# Quantifying Tradeoffs in the IEEE 802.15.4 protocol through simulation

**Submitted To**

**Pierre Collinet**
**Dr. Brian L. Evans**

**Prepared By**

**Chinmoy Gavini**

**EE464 Senior Design Project**
**Electrical and Computer Engineering Department**
**University of Texas at Austin**

**Summer 2007**

CONTENTS

TABLES ..... iv

FIGURES ..... vi

EXECUTIVE SUMMARY ..... v

1.0 INTRODUCTION ..... 1

2.0 DESIGN PROBLEM STATEMENT ..... 2

3.0 DESIGN PROBLEM SOLUTION ..... 2

3.1 BACKGROUND INFORMATION ..... 3

3.1.1 DBPSK Modulation ..... 3

3.1.2 OQPSK Modulation ..... 4

3.1.3 Direct Sequence Spread Spectrum (DSSS) ..... 5

3.1.4 Channel Models ..... 5

3.2 DESIGN DECISIONS ..... 6

3.2.1 Rayleigh and Rician Models ..... 6

3.2.2 IEEE 802.15.4a Channel Model ..... 7

3.2.3 Timing recovery design decisions ..... 8

4.0 DESIGN IMPLEMENTATION ..... 8

5.0 TEST AND EVALUATION ..... 9

5.1 Bit Error Rate Analysis ..... 10

5.2 Data Rate and Implementation Complexity ..... 14

6.0 TIME AND COST CONSIDERATIONS ..... 15

7.0 SAFETY AND ETHICAL ASPECTS OF DESIGN ..... 15

8.0 CONCLUSIONS AND RECOMMENDATIONS ..... 16

REFERENCES ..... 17

APPENDIX A – GLOSSARY ..... A-1

APPENDIX B – BLOCK DIAGRAM ..... B-1

APPENDIX C – SIMULATION SOURCE CODE ..... C-1

## TABLES

1 Expected Characteristics of the IEEE 802.15.4 protocol ................................................ 2
2 BER Comparison of the OQPSK and BPSK system .................................................... 14

## FIGURES

1 Schematic representation of DBPSK modulation ........................................................ 6
2 BER plot of BPSK system with AWGN ........................................................ 10
3 BER plot of BPSK system with AWGN and channel model ........................................................ 11
4 BER plot of OQPSK system with AWGN ........................................................ 12
5 BER plot of OQPSK system with AWGN and channel model ........................................................ 13
6 IEEE 802.15.4 simulation block diagram ........................................................ B-2

## EXECUTIVE SUMMARY

The goal of this project is to quantify the tradeoffs in the modulation methods of the IEEE 802.15.4 wireless sensor network protocol through baseband simulation in Matlab. The protocol specifies the Differential Binary Phase Shift Keying (DBPSK) modulation for the 868 MHz and the 915 MHz carrier frequencies and the Offset Quadrature Phase Shift Keying (OQPSK) modulation for the 2.4 GHz frequency. The three Physical Layers use Direct Sequence Spread Spectrum (DSSS). The simulation consists of the transmitter, receiver, channel, and Bit Error Rate (BER) analysis modules. Symbol generation, DSSS spreading, modulation, pulse shaping, and transmission occur in the transmitter, while the receiver performs matched filtering, demodulation, and DSSS despreading. The channel module corrupts the signal with Additive White Gaussian Noise (AWGN) and attenuates the signal with the channel impulse response. Binary Phase Shift Keying (BPSK) is a modulation technique in which the phase of the carrier wave assumes one of two values depending on the bit to be sent. In DBPSK, the data stream is encoded into an intermediate bit stream based on the XOR result of the current unencoded bit and the previous encoded bit. The phase of the carrier wave changes if the $n^{th}$ encoded bit is different from the previously encoded bit. In Quadrature Phase Shift Keying (QPSK), the phase of the carrier wave assumes one of four values. In OQPSK, the in-phase and quadrature-phase channels are offset by ½ of a symbol period. DSSS is an encoding technique for counteracting noise and widening the frequency band of the transmitted signal. The transmitter maps or "spreads" the symbols into predefined PseudoNoise (PN) sequences before modulation. The receiver recovers the symbols after demodulation.

An appropriate channel model must account for multipath fading and noise because multipath propagation and noise are present in any wireless system. The three channel models I considered were the Rayleigh model, the Rician model, and the IEEE 802.15.4a channel modeling subgroup's model. An accurate channel model must include a channel response in terms of a "time-varying Finite Impulse Response (FIR) filter, a time-varying gain for fading, and AWGN" [2]. The IEEE 802.15.4a subgroup's model was optimal because it met all three constraints and generalized the Rician and Rayleigh models. I chose header correlation and filter delay compensation for timing recovery in the BPSK system and the OQPSK system, respectively. Because the IEEE 802.15.4 protocol is not meant to be used with channel equalization, I used guard times to counteract the effect of channel memory. The BER plots demonstrate that the OQPSK system has lower bit error rates over a range of Signal to Noise (SNR) ratio values from 0 dB to 20 dB in increments of two. The OQPSK system successfully transmitted 2560 bits per iteration compared to the BPSK system's 1000 bits per iteration. The OQPSK system offers twice the data rate of the BPSK system, but requires more memory and compute cycles. Because this project involved software simulation, there were no safety concerns. To thoroughly understand the protocol's safety and reliability, one would have to perform an RF simulation or implement the protocol on hardware. The project currently has no expenses. However, future expenses such as purchasing Matlab, the Matlab Communication Toolbox, and an IEEE 802.15.4 hardware board would total $3199 for a commercial project. The limitations of the project are its non real-time nature and exclusion of RF components. To address these limitations, one might implement an IEEE 802.15.4 system either on hardware or a Radio Frequency (RF) simulation.

# 1.0 INTRODUCTION

The purpose of this final report is to document the simulation design work, analyze the simulation results, and compare the Differential Binary Phase Shift Keying (DBPSK) and Offset Quadrature Phase Shift Keying (OQPSK) Physical Layers of the IEEE 802.15.4 protocol. Dr. Brian Evans, an Electrical and Computer Engineering (ECE) professor at the University of Texas at Austin, sponsored this project. Dr. Evans' research interests are embedded real-time digital signal processing systems such as communication systems and image processing. This design project has contributed to an understanding of the engineering tradeoffs in choosing a particular physical layer in the IEEE 802.15.4 protocol, a wireless telecommunications protocol for low power applications. Low power wireless communication is necessary in applications such as industrial sensors and medical sensors.

The report begins with the design problem statement and provides technical background information on modulation, DSSS spreading, and channel modeling. In the next section, I discuss the design decisions of the channel model, timing recovery, and DSSS despreading. After describing the design decisions, I explain the design modifications to the header correlation and the pulse shaping for the OQPSK simulation. The final section contains an analysis of the bit error rate, the data rate, the implementation complexity, and the expenses of future work. The report concludes with a summary of the project results and provides recommendations for future projects to implement the IEEE 802.15.4 protocol in an RF simulation or on hardware.

## 2.0 DESIGN PROBLEM STATEMENT

The goal of this project is to quantify the tradeoffs in the modulation methods of the IEEE 802.15.4 protocol through baseband simulation in Matlab. Offset Quadrature Phase Shift Keying (OQPSK) and Differential Binary Phase Shift Keying (DBPSK) are the modulation techniques in the protocol. Some examples of protocol applications are industrial sensors and portable medical sensors. BER plots, data throughput, implementation complexity, and verification against the IEEE 802.15.4 standard are the methods for quantifying the tradeoffs in this project. Verification of the results against the standard consists of verifying the bit rates and symbol rates against the following table:

**Table 1. Expected characteristics of the IEEE 802.15.4 protocol [1]**

| Band | Frequency Band | Bit Rate | Symbol Rate | Modulation | Chip Rate |
|---|---|---|---|---|---|
| 868 MHz | 868-868.6 MHz | 20 kb/s | 20 ksymbols/s | BPSK | 300 kchips/s |
| 916 MHz | 902-928 MHz | 40 kb/s | 40 ksymbols/s | BPSK | 600 kchips/s |
| 2.4 GHz | 2.4 – 2.4835 GHz | 250 kb/s | 62.5 ksymbols/s | O-QPSK | 2 Mchips/s |

An accurate channel model includes a channel response with a "time-varying FIR filter, a time-varying gain for fading, and Additive White Gaussian Noise (AWGN)" [2]. The simulation consists of the transmitter, the receiver, the channel, and the BER analysis modules. The transmitter includes symbol generation, DSSS spreading, DBPSK/OQPSK modulation, and pulse shaping. The channel effects consist of corrupting the signal with AWGN and attenuating the signal with the channel impulse response. The matched filter module, the BPSK/OQPSK demodulation, and the DSSS demapping constitute the receiver.

## 3.0 DESIGN PROBLEM SOLUTION

In order to implement the simulation of the IEEE 802.15.4 protocol, one has to understand the components inside the transmitter, receiver, and the channel. After studying the basic operating principles of the various components, one can choose the channel model and design algorithms for tasks such as DSSS despreading. The following sections contain technical background information and the design decisions for channel modeling, timing, and DSSS implementation. Please refer to Appendix A for a glossary of terms that are not covered in the following sections. Appendix B contains the simulation block diagram.

### 3.1 BACKGROUND INFORMATION

The IEEE 802.15.4 standard specifies DBPSK modulation for the 868 MHz and the 915 MHz Physical Layers. BPSK Modulation is a form of Phase Shift Keying (PSK). Phase shift keying is a modulation technique in which the phase of the carrier wave is modified based on what symbol is being sent. In BPSK, the symbols are the same as the bits 0 and 1. Therefore, one might send a pulse without modifying its phase to represent a 0 and send a pulse with the phase shifted by 180 degrees to represent a 1. The following section explains how DBPSK differs from ordinary BPSK.

#### 3.1.1 DBPSK Modulation

In Differential BPSK (DBPSK), the bits are modulated using the following calculation:

$$E_n = R_n \oplus E_{n-1} \tag{1}$$

In the equation, $E_n$ is the the nth bit to be differentially encoded, $R_n$ is the n-th raw bit, and $E_{n-1}$ is the previous encoded bit, and $E_0$ is 0. For example, in the data stream in Figure 1 on the next page, the first raw bit is 1 and $E_0$ is 0. Therefore, the first encoded bit is 1. The next raw bit is 1 and the result of the XOR operation is 0. Therefore, the second encoded bit is represented by the inverted pulse. Until the fifth computation, the XOR operation produces a 0 so the pulse representation is constant. The result of the sixth XOR operation results in a 1. The pulse shape inverts because 1 is different from the previous XOR result of 0. In the following sections, BPSK is interchangeable with DBPSK.

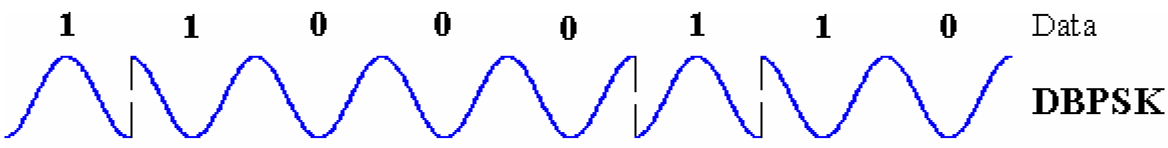


**Figure1. Schematic representation of DBPSK Modulation [3]**

Like BPSK, OQPSK is a type of phase modulation. However, OQPSK modulation can encode two bits at a time, as the following section explains:

### 3.1.2 OQSK Modulation

OQPSK modulation is the modulation method for the IEEE 802.15.4 2.4 GHz Physical Layer. While the phase of the carrier wave can be modified in one of two ways with BPSK modulation, with Offset Quadrature Phase Shift Keying (QPSK) modulation, the phase of the carrier wave can be changed in one of four ways to represent the symbols 00, 01, 10, and 11. The advantage of QPSK is that two bits can be represented per symbol unlike BPSK, which can only represent one bit per symbol. However, QPSK is more susceptible to noise than BPSK. Compared to a BPSK system, noise can cause a particular phase value to appear as the other phase values with a higher probability as the phase shift values are closer to each other in an OQPSK system. QPSK works because the sine wave and the cosine wave are orthogonal with respect to each other. The transmitter splits the data stream into the in-phase channel and the quadrature-phase channels. The in-phase channel is modulated onto a cosine wave and the quadrature-phase channel is modulated onto a sine wave. The transmitter adds the two modulated channels together before upconversion and transmission. OQPSK is a variation of QPSK in which the in-phase and quadrature-phase channels are offset in time by ½ of a symbol period.

DSSS is a technique used to encode the symbols in order to counteract the effects of noise and widen the frequency band of the transmitted signal. An unencoded symbol that is corrupted by noise might be undecipherable at the receiver. However, a symbol that is encoded with a PN sequence has a higher probability of being correctly deciphered, as the following section explains:

### 3.1.3 Direct Sequence Spread Spectrum (DSSS)

DSSS is a technique in which the symbols at the transmitter are replaced by PN sequences before modulation occurs. DSSS "spreads" or widens the frequency spectrum of the digital signal. One of the advantages of spreading the frequency band is to "lower the Power Spectral Density (PSD) of the original symbol and lower the probability that the signal interferes with foreign narrowband signals" [4].

In the IEEE 802.15.4 DBPSK system, the transmitter maps the symbol '0' to a 15-bit long PN sequence and the symbol '1' to a different 15-bit sequence. The PN sequence is input into the modulator before transmission. The receiver "despreads" the received sequence back into the original symbol. I implemented despreading in the simulation using the XOR operation. The process consists of applying the XOR operation on the incoming sequence with both PN sequences and adding the number of 1 bits in the result of the XOR operation. The PN sequence corresponding to the minimum sum is the most likely sequence that was transmitted because the XOR operation returns a binary '0' when both inputs are the same. In the OQPSK system, the transmitter maps the symbols 0000 to 1111 (the numbers 0 to 15) to sixteen 30-bit PN sequences. The despreading algorithm is essentially the same as the DBPSK system's despreading algorithm. The difference occurs in the number of PN sequences against which the receiver checks the incoming data.

Because a wireless signal propagates over free space, a channel model for the simulation must account for the attenuation of the signal as it propagates over physical obstacles. The following section discusses multi-path propagation and long distance fading as well as their effects on the channel impulse response.

### 3.1.4 Channel Models

Multi-path propagation and noise are part of any wireless communication system. Multi-path propagation refers to the delayed and attenuated copies of the transmitted signal that arrive at the receiver in addition to the original signal. Multi-path propagation consists of diffraction, reflection, and scattering. Because multi-path propagation varies probabilistically at each point in time, the impulse response of the channel varies with time. An accurate channel model's impulse response includes the fading due to multi-path propagation and attenuation of the signal over a long distance because the impulse response fully characterizes a linear system.

The major design decisions in the project involved the channel model design, the receiver timing recovery, and the DSSS implementation. The following section and subsections provide the rationale for choosing the IEEE 802.15.4a subgroup's channel model over alternative models. I also discuss the header correlation and delay compensation techniques for timing recovery.

## 3.2 DESIGN DECISIONS

For the channel model, I considered the Rayleigh model, the Rician model, and the channel model proposed by the IEEE 802.15.4a subgroup. The following sections contain more details of these channel models:

### 3.2.1 Rayleigh and Rician Models

The Rayleigh Model is a channel model that is exclusive to Non Line-of-Sight (NLOS) propagation components. The Rayleigh Model only models small-scale effects such as the time varying behavior of the channel, the frequency-selective fading, and the signal dispersion. The model does not consider large scale effects such as path loss over distance [5]. The Rician model is a general form of the Rayleigh model which addresses LOS components in addition to the NLOS components . The Rician K-factor is the ratio of the power of the LOS component to the average power of the NLOS components [5]. When $K = 0$, the Rician model converts to the Rayleigh model. Like the Rayleigh model, the Rician model does not address large-scale channel effects.

Unlike the Rayleigh and Rician models, the channel model proposed by the IEEE 802.15.4a channel modeling subgroup includes large scale effects and features specific to IEEE 802.15.4 systems. The following section describes the IEEE 802.15.4a channel model.

### 3.2.2 IEEE 802.15.4a channel model

The IEEE 802.15.4a channel modeling subgroup developed this channel model after studying IEEE 802.15.4 networks. The model is a general form of the Rician model using the Nakagami Probability Density Function (PDF). The Nakagami PDF is: [6]

$$pdf(x) = \frac{2}{\Gamma(m)}\left(\frac{m}{\Omega}\right)^{m} x^{2m-1}\exp\left(\frac{-m}{\Omega}x^{2}\right) \tag{2}$$

In the equation, $m$ is the Nakagami $m$-factor, $\Gamma(m)$ is the gamma function, and $\Omega$ is the mean-square value of the amplitude [6]. One can convert to and from a Rician model with these substitution equations [6]:

$$\mathrm{m} = \frac{(K+1)^{2}}{2K+1} \tag{3}$$

$$\mathrm{K} = \frac{\sqrt{m^{2}-m}}{m-\sqrt{m^{2}-m}} \tag{4}$$

Please refer to [6] for the channel model's derivation. For the purposes of simulation, the IEEE 802.15.4a group has provided Matlab code to generate an arbitrary number of impulse responses. The channel model uses a modified Saleh-Valenzuela model for time-varying decay in fading. As the Design Problem Statement states, an accurate channel model must include includes a channel response in terms of a "time-varying FIR filter, a time-varying gain for fading, and Additive White Gaussian Noise (AWGN)" [2]. With the exception of AWGN, the IEEE 802.15.4a channel model fits the constraint. Additionally, this model is better than the Rician and Rayleigh models because it is a more general form. I added AWGN to the signal separately in the simulation because the channel model does not include AWGN.

After finalizing the channel model, I had to choose an algorithm for receiver timing recovery and DSSS despreading. I implemented the DSSS despreading with the XOR function after discussing despreading with Pierre. Please refer to Section 3.3 above for details on using XOR for DSSS despreading. Timing recovery refers to the process of calculating the starting point of the data in the received sequence. The following section contains more details of the timing recovery algorithms in the project.

### 3.2.3 Timing recovery design decisions

After discussing timing recovery with Dr. Evans, I chose header correlation and filter delay compensation for the DBPSK receiver and the OQPSK receiver, respectively. In header correlation, one prefixes a predefined header to the actual data before transmission. The receiver cross-correlates the received data with the header in order to find the most likely index of the header. Please refer to *Telecommunication Breakdown* [7] for more details on header correlation. Filter delay compensation is a technique which works in desktop simulations like the project's simulation. In filter delay compensation, one calculates the index of the start of the data sequence based on the delay of the pulse shaping filters. The book *Simulation and Software Radio for Mobile Communications* outlines this technique [8]. Another major design aspect of the project is the quantitative comparison of OQPSK versus DBPSK. Please refer to Section 8.0 for the analysis.

## 4.0 DESIGN IMPLEMENTATION

I modified the header correlation technique for the DBPSK system and the pulse shaping filter of the OQPSK system. The IEEE 802.15.4 protocol specifies a header consisting of a string of zeroes followed by a unique sequence of zeroes and ones for synchronization purposes. I designed a header which contained a bit pattern that was not part of the PN sequences because this header did not work with the header correlation code. The header that worked is the binary sequence 0101010101101011110010 1. The pulse shape for O-QPSK in the IEEE 802.15.4 standard is the half sine shape with this equation [9]:

$$\sin\left(\pi \frac{t}{2T_c}\right) \qquad (5)$$

In the equation, $T_c$ is the PN period. I plotted the pulse shape and observed that the shape did not truncate properly because this pulse shape did not work in the implementation,. Therefore, I used the square root raised cosine pulse shape for the OQPSK system. The frequency spectrum of the square-root raised cosine pulse shape is more strictly bounded compared to the spectrum of the half-sine shape because the square-root raised cosine shape is a better approximation of the ideal sinc pulse.

The engineers developing the IEEE 802.15.4 standard specified low data rates so that an implementation should not need channel equalization [10]. After discussing the channel modeling with EE 464 TA Scott Chi, I decided to simulate a guard time by using the delay between iterations of the simulation loop to counteract channel memory. This choice also allowed me to use the `filter` function in Matlab for the DBPSK system instead of the `conv` function to convolve the signal with the channel impulse response. The simulation is not affected by the replacement of the `conv` function because the extra terms in the `conv` result do not affect future data blocks when one uses guard times.

The source code and block diagram for the simulation are located in Appendix C. The Matlab Communications Toolbox contained the functions needed for the BPSK simulation. I adapted the pulse shape filtering code and header correlation code from *Telecommunication Breakdown* [7] to implement the pulse shape filtering and header correlation in the BPSK system. The design of the raised cosine filter follows the square-root raised cosine template in *Telecommunication Breakdown* [7]. I used the modulation functions in *Simulation* [8] instead of Matlab's OQPSK modulations functions because the functions in [8] allowed me to work with the In-phase and Quadrature-phase channels separately unlike the functions in the Communications Toolbox. The OQPSK code also calls the square-root raised cosine filtering function from *Simulation.*

## 5.0 TEST AND EVALUATION

The testing process consisted of automated and manual testing. For the symbol generation, I wrote code to generate randomly-ordered symbols for both the OQPSK and the DBPSK systems to test them for a wide range of data. I tested the modulation and pulse shaping individually before combining them with the timing recovery, DSSS, and AWGN portions. Once the simulation worked with a simple AWGN channel, I implemented and tested the simulation with the channel model. For automated testing of symbol recovery, I used Matlab's biterr function and plotting the results in logarithmic BER plots. To manually test the code, I compared the recovered data and the original data at several places to check for equality. The following section contains a discussion of the BER plots.

**5.1 Bit Error Rate Analysis**

Figure 2 below shows the BER for a BPSK system with DSSS in a simple AWGN channel. The red points denote the BER for the system. The only purpose of the Reference BER plots in Figures 2, 4, and 5 is to adjust the SNR scale to display all the SNR values in the simulation. In this report, the term BER is interchangeable with the Symbol Error Rate (SER) for both the BPSK and the OQPSK systems. In Figure 2, the BER is close to zero for all SNR values after 4dB because the number of bits corrupted in the received signal did not affect the XOR despreading process. Please see Section 3.3 for more details on using XOR for DSSS despreading.

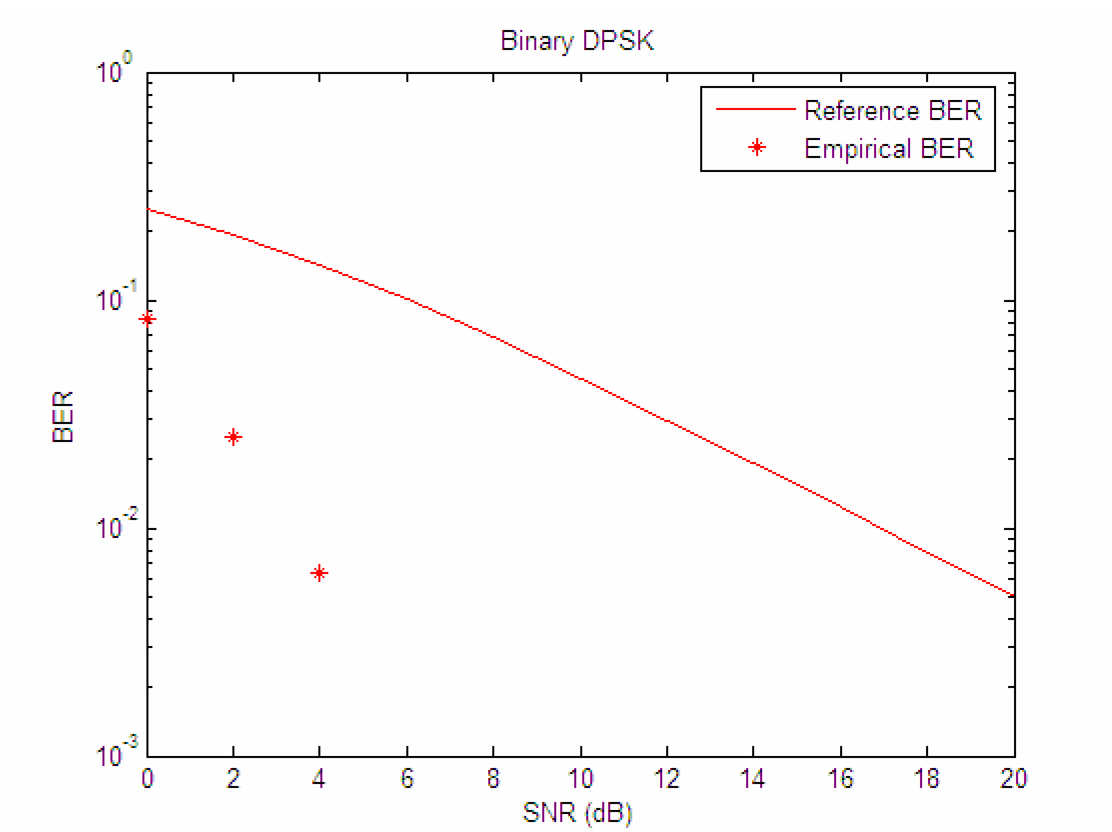


**Figure 2. BER plot of BPSK system with AWGN**

Figure 3 shows that the BER continually decreases with increasing SNR. This result is consistent with the expectation that a higher SNR results in a lower bit error rate.

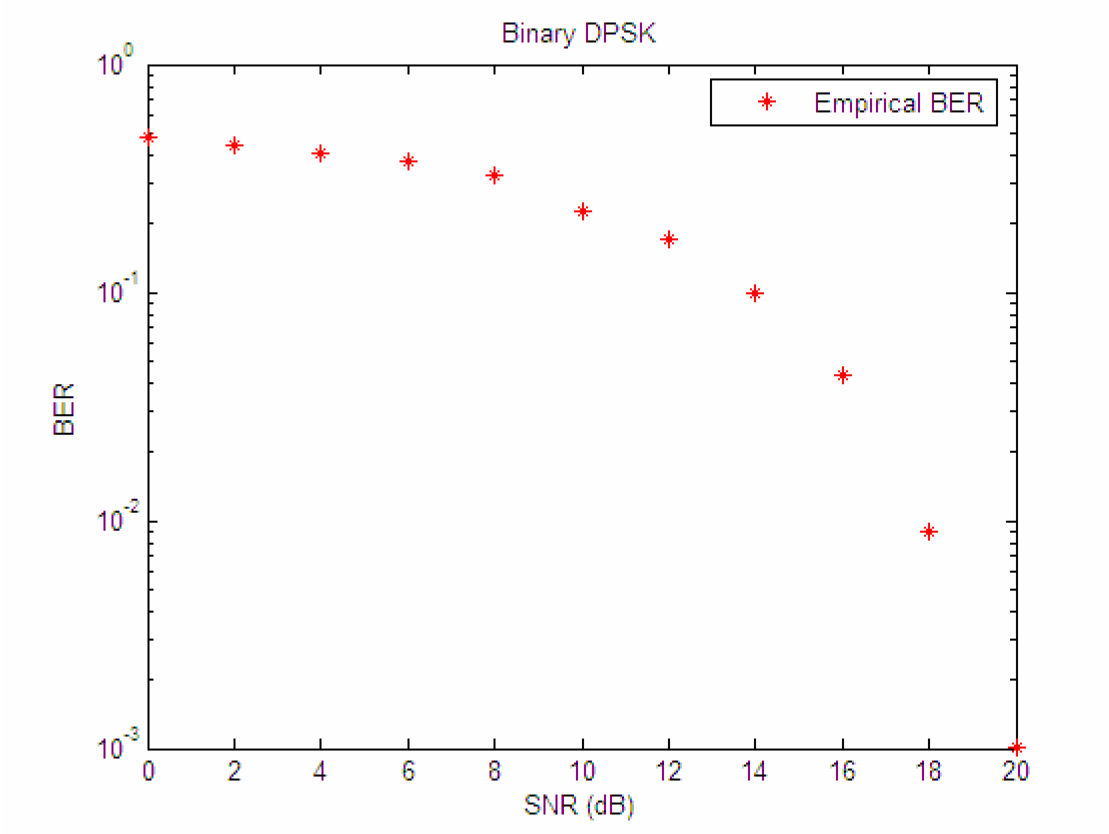


**Figure 3. BER plot of BPSK system with AWGN and channel model**

Figure 4 on the next page contains the BER plot of the OQPSK system in a simple AWGN channel.

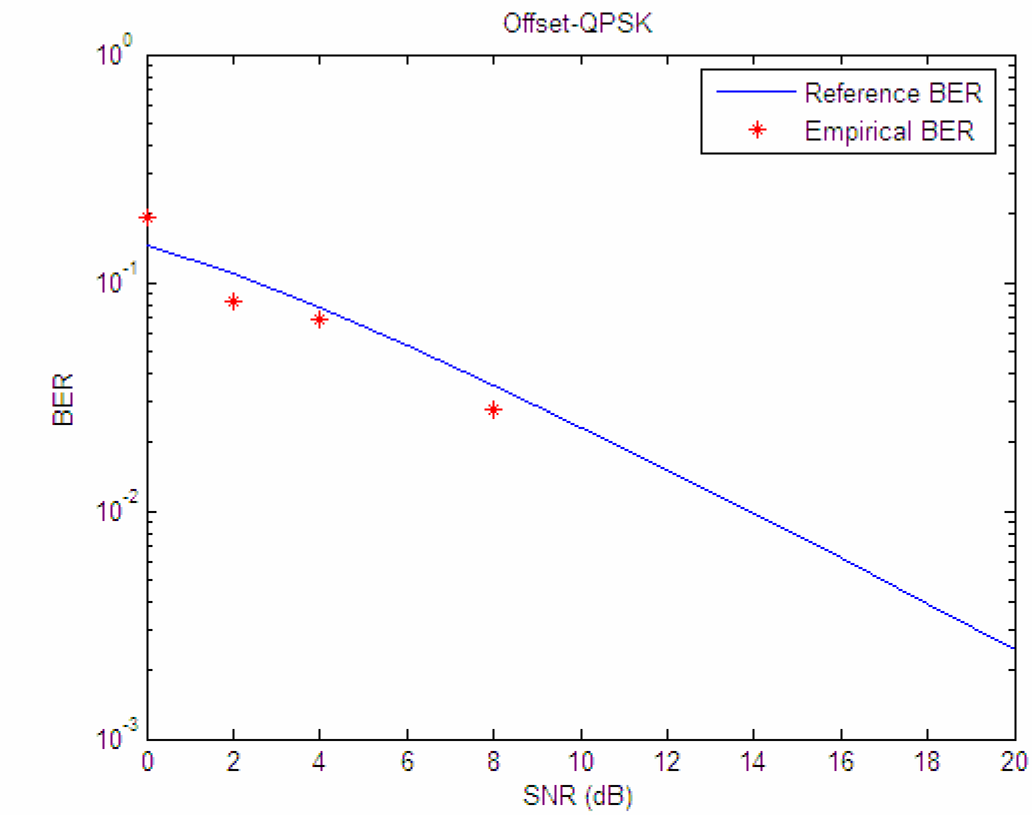


**Figure 4. BER plot of OQPSK system with AWGN**

Comparing Figure 2 and Figure 4, one can observe that the error rate in the BPSK system is better because it has an upper bound of 1/10 and a lower bound of 1/1000, while the OQPSK system contains probabilities of errors higher than 1/10 and higher than 1/100. The BER in Figure 4 is zero for SNR values above 8 dB because of the robustness of the DSSS despreading process. Figure 5 on the next page contains the BER plot for the OQPSK system with the channel model and AWGN corruption.

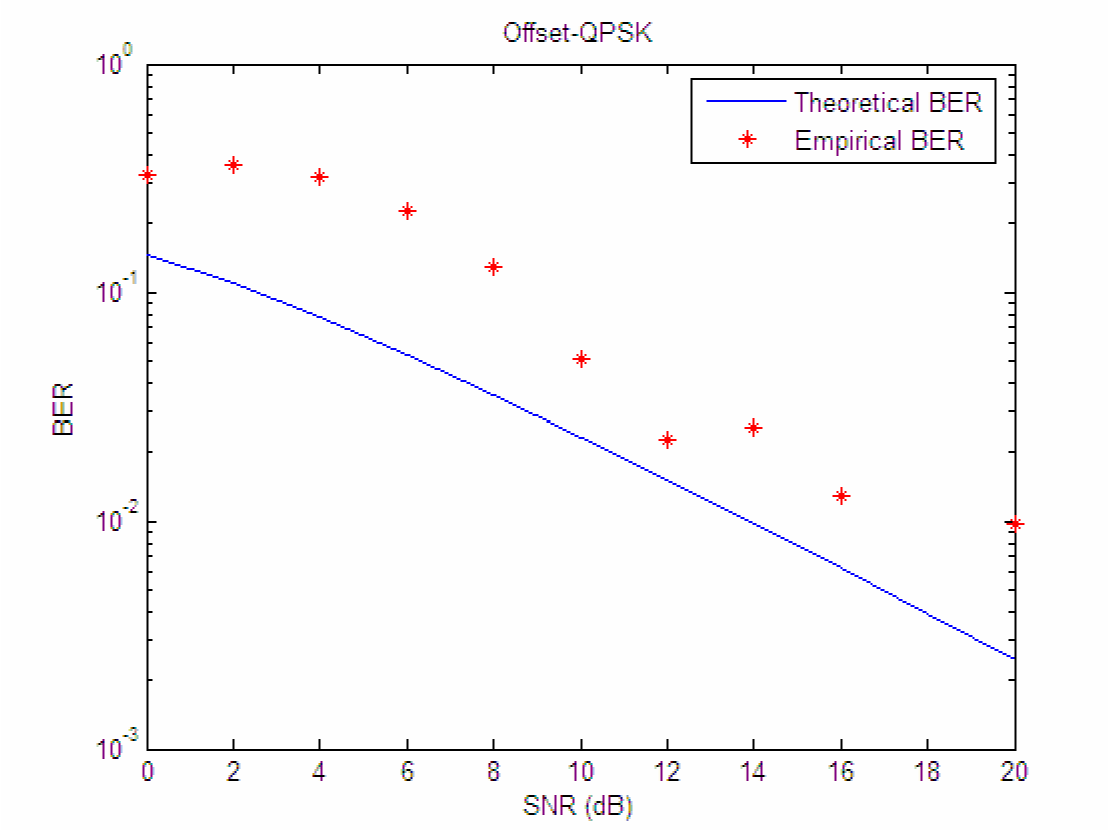


**Figure 5. BER plot of OQPSK system with AWGN and channel model**

The BER trend in Figure 5 is consistent with Telecommunications theory which states that the bit error rates decreases with increasing SNR. The only points at which the data deviates from the general trend are 2dB and 14dB. However, the plot is accurate because the error rate does not drastically increase by an order of magnitude at 2dB and 14dB. The following table contains the data points from Figure 5 and Figure 3. Table 2 on the next page displays the higher bit error rates of the BPSK system, when compared to the OQPSK system. In this case, the empirical data is inconsistent with theory. As I mentioned in Section 3.1.2, the bit error rate is theoretically higher in an OQPSK system compared to a BPSK system because of the higher probability of incorrect demodulation when a particular phase shift transforms into the other phase shifts due to noise and channel attenuation. To further investigate that the OQPSK system offers better error rates, an RF simulation is necessary. Because the OQPSK system uses the popular 2.4 GHz carrier frequency, one might find that the interference in the OQPSK system's 2.4 GHz frequency band leads to higher bit error rates when compared to the 815 MHz or 915 MHz BPSK systems.

**Table 2. BER Comparison of the OQPSK and BPSK systems**

| SNR (dB) | BPSK Bit Error Rate | OQPSK Bit Error Rate |
|---|---|---|
| 0 | $10^{-0.5}$ | $10^{-0.7}$ |
| 2 | $10^{-0.6}$ | $10^{-0.65}$ |
| 4 | $10^{-0.61}$ | $10^{-0.7}$ |
| 6 | $10^{-0.65}$ | $10^{-0.9}$ |
| 8 | $10^{-0.7}$ | $10^{-0.9}$ |
| 10 | $10^{-0.8}$ | $10^{-1.5}$ |
| 12 | $10^{-0.9}$ | $10^{-1.75}$ |
| 14 | $10^{-1}$ | $10^{-1.7}$ |
| 16 | $10^{-1.5}$ | $10^{-1.9}$ |
| 18 | $10^{-2.1}$ | 0 |
| 20 | $10^{-3}$ | $10^{-2}$ |

Other methods to quantitatively compare the BPSK and the OQPSK system are implementation complexity and data rate. The following section contains a comparison of data rate and implementation complexity.

## 5.2 Data Rate and Implementation Complexity

The BPSK system transmits 1000 symbols (bits) from the transmitter to the receiver after in each iteration of the simulation loop, while the OQPSK system successfully sends 1280 symbols in each iteration. Because two bits form a symbol in OQPSK, the OQPSK system transmits 2560 bits. The algorithmic complexity of both the OQPSK simulation and the BPSK simulation is O( $N^3$ ) due to the simulation loop, the BER calculation loop, and the DSSS despreading loop. The bit rates for the OQPSK system and the BPSK system are in the same order of magnitude as the number of bits transmitted in one second in column three of Table 1. The implementation complexity of the OQPSK simulation is greater than the complexity of the BPSK simulation in terms of memory and compute cycles. The BPSK system only requires 30 bits for storing the PN sequences, while the OQPSK system requires 512 bits for storing its PN sequences. The DSSS despreading loop for the OQPSK system requires sixteen iterations to identify the transmitted PN sequence, while the BPSK system requires only two iterations. The OQPSK modulator and the demodulator have approximately twice the computational requirements as the BPSK modulator and the demodulator because QPSK is a two-dimensional version of BPSK.

## 6.0 TIME AND COST CONSIDERATIONS

The project met the time and budget constraints. There was no cost associated with the project because Matlab and the Communications Toolbox are available free of charge on the department's computers. However, one would have to purchase Matlab and the Communications Toolbox for future work on this project. For an RF simulation of the IEEE 802.15.4 protocol, one would have to purchase a license for an RF simulator. A hardware implementation would require Wireless Sensor Network (WSN) motes for a hardware implementation. A commercial Matlab license that includes the Communications Toolbox costs $2900 [11]. One can purchase the Crossbow Imote2 boards with battery slots for $299 for a hardware implementation [12].

## 7.0 SAFETY AND ETHICAL ASPECTS OF DESIGN

Safety is a concern in IEEE 802.15.4 systems in implementations such as medical sensors and industrial transducers. Because this project involves a software study of the IEEE 802.15.4 protocol instead of a hardware implementation, the project does not directly affect the safety of human beings or the environment. To ensure accurate results, I used random generation of symbols in each transmitter to ensure that the protocol worked independently of the particular data. Additionally, I ran the simulation many times to ensure the repeatability of the results. This project is useful to an engineer who needs to implement an IEEE 802.15.4 system for medical applications, industrial applications, or the entertainment industry. The results of this project would help a telecommunications engineer decide whether the OQPSK Physical Layer or the DBPSK Physical Layer is suitable for an IEEE 802.15.4 implementation. To thoroughly investigate the safety and reliability of the IEEE 802.15.4 protocol, however, one would have to create an RF simulation of the protocol or implement the protocol on hardware.

## 8.0 CONCLUSIONS AND RECOMMENDATIONS

In this Final Report, I have presented the design documentation on quantifying tradeoffs in the IEEE 802.15.4 protocol. This paper has discussed the principles of BPSK and OQPSK modulation, DSSS, and channel modeling. The design decisions involved choosing the channel model and implementing the DSSS and timing recovery algorithms. In the actual simulation, I designed a header that would work with the header correlation technique. I separated the blocks of data to be transmitted using guard times because the IEEE 802.15.4 protocol is designed to work without channel equalization. The simulation's BER plots and Table 2 show that the OQPSK Physical Layer has better bit error rates than the BPSK Physical Layer. Because this result contradicts theory, an RF simulation is necessary to further study the bit error rates. The OQPSK system offers twice the data rate of the BPSK system, but the OQPSK system requires more memory and compute cycles. Based on the results of this project, one would choose the 868 MHz or 968 MHz BPSK Physical Layer to optimize the hardware cost and implement the 2.4 GHz Physical Layer to maximize data rate.

The final solution meets the project requirements in offering a baseband simulation of the IEEE 802.15.4 Physical Layer and analysis of the modulation techniques. The limitations of the project are its non real-time nature and its exclusion of RF simulation. Because the 2.4 GHz frequency is unlicensed by the Federal Communications Commission (FCC) and other regulatory agencies, several devices use this frequency. While the IEEE 802.15.4a group designed the channel model based on real studies of interference, an RF simulation would be necessary in studying the effects of interference from other 2.4 GHz sources on the IEEE 802.15.4 2.4 GHz system. A hardware implementation would offer the opportunity for a real-time implementation of the protocol. In summary, future work on the IEEE 802.15.4 protocol would involve either a hardware implementation or a software RF simulation. Future work on this project might also quantify the effect of guard times on data throughput. A study of the effect of guard times on data throughput would verify that the guard times do not decrease the data throughput below the levels in Table 1.

## REFERENCES


[1] Email Communication with Dr. Brian Evans, May 26, 2007.

[2] J.A. Gutierrez, E.H. Callaway, and R.L. Barrett, *Low-Rate Wireless Personal Area Networks*, New York, N.Y., 2004, p.40.

[3] "Phase-shift keying" *Wikipedia*, http://en.wikipedia.org/wiki/Image:DBQPSK_timing_diag.png (current 7 Aug. 2007)

[4] J.A. Gutierrez, E.H. Callaway, and R.L. Barrett, *Low-Rate Wireless Personal Area Networks*, New York, N.Y., 2004, p.28.

[5] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization", *IEEE Communications Magazine,* vol. 35, no. 7, July 1997, pp. 90-100.

[6] A.F. Molisch, et. al., "IEEE 802.15.4a channel model - final report", *University of Berkeley Electrical Engineering and Computer Science* http://www.eecs.berkeley.edu/~slansize/research/channel_models/ 15-04-0662-00-004a-channel-model-final-report-r1.pdf (current 6 August 2007).

[7] C.R. Johnson and W.A. Sethares, *Telecommunication Breakdown*, Prentice Hall, Upper Saddle River, N.J., 2004.

[8] H. Harada and R. Prasad, Simulation *and Software Radio for Mobile Communications*, Norwood, MA, 2002.

[9] IEEE Std. 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE, New York, N.Y., 2006.

[10] Ed Callaway, et. al., "Home Networking with IEEE 802.15.4, a Developing Standard for Low-Rate Wireless Personal Area Networks," *IEEE Communications Magazine, Special Issue on Home Networking*, vol. 40, no. 8, August 2002, pp. 70-77.

[11] The Mathworks Store: http://www.mathworks.com/store/default.do (Current 6 August 2007)

[12] Crossbow Inertial Systems: http://www.xbow.com/Products/productdetails.aspx?sid=267 (Current 6 August 2007)

# APPENDIX A – GLOSSARY

## APPENDIX A – GLOSSARY

**Baseband:** A frequency band centered at 0 Hz.

**Bit Error Rate (BER) :** Probability of Error. The probability with which a bit might be corrupted.

**Channel Equalizer:** A filter which counteracts the effect of channel memory.

**Impulse Response:** The output of a system to the delta function. In continuous-time systems, this is the response of the system to the Dirac delta function. In discrete-time systems, this is the response of the system to the Kronecker delta function. The impulse response characterizes a linear system because the output of a linear system to any input is the convolution of that input and the impulse response.

**Symbol:** A group of bits that can be encoded as one unit. In a BPSK system, the Each bit is also a symbol because only one bit can be modulated at a time. In an QPSK or OQPSK system, two bits can be modulated at one time because the carrier wave can assume any one of four phase values. Therefore a symbol in a QPSK system consists of two bits.

## APPENDIX B – BLOCK DIAGRAM

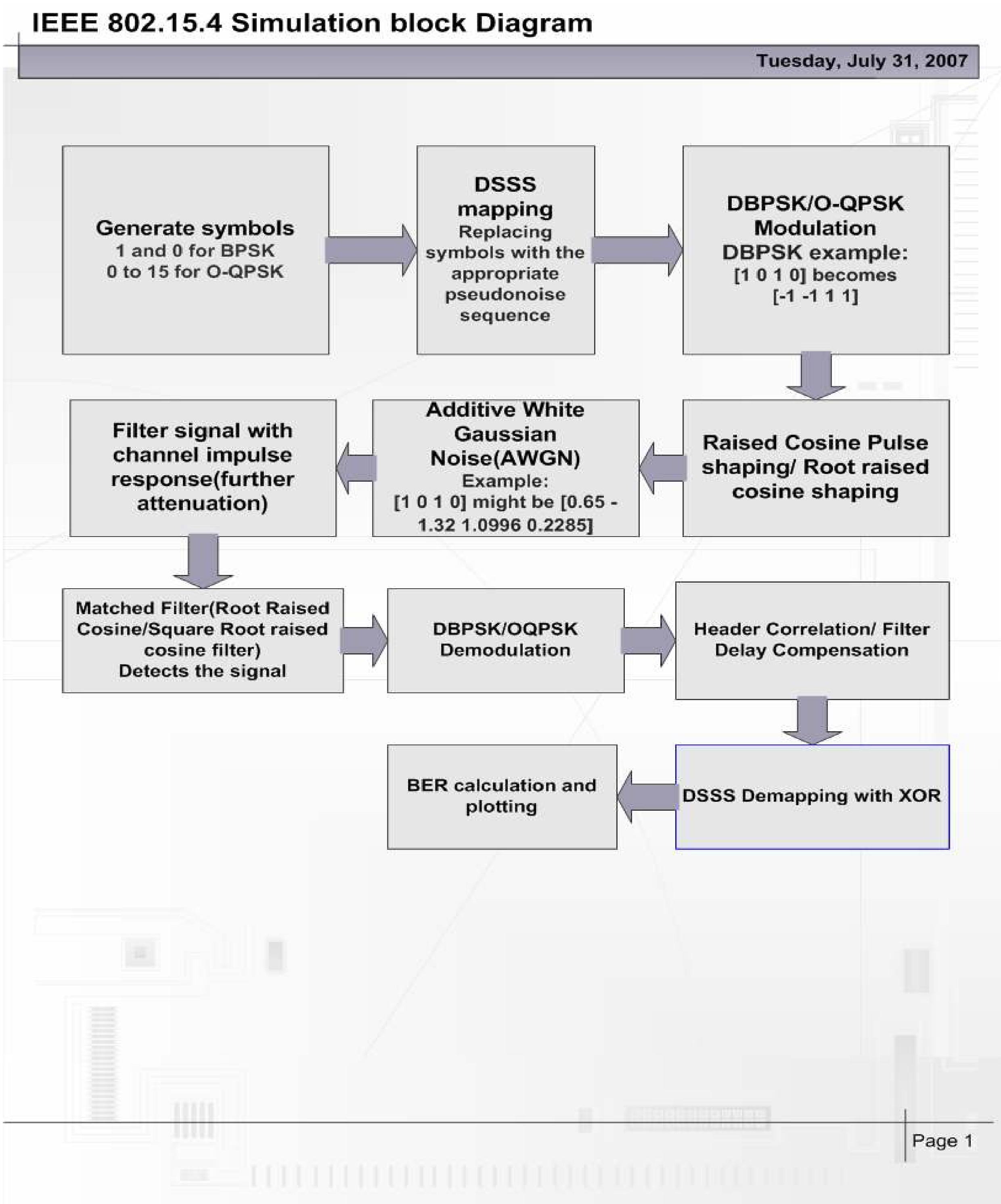


**Figure 6. IEEE 802.15.4 simulation block diagram**

**This page is intentionally blank (Please omit)**

## APPENDIX C – SIMULATION SOURCE CODE

# APPENDIX C – SIMULATION SOURCE CODE

## BPSK System

```
%PN Maps
one_map = [0  0  0  0  1  0  1  0  0  1  1  0  1  1  1];
zero_map = [1  1  1  1  0  1  0  1  1  0  0  1  0  0  0];
chiplen = 15;

M = 2;          % Number of modulation levels
data_length = 1000;
osamprate = 10;     % Number of oversamples
iter=10;   % Number of simulation loops


%read in the impulse responses from a file
imrmatrix = csvread('cm1_imr.csv',0,0);


chnidx = 2;

for iii=1:iter
    chn = imrmatrix(1:end,chnidx);


%******************** Data generation ********************************
    header = [0  1  0  1  0  1  0  1  0  1  1  0  1  0  1  1  1  1  0  0  1  0  1];
    tx = [header];
    rawmsg = randsrc(data_length,1,[0:M-1]);

%********************DSSS spreading**********************************
    for k = 1:length(rawmsg)
        if (rawmsg(k) == 1)
            tx = [tx one_map];
        else
            tx = [tx zero_map];
        end;
    end;
%*************BPSK Modulation and Pulse Shaping***********

    N = length(tx);
    dpsksig = dpskmod(tx,M);
    realsig = real(dpsksig);

    %Oversample the data using osamprate
    mup = zeros(1,N*osamprate);
    mup(1:osamprate:end) = realsig;

    %Get the raised cosine pulse shape coefficients
    L = 10; ps = rc(L,1,M);
    ps = ps / sqrt (sum(ps.^2));
```

```
    %convolve pulse shape with data
    g = filter(ps,1,mup);

    %convolve signal with channel impulse response
    mod_sig = filter(chn,1,g);

    %Get matched filter coefficients
    recfilt = rc(L,1,M);
    recfilt = recfilt / sqrt(sum(recfilt.^2));
%**********************************************************

    SNR=0:2:20;
for n=1:length(SNR)

  %filter signal with matched filter
  v = filter(fliplr(recfilt),1,mod_sig);

  %add AWGN
  msg_rx_data = awgn(v,SNR(n));

  %Downsample
  vdownsamp = msg_rx_data(1:osamprate:end);

  %Demodulate
  msg_demod = dpskdemod(vdownsamp,M);

   %locating the header through cross-correlation
   y = xcorr(header,msg_demod(2:end));
   [p,ind] = max(y);
   headerstart = length(msg_demod)-ind;
   if headerstart > 100
        headerstart = 4; %if header correlation fails, make a guess based on
empirical observation.
   end;

   %find index where actual data starts
   seqstart = headerstart + length(header) + 1;

   % Despreading the signal
   recovered = [0];

   for qp=seqstart:length(msg_demod)
    if((seqstart+(chiplen-1)) < length(msg_demod))
      one_map_result = xor(msg_demod(seqstart:seqstart+(chiplen-1)),one_map);
      zero_map_result = xor(msg_demod(seqstart:seqstart+(chiplen-
1)),zero_map);

      if( sum(one_map_result) < sum(zero_map_result))
          recovered = [recovered 1];
      else
          recovered = [recovered 0];
      end;
```

```
      qp = qp + chiplen;
      seqstart = seqstart + chiplen;
    end
   end;



   %rawmsg is a column vector, so take transpose
   compmsg = rawmsg';


   %the if statement is just a sanity check
   %Use the biterr function to find the BER

  if(length(recovered)==length(compmsg))
   [nErrors,BER(n)] = biterr(recovered(2:end),compmsg(1:(length(rawmsg)-1)));
  else
   [nErrors,BER(n)] =
biterr(recovered(2:end),compmsg(1:(length(recovered))));
  end;


end;

if (chnidx < 12)
    chnidx = chnidx + 2;
end;

end

%***************Plotting Results**************************************

% Compute theoretical performance results, for comparison.
%BERtheory = berfading(SNR,'dpsk',M,1);

% Plot BER results.
 %semilogy(SNR,BERtheory,'b-',SNR,BER,'r*');
 semilogy(SNR,BER,'r*');
 legend('Empirical BER');
 xlabel('SNR (dB)'); ylabel('BER');
 title('Binary DPSK');
```

## OQPSK System

```
% Modulation, Demodulation, and Pulse shaping adapted from the book
Simulation and Software Radio by H. Harada and R. Prasad.(Please See the
References section)

%*****PseudoNoise table****************************************
PN = [1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0;
      1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0;
      0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0;
      0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1;
      0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1;
      0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0;
      1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1;
      1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1;
      1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1;
      1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1;
      0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1;
      0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0;
      0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0;
      0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1;
      1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0;
      1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0;
    ];
%******************** Preparation part ************************************

sr=256000.0; % Symbol rate
ml=2;        % ml:Number of modulation levels (BPSK:ml=1, QPSK:ml=2,
16QAM:ml=4)
br=sr.*ml;   % bit rate
nd = 1280;   % Number of symbols that simulates in each loop
EbN0=0:2:20;
IPOINT=8;    % Number of oversamples
imrmatrix = csvread('cm1_imr.csv',0,0);
chnidx = 2; %Starting point where we read channel impulse responses

%************************* Filter initialization **************************

irfn=21;                    % Number of taps
alfs=1.0;                   % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1);   %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0);  %Receiver filter coefficients


nloop=10;  % Number of simulation loops
chiplen = 32;


rawmsgone = randsrc(80,1,[0:15]);
rawmsgtwo = randsrc(80,1,[0:15]);
tx = [1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0];
reference = [0];

    %*************DSSS spreading ********************************
```

```
    for k = 1:length(rawmsgone)
      sym = rawmsgone(k);
      tx = [tx PN((sym+1),1:end)];
    end;
    %***********************************************************




for iii=1:nloop
  for n=1:length(EbN0)
    ebn0=EbN0(n);
    txvec = tx;
    chn = imrmatrix(1:end,chnidx);





%*************************** OQPSK Modulation
********************************

    [ich,qch]=qpskmod(txvec,1,nd,ml);
    [ich1,qch1]=compoversamp(ich,qch,length(ich),IPOINT);
    ich21=[ich1 zeros(1,IPOINT/2)];
    qch21=[zeros(1,IPOINT/2) qch1];
    [ich2, qch2]=compconv(ich21,qch21,xh);

%**************************** Attenuation Calculation **********************

   spow=sum(ich2.*ich2+qch2.*qch2)/nd;
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn);





%********************* Add White Gaussian Noise (AWGN) **********************

    [ichchm,qchchm] = compconv(ich2,qch2,chn);
    [ich3,qch3]= compconv(ichchm,qchchm,xh2);
  [ich4,qch4]= comb(ich3,qch3,attn);% add white gaussian noise

    syncpoint=irfn*IPOINT+1;
    ich5=ich4(syncpoint:IPOINT:length(ich4));
    qch5=qch4(syncpoint+IPOINT/2:IPOINT:length(qch4));

%**************************** OQPSK Demodulation
*****************************

    [demodata]=qpskdemod(ich5,qch5,1,nd,ml);
```

```
    seqstart = 33;
    %********DSSS De-spreading******************************************
      recovered = [-1];
      temp = 0;
      result = zeros(16,32);
      while (seqstart+(chiplen) < length(demodata))
    for j=1:16
        result(j,1:end) = xor(demodata(seqstart:seqstart+(chiplen-
1)),PN(j,1:end));
    end
          for ii=1:16
        r(ii) = sum(result(ii,1:end));
          end
          [m,minidx] = min(r);

        recovered = [recovered (minidx-1)];
        seqstart = seqstart + chiplen;
      end;

    compmsg = rawmsgone';


 [nErrors,BER(n)] = biterr(recovered(2:end),compmsg(1:length(compmsg)-2));

  end % for n=1:length(EbN0)
  if (chnidx < 12)
    chnidx = chnidx + 2;
  end;
end % for iii=1:nloop

BERtheory = berfading(EbN0,'psk',4,1);
semilogy(EbN0,BERtheory,'b-',EbN0,BER,'r*');
legend('Theoretical BER','Empirical BER');
xlabel('SNR (dB)'); ylabel('BER');
title('Offset-QPSK');
```

**Raised Cosine Channel Model**

```
function s=rc(syms, beta, P, t_off);

%Based on srrc.m from Telecommunication Breakdown.
% s=rc(syms, beta, P, t_off);
% Generate a Square-Root Raised Cosine Pulse
%       'syms' is 1/2 the length of srrc pulse in symbol durations
%       'beta' is the rolloff factor: beta=0 gives the sinc function
%       'P' is the oversampling factor
%       't_off' is the phase (or timing) offset
```

```
if nargin==3, t_off=0; end;
% if unspecified, offset is 0
k=-syms*P+1e-8+t_off:syms*P+1e-8+t_off;
% sampling indices as a multiple of T/P
if (beta==0), beta=1e-8; end;
% numerical problems if beta=0

s=(sin(pi*k/P)./(pi*k/P)).*( cos(pi*beta*k/P)) ./ (1-
4*(beta.^2)*(k/P).^2)
```