RTS	160	0	bits
H_m	224	224	bits
H_p	192	192	bits
L_d	19200	192000	bits
Code Rate		10	bits
R_h	6	45	Mbps
R_d	54	45	Mbps
CW_{min}	16	0	
CW_{max}	1024	0	

Because the proposed DS-CDMA scheme uses PN coding, meaning every bit that is transmitted will be represented by 10 bits in our analysis. This causes problems in trying to fairly

analyze these two models. This is because if we use a 1500 Byte payload, the DS-CDMA model will be fragmented whereas the IEEE 802.11g model will not. Therefore, we have chosen a 2400 Byte payload to ensure both models will fragment at least once due to their data size being greater than the maximum MTU size.

By utilizing (4.1)-(4.13) above, we can show the increase in throughput offered by the DS-CDMA scheme over the IEEE 802.11g scheme. We find the time wasted in DCF by using (4.8),

$$\begin{aligned} T_{wd} &= \frac{(16 - 1) * 9}{2} \\ &= 67.5\mu s. \end{aligned} \quad (4.14)$$

Using (4.13), we find the total time for successful transmission is,

$$\begin{aligned} T_{sfd} &= \frac{\left\lceil \frac{19200}{18432} \right\rceil * (224 + 192 + 112)}{6} + \frac{19200}{54} + 10 + 28 + \left(\left\lceil \frac{19200}{18432} \right\rceil * \left((2 * 10) + \frac{112}{6} \right) \right) \\ &= \frac{2 * 528}{6} + 355.555 + 38 + (2 * (20 + 18.667)) \\ &= 176 + 355.555 + 38 + 77.334 \\ &= 646.889\mu s. \end{aligned} \quad (4.15)$$

By substituting the values in (4.14) and (4.15) into (4.1), we find that the throughput for IEEE 802.11g is,

$$\begin{aligned} S_d &= \frac{19200}{67.5 + 646.889} \\ &= 26.876Mbps. \end{aligned} \quad (4.16)$$

This seems reasonable since it is approximately what is normally realized on wireless networks. The other fact is that IEEE 802.11 is less than half-duplex—meaning that since only one STA can transmit at a time. From this half-duplex time, we have to take away any wasted time, so the 26.876 Mbps is a reasonable answer in a 54 Mbps network.

When the parameters are plugged into the proposed DS-CDMA scheme, the throughput less than ideal for the rates offered by IEEE 802.11. The benefit will be shown shortly. In order to not skew the numbers in favor of the proposed scheme, we will continue to utilize

only one STA, although the scheme has the capability of handling more STAs simultaneously. Since there is no wasted time in this scheme, we can look directly at the time for a successful transmission.

$$\begin{aligned}
 T_{sfc} &= 1 * \frac{\lceil \frac{192000}{18432} \rceil * (224 + 192 + 112) + 192000}{45} \\
 &= \frac{11(528) + 192000}{45} \\
 &= \frac{197808}{45} \\
 &= 4395.73\mu s.
 \end{aligned} \tag{4.17}$$

And when this value is inserted back into (4.10) we get,

$$\begin{aligned}
 S_c &= 1 * \left(\frac{19200}{4395.73} \right), \\
 &= 4.367Mbps.
 \end{aligned} \tag{4.18}$$

By comparing (4.16) and (4.18), it is not immediately apparent that the DS-CDMA scheme offers far superior throughput than regular IEEE 802.11g. This is because we tend to look only at the immediate numbers. However, because the proposed scheme lacks the wasted time, contention, and timing intervals that plague the DCF operation, it is free to serve multiple users simultaneously.

To understand this graphically, look at Figure 4.6. Although the proposed DS-CDMA scheme starts out with a seemingly low throughput, it quickly makes up for it. In a network with $N > 6$ users, DS-CDMA quickly outperforms the IEEE 802.11g system. This will become especially apparent when users continually demand greater throughput for network intensive applications. Applications that require a constant bit rate (CBR) will especially perform better on the proposed scheme because the medium is always free.

Figure 4.7 really paints the picture on how damaging the contention window (CW) can be to throughput. Remember that CW is just one component of the overall contention metric. As the delay incurred by the contention window increases, the throughput decreases dramatically. In order to simplify the analysis, the CW was assumed to be CW_{min} or 16. However, the CW can be as great as CW_{max} or 1024. This results in a throughput range of 3.697 Mbps to 26.876

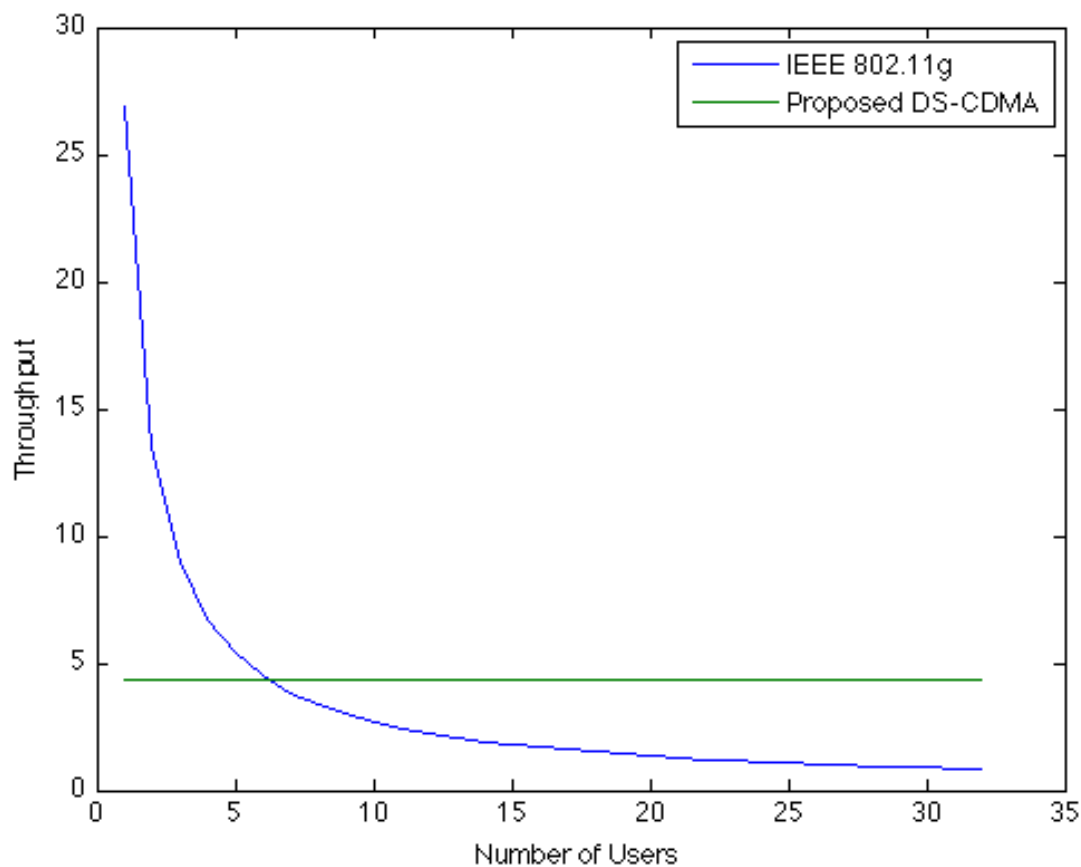


Figure 4.6 Throughput Comparison Between IEEE 802.11g and Proposed DS-CDMA Scheme.

Mbps. Since the CW is a "pseudo-random number drawn from a uniform distribution" over the interval, it is easy to see from Figure 4.7 that the maximum throughput of 26.876 Mbps represents the best case scenario for IEEE 802.11g. It quickly goes downhill from there. Also included in the figure is a change in payload to show the effects payload size has on throughput.

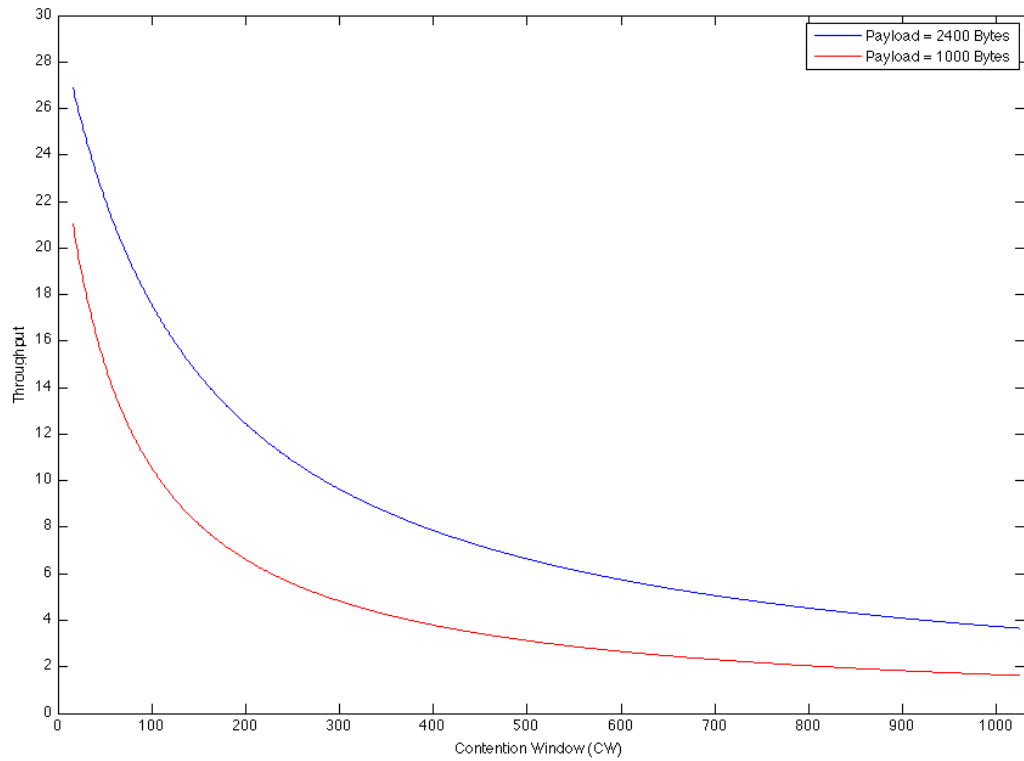


Figure 4.7 Effects of Contention Window Size on Throughput.

CHAPTER 5. SIMULATION

As with any product, determining whether the product is going to be successful in the marketplace is vitally important. Prior to production, one would like to simulate the hypothesis to ensure that it is going to perform to the specifications set forth in the initial analysis. This scheme is no different. Before vendors and consumers decide whether to invest heavily into this product, they will want to know, with some assuredness, that it is going to perform as advertised.

From simulation, products may be tested to ensure they will hold up. The simulation may then be transposed into a field programmable gate array (FPGA). FPGAs allow for quick hardware implementation of test products to validate the analysis and the simulation. Reference [15] has products in place that allow for quick implementation into FPGAs. Once proven in FPGAs, the transition to actual hardware is much quicker and provides quicker time to market. Using such a product on this scheme would provide useful in obtaining assurance in the product's viability in a very short amount of time.

5.1 Simulation Selection

As can be seen, there is a huge importance placed on accurate simulation. We recognized this and have place a considerable amount of time in attempting to produce a valid, accurate, and comprehensive simulation of the scheme. This being said, there were numerous factors involved in producing a legitimate simulation. First and foremost, the simulation must capture the parameters that are most important to not only the creators, but also to the eventual end users of the system. In our scheme, we determined that the foremost important factor was to determine the overall throughput of the link. This is because we believe that vendors mislead

Table 5.1 Simulator Decision Matrix.

	NS-2	NS-3	OPNET	Simulink	ShoX	Emulab
Capture Parameters (throughput)	X	X	X	X	X	X
Accurately Represent Protocol	X	X	X	-	X	X
IEEE 802.11g Model Available	X	X	X	-	X	X
CDMA Model Available	-	-	-	-	-	-
Recognized Modeler	X	-	X	X	-	-
Validated	X	-	X	X	-	-
Uniformity	-	-	-	-	-	-
Installation	-	X	X	X	X	-
Programming Language	-	X	X	X	X	-
GUI	-	-	X	X	X	-
MAC Layer	X	X	X	-	X	X
PHY Layer	-	-	X	X	X	X

the public by advertising the net bit rate versus the actual throughput. This net bit rate can be significantly more than what will be actualized by a user. Secondly, there are numerous simulators that exist today. Some are good representations of the actual product or protocol and some provide very poor results. In some instances, one simulator may provide great results for one protocol, but that same simulator may not provide viable results for another protocol. With that in mind, it was necessary to parse through the plethora of simulators. Next, simulators must be recognized in general by the community. This is because simulators must be validated. If we created our own simulator, for example, who is to say that we did not skew the code to provide results that favored our scheme? For this reason, we attempted to stay with the simulators that are frequently used in similar topics. Finally, we recognized that the data had to be uniform across both the proposed scheme and the control scheme. With this in mind, the decision was made to choose one simulator for both protocols, vice simulating one scheme on simulator X and simulating the other on simulator Y. In this way, there would not be any way to call into question the final results that would be presented.

5.2 Simulators Attempted

Through the course of the development of this scheme, we looked at and evaluated numerous simulators. Many of them had very strong characteristics that made them very appealing to the work. However, one-by-one, each one failed to meet the criterion laid forth in the previous section and Table 5.1.

5.2.1 NS-2

Network Simulator 2 or NS-2 [16] is by far the most prevalent simulator used in network communications simulation. For this reason, it was the first simulator that we investigated. NS-2 is an open source program, meaning that the code is open for contributions from the general public. As such, there is a considerable amount of code that other NS-2 users have contributed to it.

There are a few drawbacks that persuaded us away from using NS-2. First of all, NS-2 is a command line only simulator. While this is not an immediate disqualifier, it certainly provides some difficulty when the researcher is used to graphical user interface (GUI) simulators. Even the results require an additional package, Network Animator (NAM), to be viewed in any type of graphical format. To make matters worse, programming within NS-2 is done in the MIT Object Tcl, or OTcl for short, programming language. While we were not familiar with this language at all, we pressed on since lack of programming language knowledge in the past has been at most a large speed bump, not a road block. Upon installation, all hopes of using NS-2 fell apart as it completely failed installation. We tried twice on two different systems with no luck, thus forcing our hand toward a different simulator.

5.2.2 NS-3

From the onset, Network Simulator 3 or NS-3 [17] proved to be a much nicer fit to our research. It installed correctly on the first attempt. Like NS-2, it is open source and many user contributions can be found. Contrary to popular belief, NS-3 is not an upgrade to NS-2. It was developed from the ground up and is “intended as an eventual replacement for the popular ns-2

simulator.” [17] Where NS-2 is written in OTcl, NS-3 is written in C++. Although again we were not completely familiar with C++, it is close enough to the C programming language in which we do have a working knowledge. Additionally, NS-3 provides a very nice and extensive tutorial that is very easy to follow.

After attempting to implement our scheme utilizing the NS-3 simulator for about a week and a half, we sent an email to one of the code contributors. This contributor has a very strong knowledge of not only the NS-3 simulator, but also of the IEEE 802.11 protocol and CDMA. He was able to answer many questions in extensive detail. However, during multiple email exchanges, he mentioned that what I was trying to do would not be “a trivial modification” and that it would need “significant functionality in order to make this all work” [18]. Some of the challenges cited were overcoming the near-far problem and splitting the existing channel up into separate send and receive channels. While theoretically possible, it was not going to be a small undertaking — especially by someone with limited knowledge of the C++ language.

5.2.3 OPNET Modeler

The next logical simulator was OPNET Modeler [19]. OPNET is a commercial-off-the-shelf (COTS) product that is also very heavily used in research. Many papers in this area have been simulated using OPNET. Despite having a GUI front end, OPNET is not the easiest product to use. The author had some experience using OPNET from previous lab exercises and coursework. Once the simulations are built, it provides solid and consistent simulations time after time. However, the crux is getting the simulation built. Even the very well scripted lab exercises were difficult to complete. For this reason, the author did not automatically attempt the simulation in OPNET from the beginning.

Despite its shortcomings, OPNET has a very verbose library. A IEEE 802.11g framework is readily available and a network can be setup within a matter of minutes. This was nice because it provided for a quick setup of the control group. The simulation also provided results that were within tolerance of the results from the initial analysis. Developing a CDMA simulation within OPNET is a different story. OPNET Modeler does not contain a default CDMA library.

Although there are also user contributed models, there is only one that uses CDMA. This model was built many years ago in a previous version and no longer works with the version that is available at the university. This will be discussed more in a later section.

5.2.4 Simulink

MathWorks Simulink [20] is also another great simulation product. It provides a robust communications library that is used extensively in the Electrical Engineering community. The interface is very simple to use and can be picked up relatively quickly. Simulink also provides tutorials that cover the basics for getting a beginner started. Along with this, there are also many user contributed models to choose from.

However, there are a few problems with using Simulink to simulate the proposed scheme and the control scheme. The first is that an IEEE 802.11g model does not exist. One can find IEEE 802.11a and IEEE 802.11b models from either the built-in or user contributed libraries. Despite not having a suitable model, there is even a greater problem. The aforementioned models do not have a MAC layer attached to them. Therefore, they do not accurately model the CSMA/CA architecture. In doing so, the throughput data is high. In testing, we witnessed that the IEEE 802.11b model in Simulink provided a throughput of 9.91 Mbps (Figure 5.1), whereas the OPNET simulation resulted in a more realistic throughput of 4.79 Mbps on a similar 11 Mbps link (Figure 5.2). Without having an appropriate MAC layer, it does not matter if Simulink provides a simple, intuitive interface. It cannot be used within the context of our simulation parameters.

5.2.5 Other Simulators

In the quest to find an appropriate simulator, we briefly explored other simulators. This was an attempt to not only locate a simulator that met the simulator criterion in Table 5.1, but it also was a last ditch effort to find a simulator that was easier to use than OPNET, NS-2, or NS-3. In the search, we came across many different simulators. However, we only seriously considered two additional simulators: ShoX and Emulab.

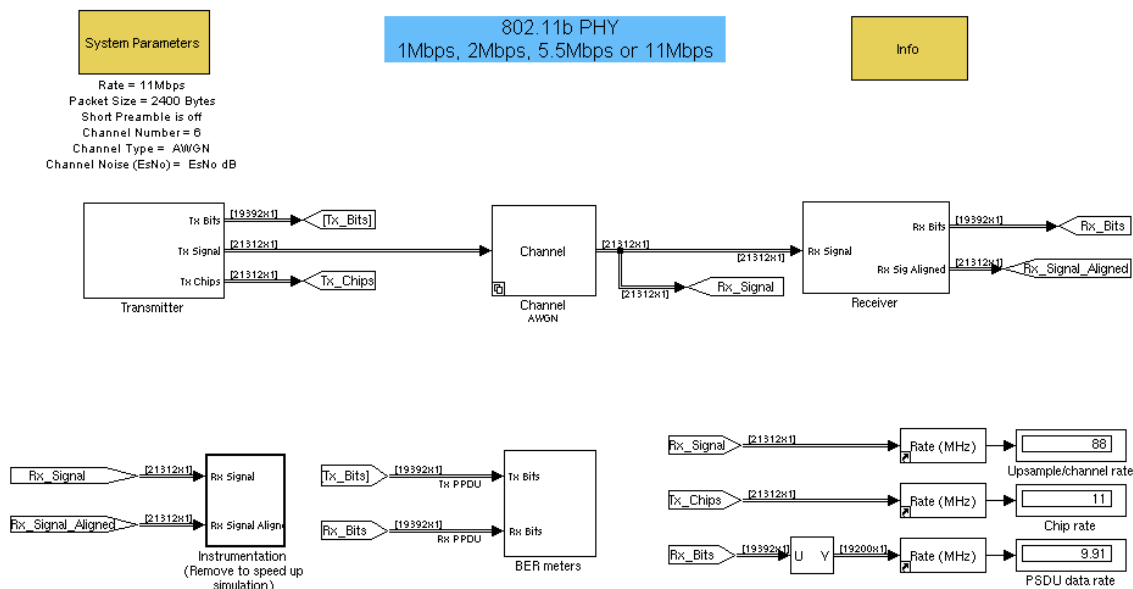


Figure 5.1 Simulink IEEE 802.11b Model.

5.2.5.1 ShoX

The first one that was considered was ShoX [21]. There were many items that provided the draw to this product. First, was that it utilizes Java — a more familiar programming language. Also ShoX is claimed to be “geared towards mobile wireless networks.” Additionally, according to its website, “ShoX is very straightforward to use” and “very easy to implement.” It also has GUI support and includes IEEE 802.11 DCF. All these were very enticing and very desirable.

Despite never hearing of this simulator before, we were willing to give it a shot because of all the capabilities that were previously documented. However, in reality, this simulator was anything but simple and “very easy to implement” as they claimed. It required setting up and using a subversion (SVN) server to retrieve the source code. We were able to get that all setup, but the documentation was so lacking that we were unable to progress any further, despite the GUI front-end.

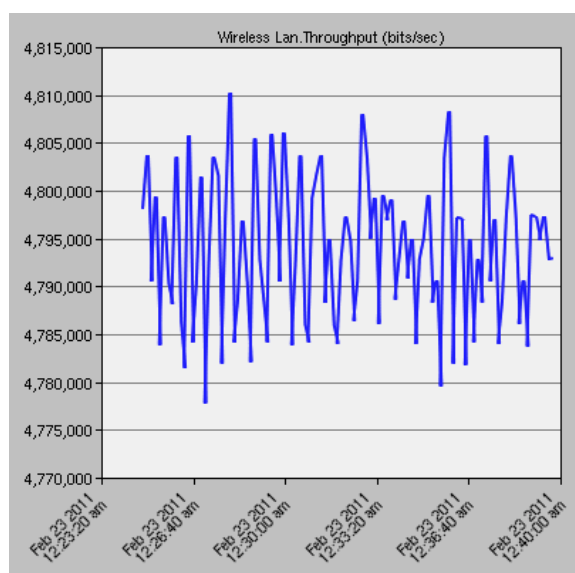


Figure 5.2 OPNET Modeler's 802.11b Model.

5.2.6 Emulab

Emulab [22] is a network of facilities worldwide that combine both hardware and software components. The only reason this simulator was even considered was because of the hardware. It would have been great to have been able to implement the scheme directly on hardware instead of just simulation. Although it would have been nice to have been able to go directly to hardware, we quickly realized that an important step was being skipped in the process. Therefore, the idea of Emulab was quickly abandoned. However, future work and researchers may find Emulab a very suitable testbed for hardware application.

5.3 OPNET Modeler: The Simulator of Choice

At the end of the day, a decision had to be made regarding which simulator to use. For the most part, the criteria listed in Table 5.1 were the driving force. However, we did rely on other factors such as ease of use and overall capability. We looked at the fact that some of the simulators were not capable of performing part of all of the experiment. The truth of the matter is that there is not one sole simulator out there that exists (to our best knowledge) that will perform all the tasks outlined. A simulator was needed that could perform the functions

of both the MAC layer and the PHY layer.

One more benefit of using OPNET was the fact that it is not an open source product, but a COTS product. Some may argue that open source may provide users with a large and vast community of users to ask questions and draw knowledge from. However, in this case, it is important to note that being a COTS product of this nature, it comes with technical support that is maintained contractually by the university. And because OPNET supports user contributions to their product in the form of various models, the open source type community is not completely lost.

5.3.1 The Best Choice

To this end, the decision was made to press forward with OPNET Modeler. OPNET was chosen primarily for the fact that it modeled not only the MAC layer but also the PHY layer. It also implemented a GUI front-end that allows for dragging objects from the object palette to the work area as seen in Figure 5.3. This makes setting up simulations quick and easy. It also came with a built-in IEEE 802.11g model.

5.3.1.1 Building the Model

OPNET is hosted on a number of university Linux servers. Accessing OPNET is as easy as opening a secure shell (ssh) session to one of the OPNET servers. OPNET relies on the x-forwarding technology to bring windows to your local machine. This requires an x-forwarding client like X11 or CygWin on the local computer. Typing “opnet” at the command line will bring that window to the local machine. Since OPNET comes with a built in IEEE 802.11g model (Figure 5.4), setting up the control network was quite trivial. Once the AP and one STA was configured, building the network larger was as easy as copy and paste.

As was discussed, setting up the control IEEE 802.11g network was rather trivial and can be done in a matter of minutes. The proposed DS-CDMA network is a whole different story. Since there was not a built-in DS-CDMA model within OPNET Modeler, we were left with a few options. 1) Search the user contributed models for an appropriate model. 2) Build our

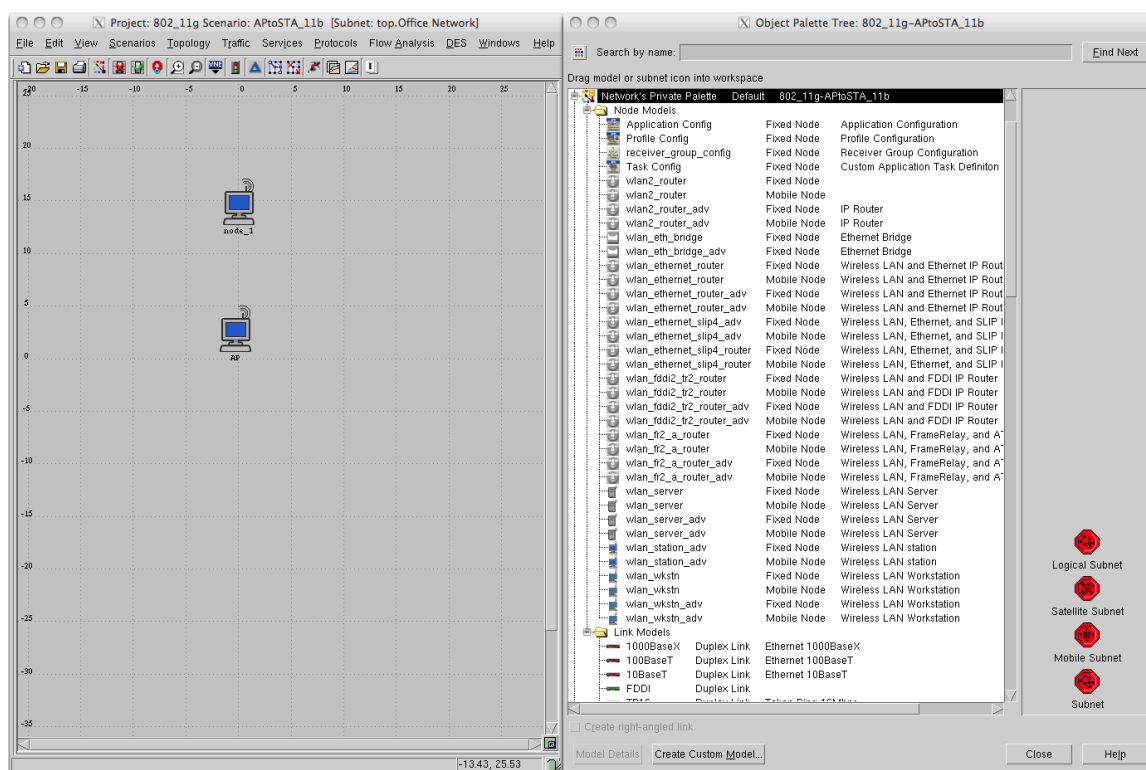


Figure 5.3 OPNET Modeler's GUI.

own DS-CDMA model. 3) Modify an existing model that closely resembles DS-CDMA.

Contributed Models We searched the internet extensively for an appropriate DS-CDMA model. In the end were only able to find one CDMA model. This model had been designed in an older version of OPNET Modeler and is no longer compatible with the current version that the university employs. Therefore, a case with OPNET Technical Support was opened to verify the fact that no model existed and it was just being missed. Technical support responded saying that they do not support CDMA or GSM as built in models. They referred us to the TDMA model [23].

Building Models Short of having an appropriate model to use, the next idea was to build a model. However, the way OPNET works is by building multiple models upon each other. The basic model is the packet format. The packet model outlines what the actual data packet looks like. Each field, such as the source and destination addresses, type, CRC, and data

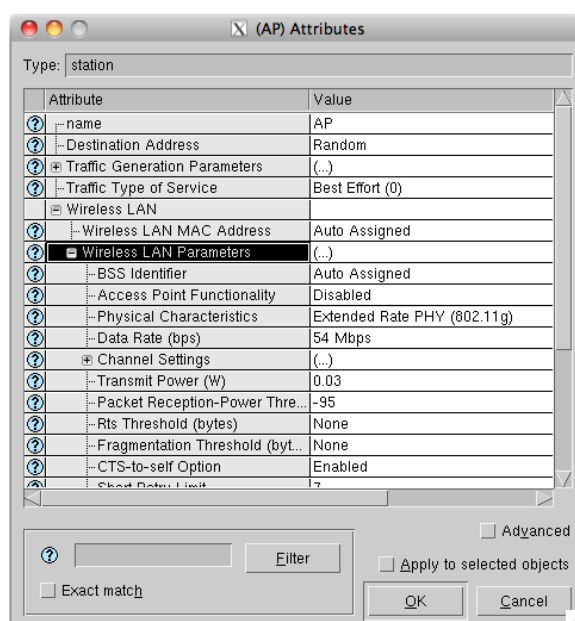


Figure 5.4 Options to Setup IEEE 802.11g Models.

are outlined in the packet format. Another model is the process model. Inside process models, instructions are given so that decisions can be made depending upon the state and the actions that are acting upon it. The process model is more or less the brains of the entire model. The final model is the node model. Node models are what users see when they drag icons into the OPNET Modeler workspace. Node models are made up of many different process and packet models that determine how the node model is going to act.

Building a packet format would be trivial, and in this case we could use the existing packet format and modify it just a bit. The difficulty in building a CDMA model comes in building the process models. Great care must be taken in order to account for every action that occurs within a data transaction. Items like calculating at what receive power is it a packet or just noise? What happens when a packet arrives from the higher or lower layers? Application of the PN codes. Every little detail must be accounted for. Building a CDMA model from scratch would not be a simple task and would probably take degrees in Computer Science and Electrical Engineering to complete properly. Since neither of these were at hand, the third option looked tantalizing.

Modify Models Modifying existing models within OPNET Modeler seemed like a task that would be able to be accomplished. The idea is simple. Choose the model that most closely resembles the proposed scheme, get to the source code, modify the source code to the specifications desired, and recompile. While all this sounds simple, in actuality, it turns out that it is not. We will discuss the trials encountered in modifying models in the next section.

5.3.2 Modifying the Target Model

In theory, the idea of modifying an existing model should be relatively simple to do. As long as there is a plan in place for the items that need to be changed in order to accomplish the goal, modification should be straightforward. However, we soon found out that *nothing* inside of OPNET Modeler is straightforward.

The existing model that most closely resembled DS-CDMA was the IEEE 802.11b model. This is because IEEE 802.11b is built upon a foundation of Direct Sequence Spread-Spectrum (DSSS). DSSS is a form of CDMA. Since IEEE 802.11b uses Barker codes and Complimentary Code Keying (CCK) to spread the data across the spectrum, IEEE 802.11b seemed like it would be a nice fit. The plan was to disable the CSMA/CA functionality that occurs in the MAC layer. To do this, the idea was to turn the SIFS, DIFS, slot, CW_{min} , CW_{max} , and the backoff window values all to zero. The effect of this would be that no packet would ever wait in the MAC layer and be immediately processed to the PHY layer, thus bypassing the CSMA/CA functionality.

The place where this is done is within the function block of the process model named “wlan_mac.pr.m.” The extension “pr.m” identifies this as a process model. A node model would carry the extension of “nd.m.” As previously discussed, the process model is where all the decisions are made. Figure 5.5 depicts the intricacies of the “wlan_mac” process model. Each circle represents a finite state machine (FSM) within the process model. When an FSM is invoked, it will move along the lines, called transitions, to the next FSM and wait for the next commands. Inside each FSM are lines of code. This code is written in proto-c and can be difficult to follow and understand at times.

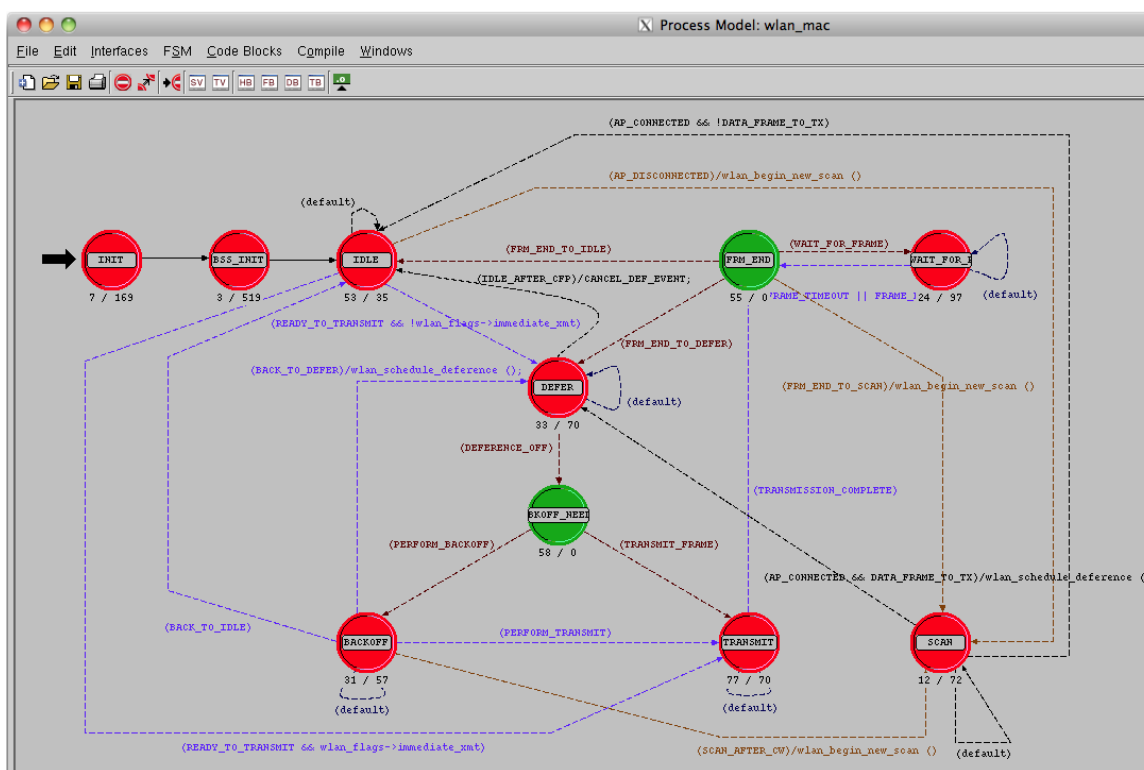


Figure 5.5 OPNET wlan_mac Process Model.

One thing that was made apparent quite quickly in this whole process was that the entire library on the OPNET server was locked to be read-only. While this is probably good practice from an enterprise standpoint, it was hindering our ability to modify the necessary code. Copies could be made, however, it created a “snowball” effect. Even though we may have copied the file to another one i.e. “wlan_CDMA_mac.pr.m,” the node file was also write protected. So there was no way to call the new filename from the node model file once it was copied.

The solution was to contact the Computer Services Group (CSG). They quickly created a virtual machine (VM) to use that had OPNET Modeler installed on it. With full read and full write privileges, we were able to make the changes to the original process model files. This would then allow the same unmodified node model to access the same process model file without any further consequences.

The following code snippet comes from the “wlan_mac.pr.m” file that was modified. This will be invoked when the IEEE 802.11b model with Direct Sequence is selected. This selection

would be made in Figure 5.4 under the Physical Characteristics pull-down menu. Note that if a STA were to be configured with IEEE 802.11g, this code would not be invoked as its durations are configured in another case. The nice thing about this setup is that in theory, only one simulation network would need to be setup. To gather simulation results from the other protocol, simply switch the nodes to the other protocol and run the simulation again. This would allow the nodes to be in the exact same position and would diffuse any question as to any discrepancies with the layout of the simulations.

```

case WlanC_Direct_Sequence:
{
/* Slot duration in terms of seconds. */
slot_time = 00E-06;
/* Short interframe gap in terms of seconds. */
sifs_time = 00E-06;
/* PLCP overheads, which include the preamble and header, in */
/* terms of seconds. */
plcp_overhead_control = WLANC_PLCP_OVERHEAD_DSSS_LONG;
plcp_overhead_data    = WLANC_PLCP_OVERHEAD_DSSS_LONG;
/* Minimum contention window size for selecting backoff slots. */
cw_min = 0;
/* Maximum contention window size for selecting backoff slots. */
cw_max = 0;
/* Set the PHY standard as 11b for the technologies specified */
/* in 802.11 and 802.11b. */
phy_type = WlanC_11b_PHY;
break;
}

```

Clearly, the code sets the interframe space times to zero. The CW_{min} and CW_{max} times

are also accounted for. The only time that is missing is DIFS. DIFS is accounted for later in the code as: “difs_time = sifs_time + 2 * slot_time.” Since both the SIFS and slot times are set to zero, DIFS will also equal zero.

Simulations were run with the parameters in Table 5.2 using both the default values for the above code as well as modified values. However, no matter what was attempted, the simulation results always remained the same. The code is recompiled and produces another “wlan_mac.pr.m” file that has a new timestamp. During the simulation, OPNET declares this same file as a dependency. But for some unknown reason, it kept using the original values and produces the same simulation results over again as can be seen in Figure 5.6. After such simulation, one would expect the values in the code to be back to the original default values based on the results. However, all the values remained set to the modified values.

Table 5.2 Simulation Parameters for Initial Run.

Parameter	Value
Protocol	IEEE 802.11b
Data Rate	11Mbps
Offered Load	11Mbps
Interarrival Time	.00174545 Sec.
Packet Size	2400 Bytes
MTU Size	2304 Bytes
Number of STAs	1
Simulation Duration	15 Min.

It was very interesting to evaluate the results from the simulations. One thing that was clearly apparent was the fact that the MAC layer seems to be very inefficient. Figure 5.7 shows that with a constant 11Mbps source sending traffic to the MAC (offered load), we noticed that only 6.281Mbps were actually being sent by the AP. This means that nearly 5Mbps is getting dropped by the MAC layer. With all the overhead involved, we witnessed an overall network throughput of 4.793Mbps — approximately 43.5% of the advertised 11Mbps.

Parsing through the source code for “wlan_mac.pr.m” is no small chore. Just this one file contains 10,256 lines of code. This fact alone was reason enough to decide to modify code instead of create new models. We feel that the OPNET IEEE 802.11 model is very thorough

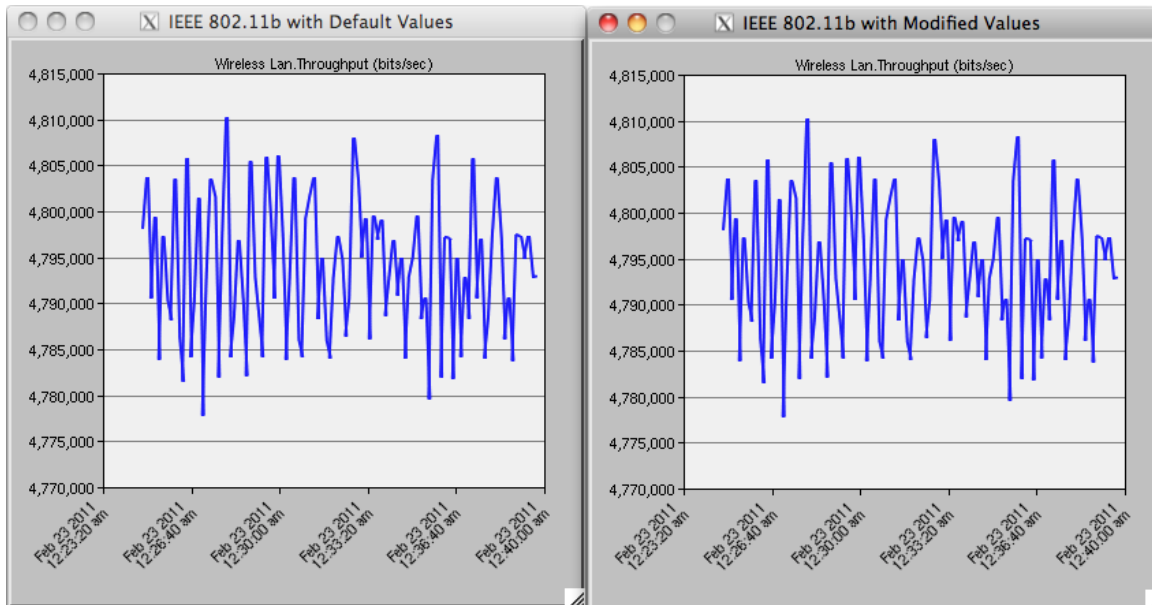


Figure 5.6 Simulation Results Using Default and Modified Values.

and complete.

Had the simulator accepted the parameters correctly, there would have been much more work to be completed. In the strictest sense of the definition, DSSS is not considered CDMA when it is used in Wi-Fi [24]. This is because in CDMA, unique codes are assigned to individual stations. In IEEE 802.11b/g, only one code is used for the entire system. Therefore, our scheme implements a system in which unique codes could be assigned to each STA. We present the following solution to this problem.

Upon association with the AP, a STA would enter a static generic code to authenticate with the AP. The AP would then associate the STA and give the STA a unique code to use. This code would then be placed somewhere in the beginning of the “wlan_mac” packet as in Figure 5.8. We have chosen a length of 10 bits to keep it consistent with the initial analysis.

Additionally, arrays and buffers would need to be set up on the AP to accommodate multiple requests simultaneously. Data would need to be sequentially XOR'd against all the codes that have been assigned. When data arrives, it would be despread by XORing using code C_i . The AP would then take the first 10 bits of the despread data and XOR it again with the code, C_i . If this resultant is zero, then it is a match and the data came from STA_i . If not, it came from

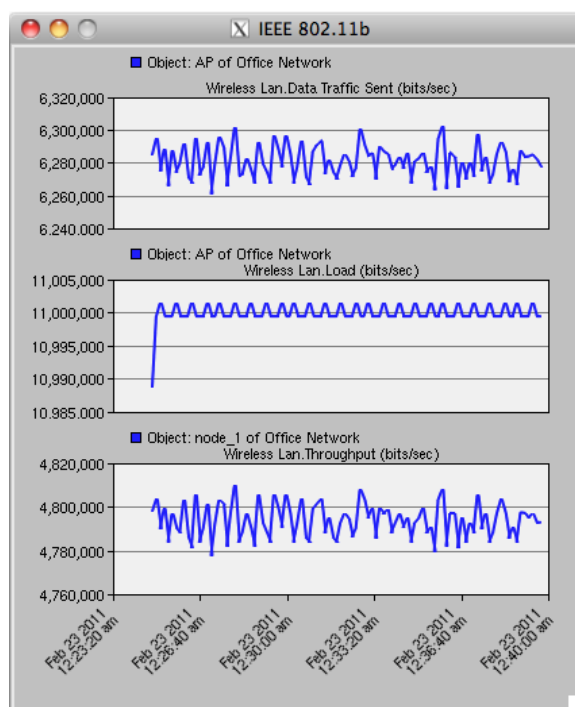


Figure 5.7 Throughput, Offered Load, and Data Sent.

another STA, so try C_{i+1} through C_n until there is a match. Similarly, on the STA side, the STA will XOR any data received with its unique code. If it matches, it forwards the data to the higher layers. Otherwise it discards the data.

Another item of the proposed scheme is the PHY channel. In IEEE 802.11, there is only one channel that is being used in a network at any given time. This channel is 22MHz wide. In the instrument, scientific, and medical (ISM) 2.4 GHz band, there are eleven channels (1 thru 11) available in the United States. To provide the maximum efficiency with the least interference,

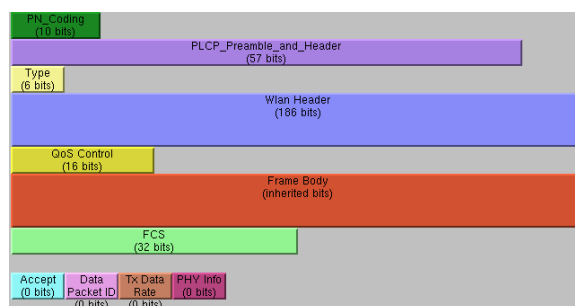


Figure 5.8 Proposed Packet Gormat.

there are three non-overlapping channels — 1, 6, and 11. Since an AP is only programmed to one channel at a time, it is sufficient to say that the maximum channel width of any network would be 22 MHz. The same channel can be used for both the transmit and receive because the MAC layer prevents multiple STAs from transmitting at the same time.

Staggered transmit and receive are not the case in CDMA. Different transmit and receive frequencies are required. Without separating the two, a transmitted signal would go directly into the receive side and completely drown out any other received signal at a particular station. However, when they are separated, a filter can be placed on the receive side to filter out the frequency of the transmit side of that node. Once the filtering problem is taken care of, one can concentrate on designing an appropriate channel for the scheme. In order to maintain consistency, we will assume the total channel size is 20 MHz — 10 MHz transmit and 10 MHz receive as was discussed in Chapter 4. However, since the entire ISM band is unlicensed, it is conceivable that all 83.5 MHz could be used making it into 41 MHz transmit and a 41 MHz receive channels.¹

The next item of concern are the receivers. We use the word *concern* here because there is much confusion when dealing with DS-CDMA. Most people are familiar and can relate to the concepts of FH-CDMA because it makes more logical sense. Add into the mix the idea that DS-CDMA can be both slotted and unslotted, and it gets even more convoluted. Slotted means that the data is synchronized (in time) between the STAs. When it is unslotted, data is not choreographed between all the STAs, leading to data being transmitted and arriving at different times. Differentiating between the different data streams becomes tricky, but it is not at all impossible. Because the concept of data reception in unslotted DS-CDMA may seem loose and abstract, most people will automatically revert to something that they know and are comfortable with — something slotted or synchronous.

The crux here is that in the IEEE 802.11g model, there is no reason to worry about carefully crafted receivers. This is because the MAC layer prevents multiple STAs from transmitting at the same time. If more than one STA transmits at the same time, the packets collide and

¹The FCC does place restrictions on spread spectrum operations that would limit the use of such channel sizes. Special authorization from the FCC would have to be granted.

the data is dropped — for which there are mechanisms in place to recover from. However, in a system that is designed so that multiple STAs transmit on the same channel, the difficult problem becomes receiving and differentiating multiple data streams. Add to this the fact that this scheme is an asynchronous system, meaning that STAs are not synchronized in their transmissions, and things get very complicated.

One common technique is to use a Rake receiver. A Rake receiver looks at multiple instances of the same data to separate it out from other data. According to [25], a Rake receiver “searches through the different multipath delays for code correlation and thus recovers delayed signals that are then optimally combined with the output of other independent correlators.” Figure 5.9 clearly shows how this can occur.

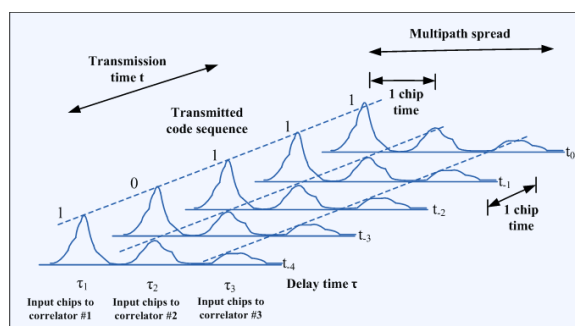


Figure 5.9 Detection Using Rake Receiver.

Differentiating between multiple sources is made possible by the fact that each STA will be received with different power from other STAs. This is due to multiple factors to include distance, transmitting power, and other noise on the channel. By passing the received data through Rake receivers and correlators, individual user data will be able to be extracted. Figure 5.10 [26] gives an example of how a user data is extracted among multiple other users on the same channel.

All of these changes would need to be made within the OPNET “wlan_mac.pr.m” process model. This is clearly a daunting task, especially when contemplating the idea of creating multiple receiver arrays, multiple channels, and a Rake receiver. However, without the ability to directly modify the code within OPNET, it was just not possible to accomplish.

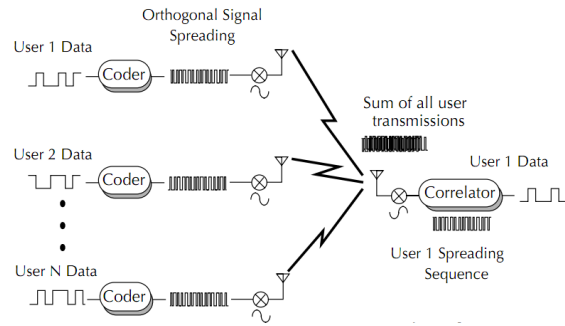


Figure 5.10 Correlation of User Data.

For the few aforementioned issues and items, we can clearly see why a CDMA simulator is not readily available. Not only are there many different “flavors” of CDMA, but there are many different applications of it as well. Once a flavor and application has been decided upon, a simulator can be crafted to meet the needs of that particular application. A generic or one-size-fits all model would quickly grow into something that would be extremely difficult to manage. Our particular application of DS-CDMA has not been attempted in a simulation environment to the best of our knowledge. With the shortage of related work, we believe we have documented as to the best model course of action. Should a model ever be designed, one could use this as a basis for the model.

5.3.3 The Quest

In an effort to locate a suitable simulator and model, many different entities were contacted. Since there are many organizations that rely on CDMA as part of their business model, surely, there had to be someone out there that had a CDMA simulator. We did not specify a software or hardware simulator, although we preferred software. This would have enabled us to integrate it with the IEEE 802.11 results that we had already obtained. The appendix contains a list of the persons that were contacted, the organization represented, and their response plus any qualifying comments.

As can be seen, there are some pretty large names and organizations on the above referenced list. However, in the end we were still left without an appropriate simulator or simulation model

to use. It is difficult to reason that a CDMA simulator does not exist. However, we were unable to locate one. Just because a model was not found and we were not able to simulate properly does not mean all is lost. By analyzing the initial analysis, one can see that it is sufficiently good enough to merit further investigation.

CHAPTER 6. LIMITATIONS AND KNOWN ISSUES

As with any system, there are limitations. This is because unfortunately, no system is ideal. This scheme is not without its own flaws. The first being interference. While spread spectrum actually reduces interference due to a reduction in signal-to-noise ratio, there may be other interference injected. Depending on the saturation and proximity of users, the amount of noise may increase. This noise is sometimes referred to as multiple access interference (MAI) or multiple user interference (MUI) [10].

Along with this interference is a phenomenon called the near-far problem. The near-far problem results from the AP not being able to effectively control its power. Because the AP needs to be heard at the furthest STA, those STAs close to the AP may get drowned out by an overly strong AP transmission [11].

Another issue that is foreseen is running out of bandwidth. Generalities were taken in sections 3.1 and 3.2 in describing the functionality of DS-CDMA. However, it must be noted that bandwidth usage is additive. For example, if in Table 3.1 the resulting transmissions would have been 0100 and 0110, there would be a total bandwidth of 2 during $T=2$. A good visual reference is given in Figure 1.6 in [8].

At some point, the medium will be so saturated, that recognizable transmissions and data will not be able to be distinguished. When this occurs, the network will become essentially useless. Mechanisms will need to be put in place to prevent such an occurrence from happening.

The final issue is the actual implementation. Because the MAC layer will need to be modified, there is currently no hardware that can readily adapt this scheme. With this modification to the MAC layer, there is a good chance that this scheme will no longer be classified under the IEEE 802.11 umbrella. This is fine, as it is of the opinion of the author that ideas like this

should not be constrained within one family of protocols.

CHAPTER 7. RELATED AND FUTURE WORK

7.1 Related Work

Identifying related work can be tricky and confusing. If one were to try to identify works on IEEE 802.11, the list would be extremely long. Limiting the list to only those that were trying to eliminate or reduce contention in IEEE 802.11 and the list becomes shorter, but still remains ridiculously long. Work relating to CDMA, and more specifically to DS-CDMA, is rather short.

The novelty of this work is based on the application of DS-CDMA to IEEE 802.11. Currently, the author knows of no real work to get DS-CDMA implemented into the IEEE 802.11 scheme. References [4] and [28] attempt to look at the issue, however, both are utilizing MC-CDMA. In both of these references, the IEEE 802.11 DCF and MAC layer are preserved. A work that is based on applying orthogonal codes to PCF is here [29].

Hsiao-Hwa Chen presents a very compelling use of CDMA in [30]. This recent work has to do with next generation CDMA or NG-CDMA. NG typically references Gigabit speeds. Although this is also typically seen in the cell phone arena, the application is very applicable in the Wi-Fi realm. His work shows that “orthogonal complementary code sets could offer MAI-free and MI-free [MUI-free] operation for DS-spreading modulation.” Furthermore, finding better and more acceptable codes and coding techniques is certainly something for future work.

7.2 Future Work

Due to the characteristics of DS-CDMA, the growth potential of this protocol is nearly limitless. This is why future work is essential for the future of wireless communications. First,

simulation using a commercial or community accepted and validated simulator would prove to be beneficial. If the simulations come back favoring DS-CDMA, then pursuing an actual implementation would be feasible. Next, would be implementing this protocol in hardware and software. This is to test the real-life feasibility of the protocol.

This framework would be a great addition to crowded and overloaded wireless networks. Dense populations of wireless nodes prove to be a problem in that there is more probability for contention. Actual real-life data to back up the simulation would further prove the feasibility of the DS-CDMA protocol.

DS-CDMA would also be a good candidate for use in a wireless lab. Because of the limited interference with nearby STAs, students could invoke multiple virtual machines with actual physical wireless connectivity.

MC-CDMA inherently contains built-in security due to its frequency hopping nature. Since DS-CDMA is spread across the entire spectrum of the channel and not hopping between frequencies, security may be lacking. Granted, DS-CDMA utilizes spread spectrum to invoke a sense of security. This security model, used by the military years ago, is a great example of “security by obscurity.” It has been shown over and over that this model does not provide a good enough security blanket. The transmissions will appear as noise to listening stations. Since the PN codes repeat, cracking the PN code to get to the underlying data will be trivial. Therefore, security will need to be looked at in greater depth and built into the protocol.

Also, to bring this more in line with IEEE 802.16, QoS needs to be addressed. The QoS should not be an issue between STAs, as a direct link protocol (DLP) provision is not included in the proposed scheme. However, addressing the issue of QoS to deal with prioritized traffic within each STA is key.

Finally, investigation into greater bandwidth and different modulations schemes should be investigated. Since our scheme utilizes DS-CDMA, STAs can transmit simultaneously. Operating in the 2.4 GHz ISM band, there is 83.5 MHz available for use. This means that each STA could potentially utilize 40+ MHz on both the transmit and receive sides. Additionally, if this scheme were taken to the 5.8 GHz ISM band [32], there would be 200 MHz available,

making two 100 MHz channels. If it was not necessary to have equal upload and download speeds, one could conceivably create a 200 MHz receive channel using the upper and middle bands and a 100 MHz transmit channel using the upper band. To sweeten the process even more, modulation techniques may push the throughput even higher. Looking at 256-QAM or even higher modulation techniques might be a possibility.

CHAPTER 8. CONCLUSION

We have introduced a novel scheme to eliminate contention that exists within the IEEE 802.11 framework. Although DS-CDMA is not new, it has not been implemented in this way before. In doing so, contention will be eliminated and STAs will be able to freely transmit by the use of unique PN codes. We have described the major benefits to moving to said scheme to include sustained throughput especially as the number of users increase. Through theoretical analysis, we have shown that a significant increase in sustained throughput can be realized over the IEEE 802.11g scheme by implementing the proposed DS-CDMA scheme. We also listed some drawbacks that would need to be addressed or mitigated as well as some future work that could be done to enhance this scheme.

APPENDIX A. ADDITIONAL CONTACTS

Functional Area 24 Mailing List — U. S. Army — We posed the original question of a CDMA simulator to this mailing list. On this list are 342 Army Telecommunication Systems Engineers. In order to become a Telecommunications Systems Engineer in the Army, one must first possess a Bachelors Degree in a hard science as well as attend the Army's Telecommunications Systems Engineering Course (TSEC). TSEC is taught by hand-selected PhDs from around the nations Universities and has one of the highest attrition rates of any Army school. Only one person stepped up to the challenge (See MAJ John H. below).

LTC Brion J. — Cisco Systems, Inc. — LTC J. forwarded the request to an internal mailing list at Cisco Systems, Inc. This list was comprised of many engineers. Only one response came back. However, the person was confused between CDMA and CSMA and his comments were discarded.

Q-Lab — Qualcomm — Since Qualcomm holds the patent on CDMA, it was only fitting to go to the source. Qualcomm has their own internal lab that provides testing for businesses called Q-Lab. Due to the size of Qualcomm, we were not able to get in contact with the Q-Lab. As a matter of fact, after numerous calls to their headquarters, we were discouraged because nobody at Qualcomm knew what the Q-Lab was, nor were they able to get us in contact with anybody who knew of the Q-Lab.

Braden K. — Verizon Wireless — Although Braden is not a technician of any kind, he was unable to provide any contact information with anyone who was either.

Heather L. — Formerly of Verizon — Heather and her husband Tim worked for Verizon for many years as network engineers. The were not able to obtain any kind of simulator, nor did they know of any.

Hock S. — OPNET Technologies, Inc. — Hock responded to the trouble ticket that was submitted in reference to a CDMA model within OPNET Modeler. He responded by saying that OPNET does not support CDMA as a built in model. However, he directed to the user contributed models which contain only one model that no longer works with the current version of OPNET.

Dean A. — NS-3 Contributor — Dean provided much insight into the intricacies and difficulties of attempting to model CDMA within NS-3. In the end, he said that it would be possible, however “significant functionality” would need to be added to make it work.

Bo B. — Cisco Systems, Inc. — Bo recommended using OPNET Modeler for the IEEE 802.11g portion and using Matlab’s Simulink for the CDMA. For reasons discussed elsewhere in this paper, this solution was not pursued.

Ross O. — RaySat — Provided no useful information toward a successful simulation model.

Dr. Taieb Z. — University of Pittsburgh — Dr. Z. directed to Dr. William Osborne of Southern Illinois University. Dr. Osborne had written a paper entitled *OPNET Simulation of Random Access CDMA Networks* a few years back. He also contemplated any other solutions to the problem posed as he too has done work in this area in the past.

Dr. William Osborne — Southern Illinois University — Dr. Osborne has since retired. However, he did refer to the paper’s co-author, Dr. Murat Torlak of University of Texas at Dallas.

Dr. Murat Torlak — University of Texas at Dallas — Dr. Torlak offered up the only CDMA model found. However, it was designed in an older version of OPNET and no longer works with the current version.

MAJ John H. — U. S. Army / University of Colorado at Boulder — Although MAJ H. did not have a solution, he provide many questions and answers to questions for this research. He also posed the same questions to some of his instructors at the University of Colorado at Boulder. In an email [27], he shared the following:

We had a guest speaker yesterday in the Wireless LAN class that’s in the Wi-Fi

Alliance and IEEE 802.11 standards group. I asked him if there was any place for CDMA within 802.11 and he kinda chuckled. He said that there's such a mindset of contention-based solutions within 802.11 that many people can't even conceive of a shared medium solution like CDMA.

BIBLIOGRAPHY

- [1] Golden, P., Dedieu, H., & Jacobsen, K. (2005). *Fundamentals of DSL technology*. Boca Raton, FL: Auerbach Publications.
- [2] Dixon, R. C. (1994). *Spread spectrum systems with commercial applications, 3rd ed.* New York, NY: John Wiley & Sons.
- [3] Freeman, R. (2004). *Telecommunication system engineering, 4th ed.* New York, NY: John Wiley & Sons.
- [4] Orfanos, G., Habetha, J., & Liu, L. (2004). MC-CDMA based IEEE 802.11 wireless LAN, *Proceedings of The IEEE Computer Societys 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MAS-COTS04)*.
- [5] *802.11-2007*. (2007, June 12). Retrieved from <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>
- [6] Thompson, R., Tipper, D., Krishnamurthy, P., & Kabara, J. (2006). *The physical layer of communications systems*, Norwood, MA: Artech House.
- [7] Schulze, H. and Luders, C. (2005). *Theory and applications of OFDM and CDMA wideband wireless communications*. West Sussex, England: John Wiley & Sons.
- [8] Zigangirov, K. (1994). *Theory of code division multiple access communication*. New York, NY: John Wiley & Sons.
- [9] *802.11b wifi channels*. (2011). Retrieved from <http://www.moonblinkwifi.com/2point4freq.cfm>

- [10] *CDMA*. (2011). Retrieved from <http://www.answers.com/topic/cdma>
- [11] Saha, A. (n.d.). *Spread spectrum, CDMA and GPS*. Retrieved from http://alumni.cs.ucr.edu/saha/stuff/cdma_gps.htm
- [12] Wang, Hao-Li (2004). MAC-layer approaches for security and performance enhancement in IEEE 802.11. Ph.D. dissertation, Iowa State University, United States – Iowa. Retrieved November 27, 2010, from Dissertations & Theses @ Iowa State University.(Publication No. AAT 3136354).
- [13] Sheu, S.-T., Sheu, T.-F., Wu, C.-C., & Luo, J.-Y. (2001). Design and implementation of a reservation-based MAC protocol for voice/data over IEEE 802.11 ad-hoc wireless networks. *Proceedings of IEEE ICC01, International Conference on Communications*.
- [14] Grote, A., Grote, W., Feick, R., in *IFIP International Federation for Information Processing*, Volume 256, Home Networking, Al Agha, K., Carcelle, X., Pnjolle, G., (Boston: Springer), pp. 15-28.
- [15] *System design and verification*. (n.d.) Retrieved from <http://www.cadence.com/products/sd/Pages/default.aspx>
- [16] *The network simulator - ns-2*. (n.d.). Retrieved from <http://www.isi.edu/nsnam/ns/>
- [17] *The ns-3 network simulator*. (2011, April 01). Retrieved from <http://www.nsnam.org/>
- [18] Plass, Bryan. “NS3.” Message to Dean Anderson. 04 Feb 2011. E-mail.
- [19] *OPNET*. (2011). Retrieved from <http://www.opnet.com/>
- [20] *Simulink - Simulation and Model-Based Design*. (2011). Retrieved from <http://www.mathworks.com/products/simulink/>
- [21] Arby, D. (2008). *ShoX: a scalable ad hoc network simulator*. Retrieved from <http://shox.sourceforge.net/>

- [22] University of Utah. (2011). *Emulab - network emulation testbed home*. Retrieved from <http://www.emulab.net/>
- [23] OPNET Technical Support. "is there a CDMA within modeler (TSC 246072)." Message to Bryan Plass. 21 Feb 2011. E-mail.
- [24] Walke, B., Mangold, S., & Berlemann, L. (2006). *IEEE 802 Wireless Systems: Protocols, Multi-hop Mesh/Relaying, Performance and Spectrum Coexistence*. West Sussex, England: John Wiley & Sons.
- [25] Nguyen-Le, S. (2007, November 17). *The rake receiver applied to direct-sequence spread-spectrum (DS/SS) systems*. Retrieved from <http://cnx.org/content/m15534/latest/>
- [26] Sheng, S. (1999). Signal processing in communications II: CDMA. *IEEE Technical Committee on Microprocessors and Microcomputers, Symposium on High Performance Chips*.
- [27] Plass, Bryan. "Simulator." Message to John Holmes. 26 Feb 2011. E-mail.
- [28] Ng, P.C., Edwards, D.J., & Liew, S.C. (2007). Eliminating inter-bss co-channel interference by mc-cdma in wlans. *Proceedings of the wireless communications and networking conference, 2007.WCNC 2007. IEEE*, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4224650 10.1109/WCNC.2007.405 .
- [29] Yoon, D. *Simultaneous Packet Reservation via Orthogonal Signaling in IEEE 802.11 WLAN*.
- [30] Sasan, A., Mobashar, A., & Tofighbakhsh, M. (2010). *Fourth-Generation Wireless Networks Applications and Innovations*. Hershey, PA: IGI Global.

- [31] Xiao, Y. (2003). MAC Performance Analysis and Enhancement Over 1000Mbps Data Rates for IEEE 802.11. *Proceedings of the vehicular technology conference, 2003. VTC 2003-Fall. 2003 IEEE 58th* , <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1285349&isnumber=28570> 10.1109/VETEFCF.2003.1285349 .
- [32] *IEEE std 802.11a-1999(R2003)*. (2003, June 12). Retrieved from <http://standards.ieee.org/getieee802/download/802.11a-1999.pdf>
- [33] Ho, M.-J., Wang, J., Shelby, K., & Haisch, H. , "IEEE 802.11g OFDM WLAN throughput performance," *Proceedings of the vehicular technology conference, 2003. VTC 2003-Fall. 2003 IEEE 58th* , <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1285930&isnumber=28571> 10.1109/VETEFCF.2003.1285930 .
- [34] Gast, M. (2005). *802.11 Wireless Networks: The Definitive Guide, Second Edition*. Sebastopol, CA: O'Reilly.
- [35] *802.11n: the next generation of wireless performance*. (2011). Retrieved from http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns348/ns767/white_paper_c11-427843_v1.pdf
- [36] Hall, R.J. (2011). IEEE 802.11. *Wikipedia: the free encyclopedia*. Retrieved April 1, 2011, from http://en.wikipedia.org/wiki/IEEE_802.11