A Communication Monitor for Wireless Sensor Networks Based on Software Defined Radio

Yian Qin, Zhitao He

Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden Email: yian@sics.se, zhitao@sics.se

> SICS Technical Report T2011:03 ISSN 1100-3154, ISRN:SICS-T-2011/03-SE

> > January 26, 2011

Abstract

Link quality estimation of reliability-crucial wireless sensor networks (WSNs) is often limited by the observability and testability of single-chip radio transceivers. The estimation is often based on collection of packer-level statistics, including packet reception rate, or vendor-specific registers, such as CC2420's Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI). The speed or accuracy of such metrics limits the performance of reliability mechanisms built in wireless sensor networks. To improve link quality estimation in WSNs, we designed a powerful wireless communication monitor based on Software Defined Radio (SDR). We studied the relations between three implemented link quality metrics and packet reception rate under different channel conditions. Based on a comparison of the metrics' relative advantages, we proposed using a combination of them for fast and accurate estimation of a sensor network link.

Keywords

Wireless sensor networks, Link quality estimation, Software-defined radio

Contents

1	Inti	roduction	1
	1.1	Problem statement	1
	1.2	Solution	2
	1.3	Report Structure	3
2	Bac	kground	4
	2.1	Software Defined Radio (SDR)	4
	2.2	GNU Radio and USRP(Universal Software Radio Peripheral)	4
		2.2.1 GNU Radio	4
		2.2.2 USRP	5
	2.3	IEEE 802.15.4	5
	2.4	CC2420	9
	2.5	UCLA ZigBee PHY library	9
	2.6	WxPython GUI toolkit and some GNU modules	9
	2.7	Tmote Sky	10
3	\mathbf{Stu}	dy of The UCLA ZigBee PHY Library	11
3	Stu 3.1	dy of The UCLA ZigBee PHY Library Digital communication receivers	11 11
3	Stu 3.1	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion	11 11 12
3	Stu 3.1	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery	11 11 12 12
3	Stu 3.1	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recovery	 11 12 12 13
3	Stu 3.1 3.2	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recoveryM &M timing recovery in digital synchronous data recovers	 11 11 12 12 13 15
3	Stu 3.1 3.2 3.3	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recoveryM &M timing recovery in digital synchronous data recoversMMSE FIR interpolator	 11 12 12 13 15 15
3	Stu 3.1 3.2 3.3 3.4	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recoveryM &M timing recovery in digital synchronous data recoversMMSE FIR interpolatorO-QPSK implemented as FSK	 11 12 12 13 15 15 16
3	Stu 3.1 3.2 3.3 3.4 3.5	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recoveryM &M timing recovery in digital synchronous data recoversMMSE FIR interpolatorO-QPSK implemented as FSKSummary of UCLA ZigBee PHY receiver	 11 12 12 13 15 15 16 18
3	Stu 3.1 3.2 3.3 3.4 3.5 Des	dy of The UCLA ZigBee PHY Library Digital communication receivers 3.1.1 Downconversion 3.1.2 Timing recovery 3.1.3 Phase recovery M &M timing recovery in digital synchronous data recovers MMSE FIR interpolator O-QPSK implemented as FSK Summary of UCLA ZigBee PHY receiver ign and Implementation of The Monitor	 11 12 12 13 15 16 18 20
3	Stu 3.1 3.2 3.3 3.4 3.5 Des 4.1	dy of The UCLA ZigBee PHY LibraryDigital communication receivers3.1.1Downconversion3.1.2Timing recovery3.1.3Phase recoveryM &M timing recovery in digital synchronous data recoversMMSE FIR interpolatorO-QPSK implemented as FSKSummary of UCLA ZigBee PHY receiverSummary of UCLA ZigBee PHY receiverSummary of UCLA ZigBee PHY receiverA simple SDR based IEEE802.15.4 communication system	 11 12 12 13 15 15 16 18 20 20

	4.3	Received signal strength indicator (RSSI)	24
	4.4	Link quality estimation by Chip Error Rate (CER)	24
	4.5	Link quality estimation by Error Vector Magnitude (EVM) .	25
	4.6	Link quality estimation at frequency domain	26
5	Cal	ibration of The Monitor	29
	5.1	Equipment for calibration	29
	5.2	Method for calibration	30
	5.3	Result	30
6	Cor	nbined Link Quality Estimation	31
	6.1	Test methods and environment	31
		6.1.1 Clean channel	31
		6.1.2 AWGN channel	32
		6.1.3 In-band noise	32
		6.1.4 Adjacent channel interference	33
	6.2	Test result and comparison	33
		6.2.1 Signal strength VS. packet reception rate	33
		6.2.2 Chip error rate VS. packet reception rate	34
		6.2.3 Error vector magnitude VS. packet reception rate	35
		6.2.4 Result of Spectrum Factor	37
	6.3	Distinguish various sources of noise and interference	37
	6.4	A combined method of link quality estimation	39
7	Cor	clusion and Future Work	40
	7.1	Conclusion	40
	7.2	Futrue Work	40

Chapter 1

Introduction

A Wireless Sensor Network (WSN) is a number of sensor nodes, ranging from less than ten to thousands, connected by radio transceivers. Nowadays, WSNs are more and more commonly used in some reliability-crucial areas, such as health monitoring, pilot-less vehicles and industrial control. To complicate matters, the networks are normally working in radio-noisy urban areas. It becomes crucial for the WSN designer to accurately estimate and effectively control the link quality of network.

Furthermore, many power-saving and reliability-guarantee optimization strategies depend on a link quality estimation. They can benefit a lot from an accurate estimation method.

1.1 Problem statement

Limited observability and testability of single-chip radio transceivers are the most ticklish matters for WSN developer. The signal-chip radio transceivers, like CC2420, make the whole radio transmission and receiving operation as a black box. It only offers network researchers with two registers about network reliability, which are RSSI and LQI. RSSI simply integrates the energy in the band. It can not separate signal from back ground noise or random interference. As studied in [11], RSSI has a low correlation with link quality. In [17], it is also mentioned RSSI can not distinguish random interference signal. [11] indicates LQI is better than RSSI, but standard deviation is still high for the measurement.

Link quality can be measured directly be sending a large amount of "hello" packets. The accuracy of this method depends on the the number of packets be sent. Also, it wastes network resource. More fundamentally, it is not real time. As studied in some papers, such as [4], current packet reception rate (PRR) could be far different from future PRR.

Furthermore, measuring link quality on physical layer with traditional radio equipment, such as spectrum analyzer or oscilloscope, is neither a feasible choice, since it's difficult to be integrated in a WSN.

So, how to open this black box of signal-chip transceiver, and estimate the link quality of physical layer accurately on real-time has become a crucial topic of designing highly reliable WSNs.

1.2 Solution

To eliminate the problem described above, we first opened the black box by building up an IEEE 802.15.4 [1] physical layer communication connection based on Software Defined Radio (SDR). Since the entire physical layer is software defined in SDR, it is possible for us to fully monitor and control the radio communication. Then we implemented a multi-function software defined monitor that can not only work as a spectrum analyzer or oscilloscope, but can also show the received signal strength and link quality. We implemented and compared three different link quality metrics. Then a combined method of link quality estimation is introduced.

The first metric of link quality estimation is measuring the Chip Error Rate (CER). Since IEEE 802.15.4 uses DSSS technique, each symbol is spread into a pseudo-noise (PN) sequence of 32 chips. The decoder compares every received sequence of 32 chips with each PN sequence, then chooses the most likely PN sequence for decoding. We assume the chip to symbol decoding is always correct, and then count the number of error receiving chips in each PN sequence received. The accuracy of this measurement depends on the success rate of chip decoding.

The second metric is measuring Error Vector Magnitude (EVM). It is supposed that the demodulation of each chip is correct, then we measure the ratio of error vector to the signal vector. Its accuracy is limited by the success rate of chip demodulation.

The third metric is "Spectrum Factor" (SF), which shows the distortion of the signal spectrum. The credibility of this method is influenced by the type of noise, because different types of noise have different spectra. Some kinds of noise, such as multi-path effect and collision with other IEEE 802.15.4 nodes, may have the same spectrum.

In this work, the above three metrics were evaluated and compared in four kinds of radio environments. They are clean channel, Additive White Gauss Noise (AWGN) channel, channel with in-band noise and channel with adjacent channel interference. In the clean channel testing, transmitter and receiver are connected by a coaxial cable. It is used to simulate the nodes in radio-free open area. AWGN channel is the most typical channel in urban communication environment. In the in-band noise testing, a noise that has almost the same spectrum was added. This noise is used to simulate multipath effect and collision with other IEEE 802.15.4 nodes. Channel with adjacent channel interference, the last one, is used to simulate interference from the communication in adjacent channel.

After the evaluating and comparison, we propose a combined method of link quality estimation, which is real time, and more accurate than the LQI offered by CC2420.

1.3 Report Structure

The remainder of this report is structured as follows. Chapter 2 gives a short introduction to SDR and the tools used in this project. Chapter 3 is a detailed study of UCLA ZigBee library, especially of its receiver. The details of implementation of the monitor are given in Chapter 4, which is followed by Chapter 5 : Calibration of the Monitor. In Chapter 6, we test and compare the methods of link quality estimation and propose for a combined method. Finally, Chapter 7 presents the conclusion and a suggestion of future work.

Chapter 2

Background

2.1 Software Defined Radio (SDR)

SDR is a radio in which some or the entire physical layer functions are software defined[8]. SDR is firstly coined by Joseph Mitola III in 1991 to signal the shift from digital radio to multi-band multi-mode software-defined radios. After almost 20 years, it has become a hot spot of communication research, because of not only its flexibility but also its observability and testability. In SDR, 80% of the functionality is provided in software, which is fully observable and testable. A general SDR diagram is shown in Figure 2.1. More information and relative research can be found in [8].

2.2 GNU Radio and USRP(Universal Software Radio Peripheral)

2.2.1 GNU Radio

As the definition on GNU Radio's official web site, "GNU Radio is a free software development toolkit that provides the signal processing runtime and processing blocks to implement software radios using readily-available, low-cost external RF hardware and commodity processors." People build a radio system by creating a graph, which contain signal processing blocks and data flows between them. The performance signal processing blocks are written in C++, and then they are glued with some other nonperformancecritical functions by Python code. From the Python point of view, the radio system is just flows of the data; all C++ processing blocks are black boxes. SWIG (Simplified wrapper and Interface Generator) is used to generate the



Figure 2.1: General SDR Diagram

interface of C++ block for Python.

2.2.2 USRP

The USRP is a general purpose hardware that makes GNU Radio work with general computers. It is composed of two parts: the motherboard that base band signal runs on, and the daughterboard, which shifts signal from baseband to RF frequency. The USB 2.0 interface is used to connect the USRP and the computer.

The mother board has 4 high-speed analog to digital converters (ADCs), 12 bits per sample and 64M Samples/sec for each. It also has 4 high-speed digital to analog converters (DACs), each at 14 bits per sample, 128M Samples/sec. These 4 input and 4 output channels are connected to an Altera Cyclone EP1C12 FPGA. The FPGA connects to a Cypress FX2 USB2 interface chip, and on to the computer.

Currently, there are 5 series of daughter boards available: basic TX/RX Daughter Boards, Low Frequency TX/RX Daughter Boards, TVRX Daughter Boards, DBSRX Daughter Boards, RFX Daughter Boards. The daughter board used in this work is RFX2400.

2.3 IEEE 802.15.4

IEEE 802.15.4 is a Medium Access Control (MAC) and Physical layer (PHY) specification, which is used by ZigBee. Since it's low power, low cost, easy to



Figure 2.2: USRP Block Diagram with GNU Radio

realize and reliable, IEEE 802.15.4 is widely used in sensor networks. The PHYs, which we focus on, is specified in Table 2.1 as introduced in [1].

The PHYs, which work at 868 or 915 MHz, support over-the-air data rates of 20 kb/s, 40 kb/s, and optionally 100kb/s and 250kb/s. The 2450 MHz PHY, which is used by CC2420 and this work, supports an over-the-air data rate of 250 kb/s [1]. We are going to introduce this 2450 MHz PHY in detail, which is used as ISM band and shared by WiFi and Bluetooth.

Frequency

For 2.4G PHY, the channel numbers are from 11 to 26 [1]. That means 16 channels in total, with a bandwidth of 5 MHz for each. The central frequency of channel can be calculated as fallow:

Fc = 2405 + 5k - 11 in megahertz, for $k = 11, 12, \dots, 26$

Where, k is the channel number.

Spreading

The 16-ary quasi-orthogonal technique is used in 2.4G PHY. For each data symbol, four information bits are used to select one of the 16 almost orthogonal pseudo-random noise (PN) sequences. Each PN sequence has 32 chips. The symbol-to-chip mapping relation is shown in Table 2.2.

O-QPSK

Chips with even index are modulated onto the in-phase (I) carrier and chips with odd-index are modulated onto the quadrature-phase (Q) carrier. Q-phase is delayed by Tc for forming the offset.

PYH	Frequency	Chip	Modu-	Bit	Symbol	Symbols
(MHz)	band	Rate	lation	Rate	Rate	
	(MHz)	(kch-		(kb/s)	(ks/s)	
		ips/s)				
868/915	868-868.6	300	BPSK	20	20	Binary
	902-928	600	BPSK	40	40	Binary
868/915						20-bit
(optional)	868-868.6	400	ASK	250	12.5	PSSS
	902-928	1600	ASK	250	50	5-bit PSSS
868/915						16-ary
(optional)	868-868.6	400	O-QPSK	100	20	Orthogonal
						16-ary
	902-928	1000	O-QPSK	250	62.5	Orthogonal
	2400-					16-ary
2450	2483.5	2000	O-QPSK	250	62.5	Orthogonal

Table 2.1: Frequency bands and data rates



Figure 2.3: O-QPSK chip offsets [1]

Pulse Shape

The IEEE 802.15.4 is adopted by half-sine pulse shape, as which described by following equation:

$$y = \begin{cases} \sin(\pi \frac{t}{2T}) & 0 \le t \le 2T_c \\ 0 & \text{otherwise} \end{cases}$$

A sample baseband chip sequence with half-sine pulse shaping is shown in Figure 2.4

Data	Data symbol	Chip values
symbol (binary)		$(c0 c1 \dots c30 c31)$
(decimal)	$(b0 \ b1 \ b2 \ b3)$	
0	0000	11011001110000110101001000101110
1	1000	11101101100111000011010100100010
2	0100	00101110110110011100001101010010
3	1100	00100010111011011001110000110101
4	0010	01010010001011101101100111000011
5	1010	00110101001000101110110110011100
6	0110	11000011010100100010111011011001
7	1110	10011100001101010010001011101101
8	0001	10001100100101100000011101111011
9	1001	10111000110010010110000001110111
10	0101	01111011100011001001011000000111
11	1101	01110111101110001100100101100000
12	0011	00000111011110111000110010010110
13	1011	01100000011101111011100011001001
14	0111	10010110000001110111101110001100
15	1111	11001001011000000111011110111000

Table 2.2: symbol-to-chip mapping



Figure 2.4: sample baseband chip sequence with pulse shaping[1]

2.4 CC2420

The CC2420 is a single chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver. It includes a digital direct sequence spread spectrum baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps [10]. A block diagram is shown in figure 2.5.



Figure 2.5: CC2420 simplified block diagram[10]

2.5 UCLA ZigBee PHY library

The UCLA ZigBee PHY Library, which was originally written by Thomas Schmid, is a part of the open source project GNU Radio. It extends GNU Radio and uses USRP to communicate with CC2420 (IEEE 802.15.4) or CC1000 single chip RF transceivers. CC2420 has been introduced in section 2.4. Since CC1000 is not used in this work, We will not pay attention on it here. For more information of CC1000, please go to the web site [9].

This library will be detailed in section 3

2.6 WxPython GUI toolkit and some GNU modules

WxPython is a plat-independent GUI toolkit for Python. All GUI in this work is based on the following modules:

stdgui2	Basic GUI stuff. I is almost always needed.
fftsink2	Used to plot FFTs.
scopesink2	Oscilloscope sink.
form	Often used input forms, such as text box, slider, button.
frame	The wx.Frame of windows.
panel	A panel to hold your wxControl widget
vbox	A vertical box sizer, used to align your widgets in the panel.

2.7 Tmote Sky

Tmote Sky is a low power wireless sensor platform with 802.15.4 compatible CC2420 RF transceiver. It is used as signal generator adjacent channel noise and calibrate the monitor.



Figure 2.6: A Tmote Sky with SMA port

Chapter 3

Study of The UCLA ZigBee PHY Library

The first step of this project is to build an IEEE 802.25.4 communication system. The existing "UCLA ZigBee PHY Library" makes this step easier. Just like many open source projects, UCLA ZigBee PHY Library has very good open source code, but not enough documents. The only limited instruction we can get is a single very simple web page. The study of this library, especial its demodulation algorithm and auction became a mandatory preliminary work for this work.

A receiver realized by UCLA ZigBee PHY Library is a fully digital receiver (see section 3.1), in which O-QPSK is modulated and demodulated as FSK(see section 3.4). Timing is recovered by M&M method (see section 3.2), and adjusted by MMSE FIR Interpolator (see section 3.3).

In this section we will describe UCLA ZigBee PHY Library's demodulator throughly, from theory to actual code.

3.1 Digital communication receivers

Our system has a fixed chip rate of 1/T, since 802.15.4 has a unique chip rate of 2M chips/s. ("chip rate" may be also called "symbol rate" in some communication books. We call it "chip" to distinguish it from the symbol before frequency spreading (see section 2.3)) We use 1/Ts for the sampling frequency. Due to the time shift between transmitter and receiver clocks, samples at time $t = kT + \hat{\varepsilon}T, k \in 0, 1, 2...$ are required, if the signal is transmitted at $t = kT, k \in 0, 1, 2...$ $\hat{\varepsilon}T$ is the estimation of time shift. The process of getting $t = kT + \hat{\varepsilon}T$ is called timing recovery. Communication receivers are divided into three categories by the timing recovery method: analog, Hybrid, digital. The analog and hybrid receivers recover timing by controlling the sampling instant of the received signal (see Figure 3.1.a and Figure 3.1.b). The sampling process is synchronized to the symbol timing of the received signal. The analog receivers derive timing information from analog signal by an analog processor, but the hybrid receivers get it from the samples. For them, the samples and chips have the same rate. It means $T = T_s$

In the fully digital receiver (see Figure 3.1.c), which CC2420 and our SDR system belong to, there only exists sampling clock at $t = kT_s$, which is incommensurate with the chip rate 1/T. It means we can choose any sampling frequency 1/Ts higher than chip rate 1/T. A most simple and general example is typical PAM receiver as shown in Figure 3.2. The radio signal is first down-converted by the mixer and filter, and then sampled at any frequency higher than 1/T. After that, the frequency offset of local frequency synthesizer is compensated by the frequency error detector and frequency integrator. In UCLA ZigBee PHY library, there is a FM demodulator between the downconversion and frequency compensation. The third step is time recovery, which in realized by the time estimator , interpolator and decimator. The phase estimator in the figure does not exist in UCLA ZigBee PHY library what only cares about frequency., since it demodulates O-QPSK as FSK. The last step is symbol (which is called chip in a system with DSSS) decoding.

3.1.1 Downconversion

In a zero intermediate frequency system, like ours, baseband signal is equal to intermediate frequency signal. So, the radio signal is firstly down-converted to baseband by local mixer and oscillator that is controlled by a frequency control loop. The purpose of it is to generate a complex baseband signal with both I-phase and Q-phase components, but not coherent demodulation as that in an analog receiver. In our SDR, this process is done by the daughterboard in USRP.

3.1.2 Timing recovery

In digital receiver, the following two steps are needed for timing recovery:

1. Estimation

The fraction time delay ε_0 should be estimated. In our system M&M

algorithm is used (see section 3.2). The estimation $\hat{\varepsilon}_0$ is used as if it is the true value ε_0 . [12]

2. Interpolation and Decimation

From the samples $\{kT_s\}$ a set of samples $\{kT_s + \hat{\varepsilon}_0\}$ must be computed. This operation is called interpolation. It is realized by MMSE FIR interpolator in UCLA ZigBee PHY Library (see section 3.3).

Only one sample per symbol/chip is required for further processing. This operation of decreasing sampling frequency to symbol/chip rate is called decimation.[12]

3.1.3 Phase recovery

Since the downconversion still leaves an unknown carrier phase ϕ_0 , phase recovery is needed after timing recovery. The process of phase recovery thus comprises the following three steps: Phase Estimation, Phase Rotation, Frequency synchronization.

Because the UCLA ZigBee PHY Library implement O-QPSK by FSK (see section 3.4), only the instantaneous frequency or phase shift is used for



Figure 3.1: Three Timing Recovery Method[12] (a) Analog Timing Recovery, (b) Hybrid Timing Recovery, (c) Digital Timing Recovery



Figure 3.2: Block Diagram of a typical Digital PAM Receiver[12]

demodulation. There is no phase recovery section in our system. I'll introduce phase recovery as simple as possible. You can find more instructions in [12].

3.2 M &M timing recovery in digital synchronous data recovers

There was a class of fast converging timing recovery methods for synchronous digital data receivers published on IEEE Transactions on Communications, May 1976. This method is named under the inventors' family names as "Mueller & Müller" or M&M for abbreviation [13]. In 1995, the optimized M&M, which is used in our system, is published [6].

The optimized M&M synchronizer is a discrete-time error-tracking synchronizer, which derives an indication about the delay difference between the received signal and the sampling clock. The block diagram of a digital receiver with M&M synchronizer is illustrated as Figure 3.3.



Figure 3.3: Block Diagram of Receiver with M & M Synchronizer

3.3 MMSE FIR interpolator

The task of the interpolator is to compute intermediate values between signal samples $x(kT_s)$. The impulse response of ideal interpolator is the sampled $\sin(s)$ function

$$h_1(nT_s, \mu T_s) = \sin[\frac{\pi}{T_s}(nT_s + \mu T_s)]$$
 $n = \dots, -1, 0, 1, \dots$

The filter can be thought of as an FIR filter with infinite number of taps [12]

$$\begin{aligned} h_n(\mu) &= h_I(nT_s, \mu T_s) \\ &= \sin[\frac{\pi}{T_s}(nT_s + \mu T_s)] \quad n = \dots, -1, 0, 1, \dots \end{aligned}$$

For practice, the interpolator must be approximated by a finite order filter (see Figure 3.4). MMSE is one of the approximation methods to minimize the quadratic error between the impulse response of the ideal interpolator and its approximation. For detail of MMSE algorithm, please take a look of [15].



Figure 3.4: Fourth-Order Interpolating Filter [12]

3.4 O-QPSK implemented as FSK

In the UCLA ZigBee PHY receiver, O-QPSK is implemented as frequency modulation consisting of a sequence of positive or negative frequency shifts, each frequency shift having the value $Dfc = 1/(4T_b)$, where Tb is the bit (or chip) interval. The resulting modulation index is m = 0.5 [14].

The upper part of Figure 3.5 shows the phase of the minimum IQ symbol set 00, 01, 10, 11 with half-sine shaping (see section 2.3) over the symbol interval. The lower part shows the phase transition and frequency shift. Examining the figure carefully, we can see that the sign of Δf depends on the values of I and Q. For even bit clock intervals, the sign of Δf is calculated by the value given by

$$k = IxorQ$$

and for odd bit clock intervals, the sign of $\triangle f$ is calculated by

$$k = not(IxorQ)$$



Figure 3.5: Symbol Set and Phase Transition Diagrams For O-QPSK Modulation[14]

After a FM demodulator, we will get a date stream coded by frequency, but not phase. For matching IEEE 802.15.4 specification, the simplest approach is to encode the chip-to-symbol map directly as shown in Table 3.1.

Data	Chip values	Data symbol
symbol	$(c0 c1 \dots c30 c31)$	(binary)
(decimal)		$(b0 \ b1 \ b2 \ b3)$
0	01100000011101111010111001101100	0000
1	01001110000001110111101011100110	1000
2	01101100111000000111011110101110	0100
3	01100110110011100000011101111010	1100
4	001011100110110011100000000010011	0010
5	01111010111001101100111000000111	1010
6	01110111101011100110110011100000	0110
7	00000111011110101110011011001110	1110
8	00011111100010000101000110010011	0001
9	00110001111110001000010100011001	1001
10	00010011000111111000100001010001	0101
11	00011001001100011111100010000101	1101
12	01010001100100110001111110001000	0011
13	00000101000110010011000111111000	1011
14	00001000010100011001001100011111	0111
15	01111000100001010001100100110001	1111

Table 3.1: chip-to-symbol matching with FM coding

3.5 Summary of UCLA ZigBee PHY receiver

Now, I'd like to summarize what the UCLA ZigBee PHY receiver is: It is a full digital receiver with M&M clock recovery and MMSE FIR interpolator, which implements O-QPSK as FSK.

The block diagram of UCLA ZigBee PHY receiver is shown in Figure 3.6. The signal is downconverted to baseband by USRP and sent to PC as a complex data stream with I-phase and Q-phase signals. Firstly, PC removes signal smaller than the threshold with the squelch. Secondly, the signal is demodulated to frequency by the FM Demodulator. Thirdly, it high-passes the FM demodulator's output to remove frequency offset. Fourthly,

the samples are recovered by clock recovery and interpolation. Final, the recovered samples are sent to chip decoder and regenerated to the packet.



Figure 3.6: Block Diagram of UCLA ZigBee PHY Receiver

The best way to understand this recovery in software way is by its flow chart, as Figure 3.7. The block "gr.quadrature_demod_cf" works as the FM demodulator by extracting the phase shift between two adjacent samples. The output of FM demodulator is high-passed by removing the low frequency part. The block "gr.single_pole_iir_filter_ff" works as an IIR low-pass digital filter. The M&M clock recovery, MMSE FIR interpolator, decimator are integrated in block "gr.clock_recovery_mm_ff". Finally, the "ucla.ieee_15_4_paket_sink" decodes chips into symbols, rebuild the packet from symbols and show the packet on screen.



Figure 3.7: The Flow Chart of UCLA ZigBee PHY Receiver

Chapter 4

Design and Implementation of The Monitor

In this chapter, the design and implementation of the monitor based on software-defined radio (SDR) are described.

4.1 A simple SDR based IEEE802.15.4 communication system

Our communication system is built on *cc2420_txtest.py* and *cc2420_rxtest.py*, which are the example programs offered in UCLA ZigBee Library.

For adding a GUI, WxPython is used. On the receiver side, the original code of $cc2420_rxtest.py$ is packaged into one hierarchical block (something equal to a class in C++), which is called in the GUI top block. Furthermore, several intermediate variables in the hierarchical block are exported for monitoring.

4.2 RF power spectrum analyzer and oscilloscope

Spectrum analyzer and oscilloscope are two of the most commonly used tools in radio signal analysis. The implementation of them are based on two blocks from GNU Radio "*fft_sink_c*" and "*scope_sink_c*".

Frequency

The band width of a signal IEEE 802.15.4 channel is 5MHz, in which 2MHz is used as guard band [5]. After some attempt, the sampling rate is chosen to

be 4MHz, since it is just twice of chip rate and gives the best time recovery and demodulation results.

Implementation

From GNU Radio, there are two blocks very helpful for this project. They are " fft_sink_c " and " $scope_sink_c$ " [2].

"fft_sink_c" is a FFT sink block for complex data samples with GUI, which can be used as a spectrum analyzer. The original form of it is (It's slightly different from that in [2]):

fftsink.fft_sink_x(fg, parent, title=", baseband_freq=0, y_divs=10,

ref_scale=50, sample_rate=1, fft_size=512, fft_rate=15, average=False, avg_alpha=None, size=(640,240), peak_hold=False)

For displaying received signal strength, this FFT sink is adjusted. In this work, the parameters are set as Table 4.1.

parameter	value
title	"ZigBee Spectrum"
Baseband frequency	cordic_freq (central frequency)
Y_devs	10
ref_scale	13490
$sample_rate$	4000000
fft_size	512
fft_rate	15

Table 4.1: Parameters for fft_sink_c

On the spectrum analyzer, three more buttons are added. The first one is "Show RSSI", which is used for printing the received signal strength in the status bar on the left bottom corner. The second one, "Show LQI2", is used to show the spectrum factor in the status bar in the right bottom corner. When you click the third one, "PK SRCH", the maximum value on the spectrum will be shown.

" $scope_sink_c$ " is an oscilloscope sink block for complex data samples with graphic user interface (GUI). The original form of it is:

Scopesink2.scope_sink_x(fg, parent, title=", sample_rate=1, size=(640,240), frame_decim=1, v_scale=1000, t_scale=None)

In this work, the parameters are set as Table 4.2.

parameter	value
title	"ZigBee Scope"
$sample_rate$	4000000
$frame_decim$	1
v_scale	1000
t_scale	None

Table 4.2: Parameters for scope_sink_c

Result

Figure 4.1 shows the GUI of the spectrum analyzer view of the monitor. It has functions of "peak hold", "average", "axis adjustment" and "auto scale".



Figure 4.1: ZigBee Spectrum

The oscilloscope is shown in Figure 4.2. "Ch1" and "Ch2" mean the I-phase and Q-phase signals.

The oscilloscope can show the constellation diagram when you choose the "Channel Option" as "XY" (see Figure 4.3). As shown on the diagram, the "stars" are settled all around a cycle. There are two main reasons for this phenomenon:

- 1. For a digital receiver, the output of ADC has not been synchronized, timing recovery is fully digital.
- 2. There's half-sine phase shaping.



Figure 4.2: ZigBee Scope



Figure 4.3: ZigBee constellation diagram

4.3 Received signal strength indicator (RSSI)

Definition and algorithm

The RSSI is used to estimate received power in the selected frequency channel. In most traditional single chip receivers, it is realized by a special circuit working in time domain, which implemented essentially as integrator. RSSI calculated in the time domain is shown in the following equation [7]:

$$\begin{aligned} RSSI[dBm] &= 10 \log(\frac{S}{1mW}) + \alpha \\ S \propto \frac{1}{T} \int_0^T (I^2 + Q^2) dt \end{aligned}$$

in which, α is the offset of S.

There is a second choice, which is calculating in frequency domain, which is more accurate, since it can distinguish the signal from out of band noises. RSSI calculated in frequency domain is shown in the following equation:

$$RSSI[dBm] = 10log(\frac{S}{1mW}) + \alpha$$
$$S = \int_{-1.5MHz}^{1.5MHz} X^2(f) df$$

in which, α is the offset of S.

where X(f) stands for the Fourier transform of received signal. Finally we chose the second one.

Implementation and result

One button for RSSI calculation is added to the original FFT sink. Pushing the "Show RSSI" button, the RSSI will be calculated in frequency domain, and shown on the status bar. (See Figure 4.1)

4.4 Link quality estimation by Chip Error Rate (CER)

Definition and algorithm

The 16-ary quasi-orthogonal technique is used in IEEE 802.15.4 2.4GHz PHY. For each data symbol, four information bits are used to select one of the 16 almost orthogonal pseudo-random noise (PN) sequences. Each PN sequence has 32 chips. Not every chip is needed to be correctly received to get a correct chip-to-symbol decoding. The UCLA ZigBee PHY decoder will compare every received sequence of 32 chips with each PN sequence

(see section 2.3 and 3.4). Then the decoder will choose the most likely PN sequence from Table 3.1 for decoding. There is also a threshold to define how few missed received chips means "likely". The symbol with too many missed received chips will be discarded.

In this project, we assume every successfully decoded symbol are correct. Then we count the missed received chips in a whole packet and calculate the average CER for the packet.

Implementation and result

The counter of missed received chips is implemented by C++ language, which is put in the chips to symbol decoder of ucla.ieee_15_4_paket_sink. The average CER will be printed out on the computer monitor for each packet.

4.5 Link quality estimation by Error Vector Magnitude (EVM)

Definition and algorithm

EVM represents the distance between measured and ideal modulated signal vectors.

As introduced in [16], there are three definition of EVM.

The most classical is that take both phase and magnitude distortion are taken into account. On this occasion, EVM is simply the magnitude of noise vector that is shown as AC in Figure 4.4.

$$EVM = \frac{\sqrt{A^2 - 2A_c A \cos(\phi) + A_c^2}}{A_c} \times 100\%$$

Where A stands for the ideal vector magnitude and A_c is the measured vector magnitude.

If we only consider the magnitude distortion, there will be another definition of EVM:

$$EVM = \begin{cases} \frac{|A-A_c|}{A_c} \times 100\% & |\phi| \le \frac{\pi}{2} \\ \frac{|A+A_c|}{A_c} \times 100\% & |\phi| > \frac{\pi}{2} \end{cases}$$

In the UCLA ZigBee library, O-QPSK is demodulated as FSK signal. It means that only the phase shift is used for demodulation. Also, there is no magnitude synchronization process in the receiver. It is difficult to define



Figure 4.4: Phasor representation of complex, magnitude and phase distorted signals [16]

the magnitude of ideal signal. So, We only consider the phase distortion for EVM.

$$EVM = \sqrt{2 - 2\cos\phi} \times 100\%$$

Implementation and result

The third definition of EVM is used in this work. The code for EVM calculation is put directly in ucla.ieee_15_4_paket_sink. The average phase error is calculated for every chip in the picket, and then used for EVM calculation.

4.6 Link quality estimation at frequency domain

Definition and algorithm

Every signal has its certain spectrum in frequency domain. The spectrum of a signal with noise will be different from that of a clean signal on most occasions. To certain extent, we can obtain some link quality information by the analysis of these spectrum. An example is shown in Figure 4.6.



Figure 4.5: EVM of UCLA Zigbee Receiver

In Figure 4.6(a), it shows that the clean signal should have a large first order component and two balanced second order components. Limited by the bandwidth of our spectrum analyzer, the higher order components are filtered out by the anti-aliasing filter and decimation filter in USRP. In Figure 4.6(b), the left second order are significantly influenced by the noise from adjacent channel. The influence of additive wight Gauss noise is shown in Figure 4.6(c). The second order components look like being moved up, since noise are added in.

These frequency domain analysis is not suitable for every kinds of noise. When it come to in-bend noise as shown in Figure 4.6(d), which is used to simulate interruption from other IEEE 802.15.4 nodes, the influence are not shown evidently, since the signal and the noise have the same spectrum.

We propose a way to measure this kind of distortion on frequency domain. This way should be simple enough to be implemented by hardware. We sum up the energy of the first order component and compare it with that of two second order components, since it is easy to be implemented by bandpass filter and integrator. We call it "spectrum factor" SF. We define the received signal at frequency domain as X(f) with the central frequency f_c .

$$SF = \frac{\int_{f_c + 1.5M}^{f_c + 1.5M} X^2(f)}{\int_{f_c + 1.5M}^{f_c - 1.5M} X^2(f) + \int_{f_c - 2M}^{f_c - 1.5M} X^2(f)}$$



Figure 4.6: Comparison of spectrum with different noise

Implementation and result

Similar to the implementation of RSSI, SF is written by Python code and added to the FFT sink. There is a "LQI2" button added to show SF on the left section of status bar.

Chapter 5

Calibration of The Monitor

5.1 Equipment for calibration

The software spectrum analyzer of our monitor is calibrated with the R3131, a truly spectrum analyzer. For eliminating influence of surrounding Tmote sky, the transmitter and receiver, are connected by SMA cables and two 20dB attenuators.

A R3131 spectrum analyzer has a frequency range from 9kHz to 3GHz. There's a power measurement function useful for evaluating radio instruments using measurements such as occupied bandwidth, adjacent channel power, channel power, etc.



Figure 5.1: ADVANTEST R3131 Spectrum Analyzer[3]

5.2 Method for calibration

As shown in Figure 5.2, R3131 and USRP are both connected to the attenuator with a T connector with 50 Ohm coaxial cable having same type and length. We assume the T connector is symmetrical and cables have good conformity. So the signal received by R3131 and USRP should be the same.



Figure 5.2: Calibration system

There are two kinds of signals used, which are generated by Tmote sky at 2.475GHz. The first one is a single tone sinusoidal signal. The second one is a continuous random IEEE 802.15.4 signal. The single tone sinusoidal signal is used to calibrate the power measurement of certain frequency; The IEEE 802.15.4 signal is used to calibrate in band power or RSSI.

5.3 Result

Every calibration result shown on the following table is the average value of five independent measurements.

	R3131	SDR
Sinusoidal signal	-44.97 dBm	-44.80 dBm
802.15.4 signal (RSSI)	-45.69 dBm	-45.56 dBm

Chapter 6

Combined Link Quality Estimation

In this section we compare the test results of the three link quality metrics, and introduce a combined method of link quality estimation.

6.1 Test methods and environment

To evaluate the performance comprehensively, we simulate four most common radio environments. For calculating packet reception rate and gather statistics, 500 packets are transmitted for each test.

6.1.1 Clean channel

In a clean channel, there is no interference from other devices and very little thermal noise. The detection of signal is mainly limited by the sensitivity of the receiver. To simulate a clean channel, we connect two USRPs, which are used as the transmitter and the receiver, by a coaxial cable and an -20 dB attenuator.

We run this test by decreasing the transmission power by 20% (about 1 dBm) each time. The received signal power decreases from -54.89 dBm to -66.82 dBm. Since the test result changes sharply when received signal strength decreases from -66.56 dBm to -66.86 dBm. More precise test is done by regulating power by 1% each time in this area.

6.1.2 AWGN channel

Additive white Gauss noise(AWGN) channel is the most common channel model in wireless communication. White noise can be caused by interference from some other wireless communication system with much wider band than IEEE 802.15.4, such as IEEE 802.11. It can also be radio radiation from some electrical devices, such as microwave oven. Thermal noise of the transceiver circuits also contribute to a part of it.

For simulating a AWGN channel, a -70dBm white Gauss noise source from GNU Radio is added to the stream of received data from the USRP receiver. A diagram of the connection is shown in Figure 6.1.



Figure 6.1: Testing System for AWGN channel

Similar to the testing method used in clean channel test, we decrease the transmission power by 20% (about 1 dBm) each time for integral test. We do more precise test by decreasing power by 1% each time in the area where test result changes sharply.

6.1.3 In-band noise

We also test the system with a noise that has the similar spectrum as the signal. It is used to simulate multi-path effect and collisions from other IEEE 802.15.4 transmitters. This noise is generated by adding a -50dBm Gaussian noise at the output of UCLA ZigBee PHY transmitter. Since the noise and signal pass through the same smoothing filter in the USRP, they have the same bandwidth.

Connection diagram is shown in Figure 6.2. Integral test is done by decreasing the transmission power by 20% (about 1 dBm) each time. More precise test is done in the area where test result changes sharply.



Figure 6.2: Testing System for In-Band Noise

6.1.4 Adjacent channel interference

Sensor networks often consist of a large amount of nodes. It is possible that two nodes transmitting on adjacent channels at the same time. On this occasion, the nodes can affect each other. For simulating this radio environment, we connect an Tmote Sky node, which transmits random data on the adjacent channel continuous at -13 dBm, to our system by a Tconnector and an extra 20 db attenuator as shown in Figure 6.3

We decrease the transmission power by 20% (about 1 dBm) each time for integral test. We do more precise test by decreasing power by 1% each time in the area where test result changes sharply.



Figure 6.3: Testing System for Adjacent Channel Interference

6.2 Test result and comparison

6.2.1 Signal strength VS. packet reception rate

Figure 6.4 indicates the relation of signal strength with packet reception rate (PRR). For calculating packet reception rate, 500 packets are transmitted

for each test.

In clean channel, PRR declines sharply at about -67 dBm. Since the receiving performance in a clean communication channel is mainly limited by the sensitivity of the receiver, it means the minimum signal USRP can received is about -67 dBm.

The PRRs are larger than 98% when the signal is stronger than -57 dBm in the AWGN channel, in-band noise channel and the channel with adjacent channel noise. They decreases regularly to 0% when the signal is weaker than 63dBm.



Figure 6.4: Signal strength VS. packet reception rate

6.2.2 Chip error rate VS. packet reception rate

Figure 6.5 reveals the relationship between Chip Error Rate (CER) and Packet Reception Rate (PRR). Stand deviation of CER is plotted as error bars on each curve.

CER is monotonic to PRR in all of the four radio environments. The standard deviation is also very small. But the ranges of CER in different radio environment is far different from each other. In AWGN channel, CER increases to 4.5% when PRR decreases to 0%. But in clean channel, CER is always below 0.1%.



Figure 6.5: Packet reception rate VS. Chip error rate

6.2.3 Error vector magnitude VS. packet reception rate

Figure 6.6 shows the relation between average Error Vector Magnitude (EVM) and Packet Reception Rate (PRR).

EVM has a very good sensitivity when PRR is high. With the decreasing of PRR, it can jump from 6% up to 21% when PRR is higher than 95%. Then it will keep increasing slowly until PRR decreases to 0%.

Stability is another important factor except for sensitivity. Figure 6.7 plots the standard deviation of EVM in different radio conditions. The standard deviation of EVM is as the same sensitive as EVM to the changing of PRR. It jumps up to 2.5% when PRR decreases to 95%.

Based on the behaviors revealed above, a summary of EVM can be made. EVM is an appropriate method to estimate a PRR higher than 95%, since it has high sensitivity. But due to the poor stability, it is not suitable for a low PRR. Reliability-crucial works at high PRR environment on most occasions. So EVM's high sensitivity for high PRR links should not be ignored.



Figure 6.6: Packet reception rate VS. error vector magnitude



Figure 6.7: Standard Deviation of EVM

6.2.4 Result of Spectrum Factor

The major advantage of Spectrum Factor (SF) is that it does not rely on successful synchronization or demodulation, thus and can be performed on even unsuccessfully demodulated packets. This enables SF to estimate PRR faster than CER and EVM when the link is very weak, since estimation based on SF does not require any successfully demodulated packet.

Figure 6.8 shows the relation between signal strength and SF. In all of the four radio environments, SF is almost linear to the single strength and has a fairly small standard deviation.



Figure 6.8: Signal Strength VS. Spectrum Fector

6.3 Distinguish various sources of noise and interference

As analyzed above, all of CER, EVM and spectrum factor have different behaviors in different radio environments. We have different link quality models for different occasions. So we need first to distinguish various sources of noise or interference, and then choose the right model to estimate link quality. Radio environment can be distinguished by the spectrum of packets. An example is shown in Figure 6.9. In Figure 6.9(a), it shows that the clean signal should have a large first order component and two balanced second order components. In Figure 6.9(b), two second order components have different amplitudes. The left second order is significantly influenced by the noise from adjacent channel. The influence of additive white Gauss noise is shown in Figure 6.9(c). The whole spectrum looks like being moved up, since a white noise is added in. When it comes to in-band noise as shown in Figure 6.9(d), the influence on spectrum is not evident, since the signal and the noise have the same bandwidth. But the EVM is much higher than that of a clean signal. High EVM and high spectrum factor are the features to distinguish signal with in-band noise from a clean signal that has low EVM and high spectrum.





(a) Signal without noise

(b) Signal with adjacent channel interference



Figure 6.9: Comparison of spectrum in different radio environment

6.4 A combined method of link quality estimation

CER has the best stability and a modest integral sensitivity in all the four kinds of communication environments simulated above. EVM has the highest sensitivity for a PRR higher than 95%, but has quite bad stability when the PRR is lower than 90%. Spectrum factor has moderate sensitivity and stability. But spectrum factor should not be ignored, since it does not rely on demodulation and can be performed on a very weak signal undetected by the demodulator.

Therefore, we propose a link quality estimate approach which combines the above three methods of link quality estimation. Before any kind of estimation, the radio environment should be firstly distinguished. CER is the best metric to approximately estimate the PRR. Then, if a highly accurate estimation is needed for a PRR higher than 95%, EVM should be used. Beside, if a very low quality channel need to be estimated, spectrum factor is the only choice.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

- High accuracy link quality estimation should be radio environment based. All the three kinds of link quality estimation methods behave differently in the four kinds of radio environments.
- Radio environment could be distinguished by embedded spectrum analyzer or the characteristics of results from the three link quality estimation methods.
- CER has a very good stability and a modest sensitivity. It is a good method to integral estimate link quality.
- EVM is very sensitive to the decreasing of link quality. It can be used to estimate a PRR higher than 95%.
- Spectrum factor does not depend on the demodulation. It can be used for the estimation of very bad link quality, in which most, even all, packets are lost.
- With the combination of the above three metrics, the estimation of link quality or PRR could be more accurate than 1% in a well distinguished radio environment.

7.2 Futrue Work

There is still something that can be done to improve the accuracy of link quality estimation.

• The accuracy of EVM depends on the success rate of chip demodulation, since it assumes chip demodulation is always correct. But the chip error rate could be up to 4.5% for a received packet as measured in the experiments in Chapter 6. We can eliminate this by calculating EVM with the chips corrected by the "most-likely" chip to symbol decoding.

This work could be slightly extended on the following aspects:

- Implement the suggested combined method of link quality by SDR.
- Use the monitor to measure actual link quality of a testbed.

There are many potential research from this work.

- Various kinds of channel modeling could be done by this new link quality estimation method.
- The link quality estimation method can be applied in some existing adaptive communication protocols.

List of Figures

2.1	General SDR Diagram	5
2.2	USRP Block Diagram with GNU Radio	6
2.3	O-QPSK chip offsets $[1]$	7
2.4	sample baseband chip sequence with pulse $shaping[1] \dots$	8
2.5	CC2420 simplified block diagram[10]	9
2.6	A Tmote Sky with SMA port	10
3.1	Three Timing Recovery Method[12] (a) Analog Timing Re-	
	covery, (b) Hybrid Timing Recovery, (c) Digital Timing Re-	
	covery	13
3.2	Block Diagram of a typical Digital PAM Receiver[12]	14
3.3	Block Diagram of Receiver with M &M Synchronizer	15
3.4	Fourth-Order Interpolating Filter [12]	16
3.5	Symbol Set and Phase Transition Diagrams For O-QPSK	
	$Modulation[14] \dots \dots$	17
3.6	Block Diagram of UCLA ZigBee PHY Receiver	19
3.7	The Flow Chart of UCLA ZigBee PHY Receiver	19
4.1	ZigBee Spectrum	22
4.2	ZigBee Scope	23
4.3	ZigBee constellation diagram	23
4.4	Phasor representation of complex, magnitude and phase dis-	
	torted signals $[16]$	26
4.5	EVM of UCLA Zigbee Receiver	27
4.6	Comparison of spectrum with different noise	28
5.1	ADVANTEST R3131 Spectrum Analyzer[3]	29
5.2	Calibration system	30
6.1	Testing System for AWGN channel	32
6.2	Testing System for In-Band Noise	33

6.3	Testing System for Adjacent Channel Interference	33
6.4	Signal strength VS. packet reception rate	34
6.5	Packet reception rate VS. Chip error rate	35
6.6	Packet reception rate VS. error vector magnitude	36
6.7	Standard Deviation of EVM	36
6.8	Signal Strength VS. Spectrum Fector	37
6.9	Comparison of spectrum in different radio environment	38

Bibliography

- [1] Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 0_1-305, 2006.
- [2] Firas Abbas. Simple user manual for gnuradio 3.1.1, 2007.
- [3] ADVANTEST. R3131 spectrum analyzer operation manual.
- [4] C.A. Boano, M.A. Zuniga, T. Voigt, A. Willig, and K. Romer. The triangle metric: Fast link quality estimation for mobile wireless sensor networks. pages 1–7, aug. 2010.
- [5] Crossbow. Avoiding rf interference between wifi and zigbee.
- [6] G.R. Danesfahani and T.G. Jeans. Optimisation of modified mueller and muller algorithm. *Electronics Letters*, 31(13):1032–1033, jun. 1995.
- [7] A. Flammini, D. Marioli, G. Mazzoleni, E. Sisinni, and A. Taroni. Received signal strength characterization for wireless sensor networking. pages 207–211, apr. 2006.
- [8] The Software Defined Radio Forum Inc. What is software defined radio?
- [9] Texas Instruments. cc1000.
- [10] Texas Instruments. cc2420.
- [11] Zhu Jian and Zhao Hai. A link quality evaluation model in wireless sensor networks. pages 1–5, jun. 2009.

- [12] Heinrich Meyr, Marc Moeneclaey, and Stefen Fechtel. Digital Communication Receiver: Synchronization, Channel Estimation and Signal Processing. John Wiley & Sons, 1998.
- [13] K. Mueller and M. Muller. Timing recovery in digital synchronous data receivers. *Communications, IEEE Transactions on*, 24(5):516– 531, may. 1976.
- [14] John Notor, Anthony Caviglia, and Gary Levy. Cmos rfic architectures for ieee 802.15.4 networks.
- [15] G. Oetken. A new approach for the design of digital interpolating filters. Acoustics, Speech and Signal Processing, IEEE Transactions on, 27(6):637–643, dec. 1979.
- [16] J.L. Pinto and I. Darwazeh. Error vector magnitude relation to magnitude and phase distortion in 8-psk systems. *Electronics Letters*, 37(7):437–438, mar. 2001.
- [17] Lei Tang, Kuang-Ching Wang, Yong Huang, and Fangming Gu. Channel characterization and link quality assessment of ieee 802.15.4compliant radio for factory environments. *Industrial Informatics, IEEE Transactions on*, 3(2):99–110, may. 2007.