THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF COMPUTING

Towards Safe Usage of COTS Wireless Devices in Medical Settings: A Profiling - Policing Framework for WBAN against WiFi Interference

By

WANG YUFEI

A Thesis Submitted in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

March 2013

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signature)

WANG Yufei (Name of Student)

ABSTRACT

Medical applications nowadays are increasingly demanding *Wireless Body Area Networks* (WBAN). Medical WBANs usually use *Industrial Scientific Medical* (ISM) band due to free license and abundant supply of low cost *Commercially Off-the-Shelf* (COTS) devices. In the 2.4GHz ISM band, WBANs usually adopt low power wireless technologies, such as Zigbee, Bluetooth, and IEEE $802.15.6$. These wireless technologies may suffer from co-channel interference from WiFi due to power asymmetry.

In this thesis, we study several challenging issues on WBAN-WiFi coexistence.

First, the two most widely deployed wireless technologies in the 2.4GHz ISM band are WiFi and Bluetooth. The pervasive existence of WiFi and Bluetooth threatens co-channel medical WBAN. Without loss of generality, we evaluate a typical WBAN scheme in the context of medical multi-parameter monitoring under WiFi and Bluetooth interference. The results show that WiFi is a major threat to WBAN; while Bluetooth is not.

Second, we propose a general WiFi-WBAN coexistence design, called WiCop. WiCop is a cross-MAC-PHY-layer solution. It suppresses WiFi interferer by transmitting customized WiFi compliant signals. Our experiments show that WiCop can double the *Packet Reception Rate* (PRR) of WBAN under intense WiFi interference. Moreover, WiCop requires no modification of existing WiFi and WBAN standards.

Third, we are also interested in passive profiling using sniffers. As each sniffer can only monitor one channel at a time, and cover a fixed area, Sniffer Channel Assignment (SCA) affects directly monitoring quality. Among the algorithms solving SCA, annealed Gibbs sampler is superior due to its distributed nature. We propose several improvements to annealed Gibbs sampler that offer faster convergence and higher chance to reach global optima.

Keywords: WBAN, Zigbee, IEEE $802.15.6$, WiFi, ISM band coexistence, policing, real-time, wireless side monitoring, Gibbs sampler, Simulated Annealing.

PUBLICATIONS

Journal Papers

1. **Yufei Wang**, Qixin Wang, "Evaluating the IEEE 802.15.6 2.4GHz WBAN Proposal on Medical Multi-Parameter Monitoring under WiFi/Bluetooth Interference," in International Journal of E-Health and Medical Communications, vol. 2, no. 3, pp. $48-62$, July-September, 2011.

2. **Yufei Wang**, Qixin Wang, Guanbo Zheng, Zheng Zeng, Rong Zheng, Qian Zhang, "WiCop: Engineering WiFi Temporal White-Spaces for Safe Operations of Wireless Personal Area Networks in Medical Applications", (accepted for publication) in IEEE Transactions on Mobile Computing (TMC), 2013.

3. **Yufei Wang**, Rong Zheng, Qixin Wang, "Self-tuned Distributed Monitoring of Multi-Channel Wireless Networks Using Gibbs Sampler", under review of *Computer Networks* of Elsevier, since $2013/01$.

Conference Papers

1. Feng Tan, **Yufei Wang**, Qixin Wang, Lei Bu, Rong Zheng, Neeraj Suri, "Guaranteeing Proper-Temporal-Embedding Safety Rules in Wireless CPS: A Hybrid Formal Modeling Approach", accepted by Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013.

2. Qixin Wang, **Yufei Wang**, Rong Zheng, Xue Liu, "Curbing Aggregate Member Flow Burstiness to Bound End-to-End Delay in Networks of TDMA Crossbar Real-Time Switches," in Proc. of the 33rd IEEE Real Time Systems Symposium (RTSS'12), Dec 5 - 7, 2012. pp.14-25.

3. Lei Rao, Qixin Wang, Xue Liu, **Yufei Wang**, "Analysis of TDMA Crossbar Real-Time Switch Design for AFDX Networks," in Proc. of IEEE INFOCOM'12, March, 2012. pp.2462-2470.

4. **Yufei Wang**, Qixin Wang, Zheng Zeng, Guanbo Zheng, and Rong Zheng, "WiCop: Engineering WiFi Whitespaces for Safe Operations of Wireless Body Networks in Medical Applications," in Proc. of the 32nd IEEE Real-Time Systems Symposium (RTSS'11), Nov 29 - Dec 2, 2011. pp.170-179.

5. Lixiong Chen, Xue Liu, Qixin Wang, **Yufei Wang**, "A Real-Time Multicast Routing Scheme for Multi-Hop Switched Fieldbuses," in Proc. of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011), Shanghai, China, April, 2011. pp.3209-3217.

Invited Talk and Demo

1. **Yufei Wang**, Qixin Wang, Zheng Zeng, Guanbo Zheng, and Rong Zheng, "WiCop: Engineering WiFi Whitespaces for Safe Operations of Wireless Body Networks in Medical Applications," in the 1st Software Radio Implementation Forum, Hong Kong, 2011. The demo of this work is available at YouTube [74] [1].

Workshop Papers

1. Tao Li, Qixin Wang, Feng Tan, Lei Bu, Jian-nong Cao, Xue Liu, **Yufei Wang**, and Rong Zheng, "From Offline Long-Run to Online Short-Run: Exploring a New Approach of Hybrid Systems Model Checking for MDPnP," in Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS/MDPnP'11), Chicago, IL, April 11, 2011.

[1] Interested reader can access this video by searching "WiCop Zigbee"at YouTube.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Qixin WANG, who offered me the opportunity to pursue my PhD study in a professional environment and with talented people. His professional supervision, vast knowledge and skill in many research areas benefited me. His expertise, kindness, understanding, and patience, added considerably to my PhD experience. I am so lucky to be the first student of Dr. WANG, and I would like to thank him for supporting me over the years, for giving me so much time to read large volume of literature, for supporting my exchange visit abroad, for sharing his researching philosophies, and for training me to be a good researcher. Without his guidance and help during the difficult times in my PhD study, this body of work would not have been possible.

I would like to thank the other members of Dr. WANG's research group - Dr. Lei Bu, Mr. Lixiong Chen, Mr. Tao Li, Dr. Xue Liu, Dr. Lei Rao, Mr. Feng Tan, Dr. Zheng Zeng, Mr. Guanbo Zheng, Dr. Rong Zheng - not only for the strong and irreplaceable guidance from them, but also for the aggressive and relaxing working environment created by them. They have become a permanent part of my memory.

I would like to give special thanks to Dr. Rong Zheng, who is always willing to revise my paper, guides me into new research areas, and shows me a lot of intuitions of mathematics. Without the help from Dr. Zheng, I would not make so much progress in my study. I would also appreciate the instructive suggestions on career planning from Dr. Xue Liu. Another important thank goes to Dr. Lei Bu, supporting me to attend CPS week forum in 2011 and 2012. Another two best wishes reach Dr. Zheng Zeng and Mr. Guangbo Zheng, who did solid experiments with me, and who rushed for paper deadlines with me together.

I also would like to thank all the faculty members from whom I learnt so much in my long journey of tertiary education. Special thanks go to Prof. Jiannong Cao, Prof. Jia You, Prof. George Baciu, Dr. Pengfei Guo, Dr. Toby Lam, Dr. Li Jiang, Dr. Zhu

Li, Dr. Yan Liu, Dr. Wei Lou, Dr. Zhili Shao, Dr. Bin Xiao, and Dr. Dan Wang at the Hong Kong Polytechnic University. Furthermore, I express my deep gratitude to Mr. Paul Chan, Dr. Kunfeng Lai, Dr. Duo Liu, Dr. Zhiwei Qin, Mr. Chunwei Tam, Dr. Dawei Tian, Ms. Renita Tung, Dr. Yi Wang, Dr. Qingjun Xiao, Mr. Tao Xiong, Ms. Lei Xu, Ms. Junmei Yao, Mr. Jin Zhang, Mr. Liang Zhang, Dr. Haomian Zheng, and Mr. Pengfei Zhu, who shared with me the pleasure of Ph.D. study at the Hong Kong Polytechnic University. More, I appreciate Prof. Qian Zhang, Dr. Kun Tan, and Dr. Dawei Tian, who shared with me their equipments and debugging skills.

I must thank Prof. Lui Sha at University of Illinois, Urbana and Champaign, for offering me the opportunity to visit the Department of Computer Science in UIUC. Also, I thank all of those who helped me during my visit to UIUC. Especially, I would like to acknowledge Dr. Yan Gao, Dr. Qunxing Huang, Mr. Min-Young Nam, Mr. Lu Su, Dr. Yong Yang, and Dr. Zheng Zeng, who supported my research.

Finally, but most importantly, I would like to thank my family. No matter they stay with me or away from me, I feel their love. This love, beyond all describable words, gives me the biggest motivation to finish this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Demand

Wireless Body Area Networks (WBAN) play a key role in health care automation. Medical WBAN can adopt different wireless schemes. Among all the candidate wireless schemes, those operating in the (2.4GHz) *Industrial Scientific Medical* (ISM) band attract more attentions (in the rest of this thesis, unless explicitly denoted, WBAN shall refer to those work in the ISM band). WBAN schemes working in ISM band include ZigBee, Bluetooth, and IEEE 802.15.6 2.4GHz etc.. The common merits of these ISM band wireless technologies include low cost and low (transmission) power. However, these WBANs often have to coexist with co-channel WiFi interferers, which are widely deployed and supposedly transmit in much stronger power.

This power asymmetry challenges WBAN-WiFi coexistence. To address this, we carry out three tasks. First, we evaluate the performance of different low power WBAN under WiFi interference; second, we propose a WiFi-WBAN coexistence solution, called WiCop, to proactively protect WBAN from WiFi interference; third, we also study how to profile the WiFi interferers passively.

For the first task, many works have evaluated the performance of ZigBee or Bluetooth WBAN under WiFi interference [21] [44] [25]. However, to our best knowledge, only our work analyzes the performance of WBAN based on the IEEE 802.15.6 2.4GHz standard.

For the second task, we propose a solution to enable coexistence of WBAN and WiFi. The solution shall meet the following requirements. First, it shall require minor changes on existing WBAN devices. Next, it shall introduce minor performance degradations to WiFi devices. Third, it shall reside on programmable wireless interface to enable cross layer design. Inspired by WiFi security research, we propose a coexistence solution,

called *WiCop*, which can meet all the three requirements. WiCop proactively transmits WiFi compliant signals (the so called "policing" signals) to suppress WiFi interferers, so as to create temporal WiFi white-spaces for WBAN transmissions.

For the third task, we study how to passively profile WiFi, aka WiFi monitoring, an important service for administrating WBAN-WiFi coexistence. Specifically, we study the sniffer based WiFi monitoring [80]. WiFi sniffers can only monitor one channel (of the entire ISM band) at a time, and monitor a fixed portion of the entire area [80]. Under such limitations, *Sniffer Channel Assignment* (SCA) strategy becomes a deciding factor on monitoring quality. SCA problem is proven to be NP-hard [17], but can be empirically solved by annealed Gibbs sampler [6]. In this thesis, we propose several methods to enhance the annealed Gibbs sampler algorithm.

Corresponding to the above three tasks, the rest of this chapter is organized as follows. Section 1.2 presents the unified research framework. Section 1.3 summarizes the contributions of this thesis. Section 1.4 gives the outline of the thesis.

1.2 The Unified Research Framework

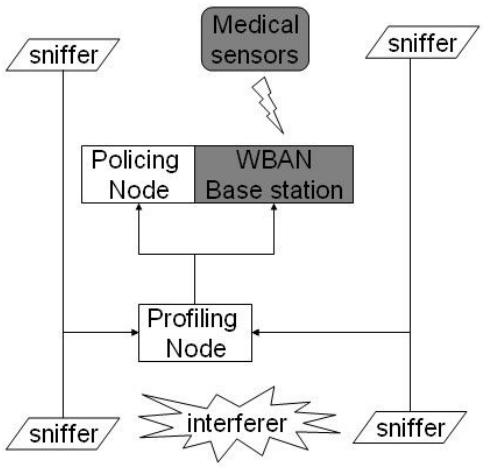


Figure 1.1: a unified framework

Fig. 1.1 gives the whole picture of our work. The two gray blocks in Fig. 1.1 are respectively the WBAN base station and medical sensors (acting as clients). Such medical WBAN may face threat from interferers (at the bottom of Fig. 1.1). To protect

and optimize WBAN, we introduce two functions: policing and profiling. Policing (wired to WBAN station in Fig. 1.1) is to regulate the behaviors of interferers. Profiling is to collect and process data from sniffers, to monitor the WBAN or the interferers. Based on the profiles of WBAN or the interferers, profiling node can tune the parameters of WBAN base station and policing node.

1.3 Contributions

The contributions of this thesis are summarized as follows.

- We are the first to evaluate the performance of IEEE 802.15.6 2.4GHz medical WBAN under WiFi/Bluetooth interference. Our evaluations conclude that WiFi poses a major threat to WBAN; while Bluetooth does not.
- We propose a solution, *WiCop*, to allow the coexistence of WBAN and WiFi. WiCop has two major advantages: 1) it requires no changes to the WBAN/WiFi standards; 2) it poses minor interferences to normal WiFi traffic.
- We implemented WiCop on *Microsoft Software Radio* (SORA). The implementation involves Windows driver programming and PHY/MAC cross layer design. The experiment result shows that WiCop can increase *Packet Reception Rate* (PRR) of WBAN by up to 116%.
- We also analyze the performance of WBAN network with/without WiCop policing. The analytical results match well the experimental results.
- We propose several methods to enhance the annealed Gibbs sampler algorithm of [6], which can increase information collected by sniffer networks and speed up convergence at the same time.

1.4 Thesis Organization

The rest of this thesis is organized as follows.

- In Chapter 2, we present the performance analysis of IEEE 802.15.6 2.4GHz medical WBAN under WiFi/Bluetooth interference.

The content of Chapter 2 is published in *International Journal of E-Health and Medical Communications* authored by *Yufei Wang, Qixin Wang* Copyright ©2011, IGI Global, www.igi-global.com. Posted by the permission of publisher.

- In Chapter 3, we present WiCop.

  The content of Chapter 3 is published (or to be published) in the following IEEE papers:

  - Copyright ©2011 IEEE. Reprinted, with permission, from *Yufei Wang, Qixin Wang, Zheng Zeng, Guanbo Zheng, Rong Zheng, "Wicop: Engineering wifi temporal white-spaces for safe operations of wireless body area networks in medical applications", in Proc. of the 32nd IEEE Real-Time Systems Symposium (RTSS'11), Nov 29 - Dec 2, 2011*
  - Copyright ©2013 IEEE. Reprinted, with permission, from *Yufei Wang, Qixin Wang, Guanbo Zheng, Zheng Zeng, Rong Zheng, Qian Zhang, "WiCop: Engineering WiFi Whitespaces for Safe Operations of Wireless Personal Area Networks in Medical Applications", (accepted for publication) in IEEE Transactions on Mobile Computing (TMC)*

- In Chapter 4, we present our enhanced annealed Gibbs sampler.

  Chapter 4 is an extension of the following IEEE paper:

  - Copyright ©2011 IEEE. Reprinted, with permission, from *Arora, P., Na Xia, Rong Zheng, "A Gibbs Sampler Approach for Optimal Distributed Monitoring of Multi-Channel Wireless Networks," in Global Telecommunications Conference (GLOBECOM 2011), Dec. 2011*

  Also, the content of Chapter 4 is under review for journal publication.

- In Chapter 5, we conclude the thesis and discuss future work.

Please note that the above reprinted materials are posted here with permissions of IGI and IEEE. Such permissions of IGI and IEEE do not in any way imply IGI/IEEE endorsement of any products or services of the Hong Kong Polytechnic University (or ProQuest/UMI). Internal or personal use of this thesis is permitted. However, permission to reprint/republish this thesis for advertising or promotional purposes or for creating new

CHAPTER 2

EVALUATING THE IEEE 802.15.6 2.4GHZ WBAN ON MEDICAL MULTI-PARAMETER MONITORING UNDER WIFI/BLUETOOTH INTERFERENCE

2.1 Demand

Health care has become a major concern for many countries across the globe. For example, the United States' health care expenditure surpassed US$2.3 trillion in 2008, which is 16.2% of the nation's GDP [1]; and China is facing the severe challenge of aging, as a consequence of long-lasting one-child policy [37].

To curb the health care crisis, medical devices and systems must be upgraded to expand capabilities, increase efficiency, improve safety, and enhance convenience. One enabling technology to these goals is *Wireless Body Area Networks* (WBAN).

A key application of WBAN is *multi-parameter monitoring* (i.e., monitoring multiple vital signs), which is widely used in medical units. For instance, during operation or intensive care, a patient must be attached with multiple electrodes to simultaneously monitor various vital signs: *ElectroCardioGraphy* (ECG), *Electroencephalography* (EEG), temperature, CO_2 level, oxygen level, blood pressure, etc.. In many cases, the patient must be plugged with these electrodes for hours, days, or even longer durations (e.g., 24×7 monitoring in Intensive Care Unit).

Traditional wired multi-parameter monitor uses wires to connect electrodes to monitor. The wires can literally tie the patient to the bed. Even worse, a small movement of the patient may stretch the wires, causing electrodes to fall off. This can be at least annoying to the patient and care givers, and sometimes even lethal.

WBAN monitoring aims to replace wired electrodes with wireless electrodes.

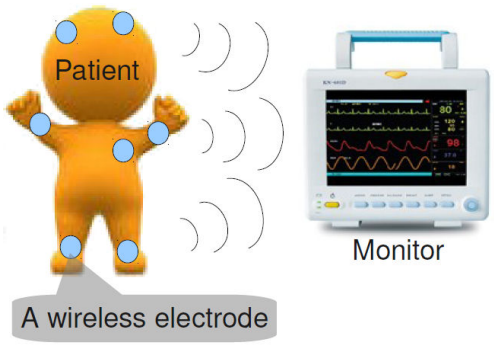


Figure 2.1: Multi-Parameter Monitoring through WBAN

Fig. 2.1 illustrates the idea. Same as wired monitoring, the patient is attached with various types of electrodes, e.g., twelve ECG electrodes, one oxygen level electrode, one blood pressure electrode, one respiration electrode, etc.. But unlike the wired case, all electrodes connect to the monitor through wireless. The monitor and all electrodes form a WBAN. The monitor serves as the *base station*, while the electrodes play the role of *clients*. We call the wireless links from the base station to clients the *downlinks*, while the wireless links from clients to base station the *uplinks*.

So far, a lot of organizations and companies have developed WBAN systems, such as CodeBlue [45] and AlarmNet [78]. Also, wireless chip vendor or medical device vendor propose a large number of wireless schemes capable of carrying out WBAN [55]. Such large number and drastic differences of proposals force the establishment of IEEE 802.15 Task Group 6 to standardize WBAN PHY/MAC in November, 2007. Now, IEEE 802.15.6 standard (released in 2012) regulates 3 categories of PHY standards – Ultra Wide Band (UWB), Human Body Communication, and Narrow Band (including several bands). The Radio Frequency bands of these PHY standard are summarized by Table 2.1.

Among these PHY standards, the Narrow Band 2.4GHz PHY is the most mature. It is mostly based on well-known PHY components, which are already widely implemented in WiFi, ZigBee, and Bluetooth. However, IEEE 802.15.6 2.4GHz standard may face the co-channel interference from WiFi and Bluetooth, which are the most two popular schemes in 2.4GHz. Therefore, it is meaningful to evaluate the IEEE 802.15.6 2.4GHz standard under co-channel interference. In this chapter, we shall focus on evaluating the coexistence performance of the IEEE 802.15.6 2.4GHz standard for medical multi-parameter monitoring under WiFi and Bluetooth interference. *For simplicity, in the rest*

Table 2.1: Number of WBAN PHY Schemes, 2012 [69]

Category	RF Band	Number of Channels
Impulse Radio UWB (IR-UWB)	$3.2 - 9.6$GHz	15
Frequency Modulation UWB (FM-UWB)	$5.9 - 9.2$GHz	≥ 2
Narrowband (NB)	$402 - 405$MHz	10
	$420 - 450$MHz	14
	$863 - 870$MHz	12
	$902 - 928$MHz	48
	$950 - 956$MHz	12
	$2360 - 2400$MHz	38
	$2400 - 2483.5$MHz	79
Total:	9	≥ 230

of the chapter, we assume WBAN PHY uses the IEEE 802.15.6 2.4GHz standard unless otherwise denoted.

The rest of this chapter is organized as follows. Section 2.2 briefly introduces IEEE 802.15.6 2.4GHz standard. Section 2.3 analyzes the Packet Error Rate (PER) of IEEE 802.15.6 standard under WiFi/Bluetooth interference. Section 2.4 uses simulation to measure the performance of a WBAN carrying out ECG monitoring. Finally, Section 2.6 summarizes this chapter.

2.2 Introduction of IEEE 802.15.6 2.4GHz

The term "2.4GHz" refers to the Radio Frequency (RF) spectrum of $2400 \sim 2483.5$MHz. The IEEE 802.15.6 2.4GHz standard divides this spectrum into 79 channels, and the carrier frequency for the n_cth ($n_c = 0, \ldots, 78$) channel is $f_c = 2402.00 + 1.00 \times n_c$ (MHz).

Regardless of the carrier frequency, in baseband, a 2.4GHz PHY packet complies with the format shown in Fig. 2.2.

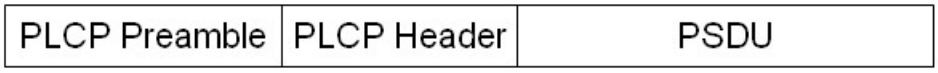


Figure 2.2: Packet Format [69]

A PHY packet consists of three segments: preamble (a.k.a., PLCP preamble), header (a.k.a., PLCP header), and payload (a.k.a., PSDU). The preamble consists of 90 fixed well-known bits: the first 63 bits are for coarse-grain synchronization, and the next 27 bits are for fine-grain synchronization. The header consists of 19 bits of information, which are expanded into 31 bits by $19/31$ BCH coding [58]. These 31 bits are repeated four times, to create the 124-bit header. The payload encodes a MAC layer packet of $9 \sim 264$ bytes with $51/63$ BCH coding, which expands every 51 bits of MAC layer packet into 63 bits.

The PHY packet preamble and header are modulated with $\pi/2$-DBPSK with a symbol rate of 600K (symbol/s). The PHY packet payload can be modulated with either $\pi/2$-DBPSK or $\pi/4$-DQPSK, both at a symbol rate of 600K (symbol/s). The $\pi/2$-DBPSK mode is mandatory. Therefore, unless explicitly denoted, we assume the PHY always uses $\pi/2$-DBPSK.

2.3 PER Analysis of 2.4GHz WBAN

In this Chapter, we focus on performance of *Modulation and Coding Schemes* (MCS) of WBAN under continuous interference. To carry out MCS performance analysis, we do below assumptions.

- WBAN, WiFi, and Bluetooth do not back to each other. This assumption is explained as follows. Fist, we suppose WBAN uses polling based MAC. In other words, WBAN devices transmit packet in Guaranteed Time Slot (GTS) [69]. Second, we suppose WiFi only backs off to WiFi device. This assumption holds in most of the cases [32]. Third, Bluetooth standard ignores *Clear Channel Assessment* (CCA).
- We regard WiFi interference as continuous noise (overlapping the whole duration of a WBAN packet). This is reasonable if we suppose WiFi interferer uses 1Mbps

data rate and carries out File Transfer Protocol (FTP), which are the basis data rate of WiFi and typical application of internet. Under such data rate and application, a typical WiFi packet lasts 12ms. Suppose WiFi uses the minimum contention windows, the typical mean back off time is 80μs. The long packet duration and short back off time supports our assumption that WiFi interference is a continuous noise. Also, considering our hypothesis that WiFi is a threat to WBAN, it suffices to suppose one typical data rate and one typical application of WiFi. Note that, in Chapter 3, we will consider the more comprehensive case where WiFi transmission may not overlap the whole WBAN packet.

- We regard Bluetooth interference as continuous noise (overlapping the whole duration of a WBAN packet). As our hypothesis is that Bluetooth is not a threat to WBAN, due to the low power of Bluetooth. Thus, it is pessimistic on WBAN side to regard Bluetooth interference as continuous noise.

The rest of this section is organized as follows. First, we will give the Bit Error Rate (BER) of WBAN in general Additive White Gaussian Noise (AWGN) channel. Second, we give the interference model of WiFi and Bluetooth. Basically, we can regard WiFi signal as white noise; while we have to treat Bluetooth signal as color noise. Third, we briefly analyze the synchronization error rate. Fourth, as IEEE 802.15.6 2.4GHz standard adopts unequal channel coding policy, we need analyze the channel coding effect of header and payload respectively. Last, we give Packet Error Rate (PER) of WBAN.

2.3.1 Bit Error Rate of 2.4GHz WBAN

In AWGN channel, the *Bit Error Rate* (BER) P_{ber} of DBPSK is:

$$P_{ber} = \frac{1}{2}\exp(-\frac{E_b}{N_0}), \tag{2.1}$$

where N_0 is the AWGN power spectrum density; E_b is the per bit energy. E_b is further decided by

$$\begin{aligned} E_b &= P_{rx}T_b, \\ P_{rx} &= P_{tx}/(10^{\alpha/10}), \end{aligned}$$

where P_{tx} is the transmitter power and α is the path loss coefficient (with the unit of dB). α is a function of transmitter-receiver distance d. $\alpha(d)$ follows the well-known log-distance model:

$$\alpha(d) = \alpha_0 + 10n \lg d/d_0, \tag{2.2}$$

where $d_0 = 0.1$(m), and α_0 and n are derived from the raw experiment data given by [69].

2.3.2 WiFi/Bluetooth Interference Model

We will show WiFi can easily jam WBAN. To show this, it suffices to show one scheme of WiFi can easily jam WBAN. Without loss of generality, we focus on IEEE 802.11b, the most basic and widely supported WiFi scheme. IEEE 802.11b PHY deploys DSSS and occupies a much wider spectrum (22MHz) than WBAN PHY (1.2MHz). Therefore, IEEE 802.11b can be regarded as Additive White Gaussian Noise (AWGN) for WBAN PHY. We can use standard AWGN analysis to derive N_0 in Equation (2.1).

Modeling Bluetooth interference is more challenging.

Bluetooth carries out GFSK modulation at 1MHz symbol rate. Let T_1 ($= 1\mu$s) denote the Bluetooth per symbol duration. Suppose a Bluetooth symbol starts at time 0, then its pass band complex equivalent signal is:

$$s(t) = \begin{cases} A_1 e^{j\phi(t)} e^{j2\pi f_c t}, & \text{when } t \in [0, T_1] \\ 0, & \text{otherwise} \end{cases}, \tag{2.3}$$

where phase $\phi(t)$ is given by

$$\phi(t) = \int_0^t 2\pi k_f b m(\tau) d\tau. \tag{2.4}$$

In Equation (2.4), k_f is a scaling constant, b is the bipolar information bit (± 1) the Bluetooth symbol represents, and $m(\tau)$ is the normalized Gaussian pulse.

Suppose a WBAN receiver receives both WBAN and interfering Bluetooth signals. As Bluetooth bandwidth and WBAN bandwidth are similar, we cannot simply model Bluetooth interference as AWGN. Rather, a finer granularity modeling is described in the following.

As Bluetooth and WBAN share identical carrier frequency specifications and similar bandwidth (the symbol duration of Bluetooth and WBAN are 1μs and 1.67μs respectively [69] [67]), the adjacent band interference from Bluetooth to WBAN is not a major

concern. Hence we focus on the case where both Bluetooth and WBAN use the same carrier frequency.

We can start from analyzing the interference from *one* Bluetooth symbol to *one* WBAN symbol.

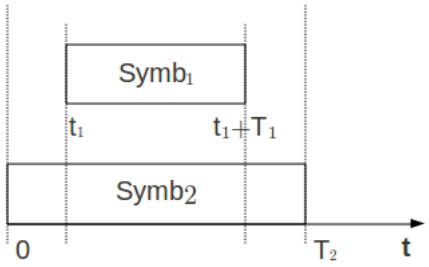


Figure 2.3: Temporal View of an Interfering Bluetooth Symbol and a Victim WBAN Symbol

Fig. 2.3 depicts the temporal relationship between an interfering Bluetooth symbol $Symb_1$ and a victim WBAN symbol $Symb_2$. Let T_1 ($= 1\mu$s) and T_2 ($= 1.67\mu$s) denote the duration of $Symb_1$ and $Symb_2$, respectively. Without loss of generality, suppose the $Symb_2$ spans $[0, T_2]$; and the $Symb_1$ spans $[t_1, t_1 + T_1]$. Since we do not need to consider multipath effects, only when $-T_1 < t_1 < T_2$ can $Symb_1$ interfere $Symb_2$ (see Fig. 2.3).[1]

Suppose at the WBAN receiver antenna, the WBAN signal carrier phase is 0, while the Bluetooth carrier phase is θ. Then the received in-phase and quadrature components from the Bluetooth symbol at time t are respectively

$$\begin{aligned} s^I(t) &= M(t)\sqrt{\frac{2}{T_1}}A_1 \cos[\phi(t-t_1)]\cos(2\pi f_c t + \theta), \\ s^Q(t) &= -M(t)\sqrt{\frac{2}{T_1}}A_1 \sin[\phi(t-t_1)]\sin(2\pi k_c t + \theta), \end{aligned}$$

where

$$M(t) = \begin{cases} 1, & \text{if } t \in [t_1, t_1 + T_1] \\ 0, & \text{otherwise} \end{cases}.$$

[1]Note since both Bluetooth and WBAN transmit at low rate (slower than 1M symbol/second), the wireless channel can be regarded flat. Hence we do *not* need to consider multipath effects.

Let n^{II} and n^{IQ} be the noise that $s^I(t)$ creates for the demodulation of $Symb_2$; and n^{QI} and n^{QQ} be the noise that $s^Q(t)$ creates for the demodulation of $Symb_2$. Then

$$\begin{aligned} n^{II} &= \frac{2A_1}{\sqrt{T_1T_2}} \int_a^b \cos[\phi(t-t_1)] \cos(2\pi f_c t + \theta) \cos(2\pi f_c t) dt \\ &= \frac{A_1}{\sqrt{T_1T_2}} \int_a^b \cos[\phi(t-t_1)][\cos(4\pi f_c t + \theta) + \cos\theta] dt \\ &= \frac{A_1}{\sqrt{T_1T_2}} \cos\theta \int_a^b \cos[\phi(t-t_1)] dt \end{aligned}$$

where

$$a = \begin{cases} \max\{0, t_1\}, & \text{when } -T_1 < t_1 < T_2 \\ 0, & \text{otherwise} \end{cases}, \tag{2.5}$$

$$b = \begin{cases} \min\{T_2, t_1 + T_1\}, & \text{when } -T_1 < t_1 < T_2 \\ 0, & \text{otherwise} \end{cases}. \tag{2.6}$$

Similarly, we have

$$\begin{aligned} n^{IQ} &= \frac{A_1}{\sqrt{T_1T_2}} \sin\theta \int_a^b \cos[\phi(t-t_1)] dt, \\ n^{QI} &= -\frac{A_1}{\sqrt{T_1T_2}} \sin\theta \int_a^b \sin[\phi(t-t_1)] dt, \\ n^{QQ} &= \frac{A_1}{\sqrt{T_1T_2}} \cos\theta \int_a^b \sin[\phi(t-t_1)] dt. \end{aligned}$$

where a and b are defined by Equation (2.5) and (2.6) respectively.

Thus, the final Bluetooth interferences received at in-phase and quadrature branches for demodulating $Symb_2$ are:

$$\begin{aligned} n^I &= n^{II} + n^{QI}, \\ n^Q &= n^{IQ} + n^{QQ}. \end{aligned}$$

The above single symbol jamming analysis can be easily extended to the symbol sequence case.

With the above method to quantify Bluetooth interference noise, we can use simulations to derive the bit error rate P_{ber} for a WBAN communication link. Note, since Bluetooth interference cannot be modeled as AWGN, we cannot use Equation (2.1) to derive P_{ber}.

2.3.3 Synchronization Error Rate Analysis

The first step for a WBAN receiver to receive a packet is to synchronize with the transmitter. This is done by testing multiple phase hypotheses on packet preamble in parallel. Without loss of generality, we assume a mainstream preamble hypothesis testing circuit as shown in Fig. 2.4 [73].

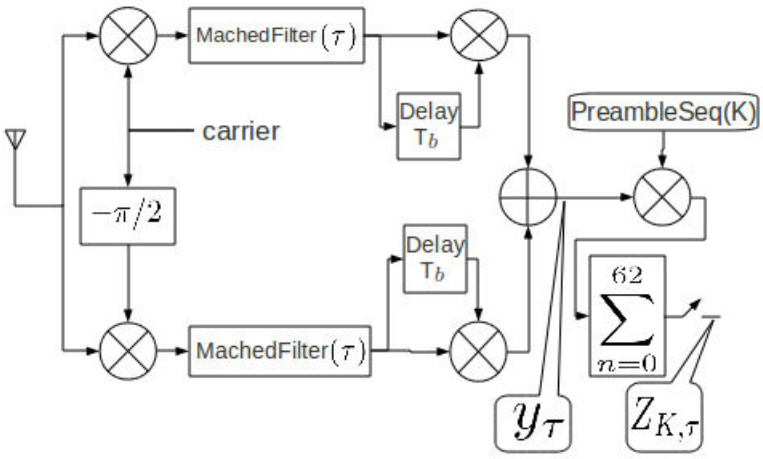


Figure 2.4: Synchronization circuit for testing one preamble phase hypothesis

In Fig. 2.4, $MatchedFilter(\tau)$ means the matched filter samples at time $iT_b + \tau$ ($i \in \mathbb{Z}$), where T_b is the preamble bit duration, and $\tau \in [0, T_b)$ is a fine-grained phase hypothesis. In practice, we try $\tau = 0$, $\frac{1}{4}T_b$, $\frac{2}{4}T_b$, and $\frac{3}{4}T_b$; $PreambleSeq(K)$ is the well-known first 63 bits of WBAN preamble, shifted cyclically by K bits. The circuit of Fig. 2.4 tests if the preamble phase is $KT_b + \tau$. If so, the output of $Z_{K,\tau}$ is maximized.

The circuit to produce y_τ in Fig. 2.4 depends on the PHY symbol modulation scheme. In our WBAN, it is $\pi/2$-DBPSK. Therefore, our interference analysis in Section 2.3.2 applies. Through MATLAB simulation, we can derive the synchronization error rate P_{pream} under WiFi/Bluetooth interference.

2.3.4 Channel Coding Analysis

After successful synchronization, the receiver needs to process the WBAN packet header and payload, which are encoded with 19/31 and 51/63 BCH code, respectively.

Both 19/31 and 51/63 BCH codes are light-weight FECs for correcting at most two error bits. The error rate $P_{word}(L)$ of a code word of L bits is:

$$P_{word}(L) = 1 - (1 - P_{ber})^L - C_L^1 P_{ber}(1 - P_{ber})^{L-1} \ldots \\ -C_L^2 P_{ber}^2 (1 - P_{ber})^{L-2},$$

where P_{ber} is Bit Error Rate (BER). Assume a segment, no matter header or payload, consists of N_w code words, the segment error rate P_{seg} is given by:

$$P_{seg}(N_w, L) = 1 - (1 - P_{word}(L))^{N_w}.$$

2.3.5 Packet Error Rate Calculation

Packet Error Rate (PER) P_{per} is obtained from the error rate of each segment: preamble, header, and payload.

The preamble error rate P_{pream} is derived through simulation (see Section 2.3.3).

The packet header has a length of 31 bits repeated four times (i.e., 124 bits in total), so the header error rate P_{header} is:

$$P_{header} = [P_{seg}(1, 31)]^4.$$

We assume the payload uses the mandatory $\pi/2$-DBPSK without repetition, and the packet length is 63×3 bits (i.e., coded from 51×3 information bits with 51/63 BCH coding), which is sufficient for most WBAN data packets in medical monitoring. The payload error rate $P_{payload}$ is then

$$P_{payload} = P_{seg}(3, 63).$$

Thus, the packet error rate P_{per} is

$$P_{per} = 1 - (1 - P_{pream})(1 - P_{header})(1 - P_{payload}).$$

2.4 Case Study

2.4.1 Simulation Scenario

In this section, we carry out a case study on multi-parameter monitoring using 2.4GHz WBAN under WiFi/Bluetooth interference. Fig. 2.5 illustrates the case study scenario. In the scenario, a centralized monitor periodically polls a patient's ECG electrodes through 2.4GHz WBAN. The distance between the monitor and the electrodes is d_2 (here we assume all electrodes have the same distance from the monitor). Meanwhile, the WBAN is interfered by two jamming sources. Both are d_1 away from the WBAN monitor. We study two cases: that the jamming sources are WiFi; and that the jamming sources are Bluetooth.

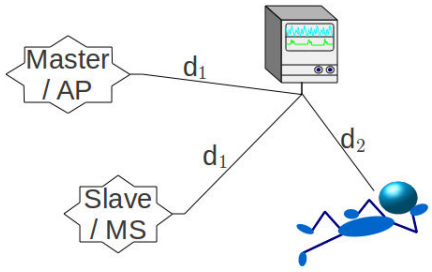


Figure 2.5: layout of simulation

For studying WiFi to WBAN interference, we assume one jamming source is an Access Point (AP), and the other is a Mobile Station (MS), as shown in Fig. 2.5.

We assume the WiFi nodes transmit at 30mW, a typical value adopted in practice [64] [24]. We assume WBAN electrodes transmit at 1mW. We are not particularly interested in knowing the WBAN monitor's transmit power due to the following reason. Our hypothesis is that WiFi *can* effectively interfere WBAN. To test this hypothesis, we need to make our evaluation *optimistic* on the WBAN side. Specifically, we assume WBAN downlink communications (i.e., from the monitor to electrodes) always succeed.

We assume both the WBAN and WiFi comply with the path loss model of Equation (2.2). For WBAN, we choose $\alpha_0 = 35.6901$ and $n = 1.81199$, which are derived from real-world measurement of [69]. For WiFi, we choose $\alpha_0 = 20.0542$ and $n = 2$ when $d_1 < 8$m; and $\alpha_0 = -4.5020$ and $n = 3.3$ when $d_1 \geq 8$m. This is a common model for WiFi evaluation [23].

We assume the WiFi AP and MS carry out continuous FTP under IEEE 802.11b 1Mbps, the most widely supported WiFi mode. The FTP data packet size is 1500 bytes (i.e., 12ms under IEEE 802.11b 1Mbps); and the WiFi RF band completely includes the WBAN RF band (here we assume the WBAN does not carry out frequency hopping; in case WBAN carries out frequency hopping, our scenario shall include three pairs of WiFi AP/MSs, which jam the whole 2.4GHz ISM band).

For studying Bluetooth to WBAN interference, we assume one jamming source is a Bluetooth Master, and the other is a Bluetooth Slave, as shown in Fig. 2.5.

We assume both the Bluetooth nodes and WBAN electrodes transmit at 1mW; while the monitor transmits at $100 \sim 1000$mW, as it is plugged to power cable, which provides sufficient power supply. For the time being, let us assume the WBAN downlink communications (i.e., from the monitor to electrodes) always succeed (we will discuss the impact of Bluetooth jamming sources to downlink communications later, see Section 2.4.5 footnote 2); and focus on the WBAN uplink communications (i.e., from the electrodes to monitor).

We assume both the WBAN and Bluetooth comply with the path loss model of Equation (2.2) with $\alpha_0 = 35.6901$ and $n = 1.81199$, which are derived from real-world measurement [69].

Our hypothesis is that Bluetooth *cannot* effectively interfere WBAN. To test this hypothesis, we need to make our evaluation *pessimistic* on the WBAN side. Specifically, we assume the Bluetooth Master is continuously transmitting to the Slave, and the Bluetooth frequency hopping is always coinciding with the WBAN RF band. Note this is an extremely pessimistic assumption. In reality, a Bluetooth Master/Slave link carries out TDMA with time slot duration of 625μs: 259μs of each 625μs time slot is idle, and every time slot has only $\frac{1}{79}$ chance of coinciding with WBAN RF band due to Bluetooth frequency hopping [67].

2.4.2 WBAN MAC Schedule

It is widely agreed that centralized polling is the proper MAC for medical multi-parameter monitoring [69]. Specifically, a polling period is called a *super frame*. A super frame

starts with a downlink beacon, followed by fixed TDMA time slots for (typically uplink) data packets.

In our case study, the WBAN consists of a monitor and four ECG electrodes sampling at 100Hz, a typical setting in ECG multi-parameter monitoring [2]. Each ECG electrode sample typically has 12 info bits [2], hence can be encapsulated into an uplink packet with PHY layer payload of 164 symbols. Under the $\pi/2$-DBPSK 600K symbol/s mode, such a packet takes 0.631ms to send (see Section 2.2).

The detail of our case study WBAN MAC schedule is depicted by Fig. 2.6. In the figure, a super frame consists of five slots of 2ms each. The 0th slot is for (downlink) beacon, the next four slots (Slot$1 \sim 4$) are assigned to the four (uplink) ECG electrodes. In each slot, an ECG packet (encapsulating one ECG sample) is repeated three times (see the zoom-in of Fig. 2.6). As such super frame lasts 10ms, we can upload 100 ECG samples per second for each ECG electrode (i.e., 100Hz sampling rate, a typical setting on ECG monitoring in medicine [35]).

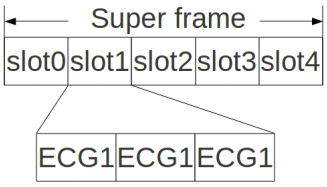


Figure 2.6: Schedule of a super frame

2.4.3 Mean Time To Failure Definition

For our case study of ECG multi-parameter monitoring, the *Mean Time To Failure* (MTTF) of WBAN depends on vital sign (in our case, ECG) sampling rate f_s and WBAN failure rate P_f^{BAN}. P_f^{BAN} depends on the number of electrodes n, and the failure rate of an individual electrode P_f, which, in turn, depends on packet error rate P_{per} and packet repetition times N_r. Thus, we have:

$$\begin{aligned} MTTF &= \frac{1}{f_s \times P_f^{BAN}}, \\ P_f^{BAN} &= 1-(1-P_f)^n, \\ P_f(i) &= (P_{per})^{N_r}. \end{aligned}$$

The packet error rate analysis are given before (see Section 2.3). The picking of f_s depends on medical domain specific knowledge. According to [35], when an ECG electrode works under monitoring mode, a reasonable sampling rate is $f_s = 100$(Hz).

2.4.4 Simulation Results on WiFi Interference

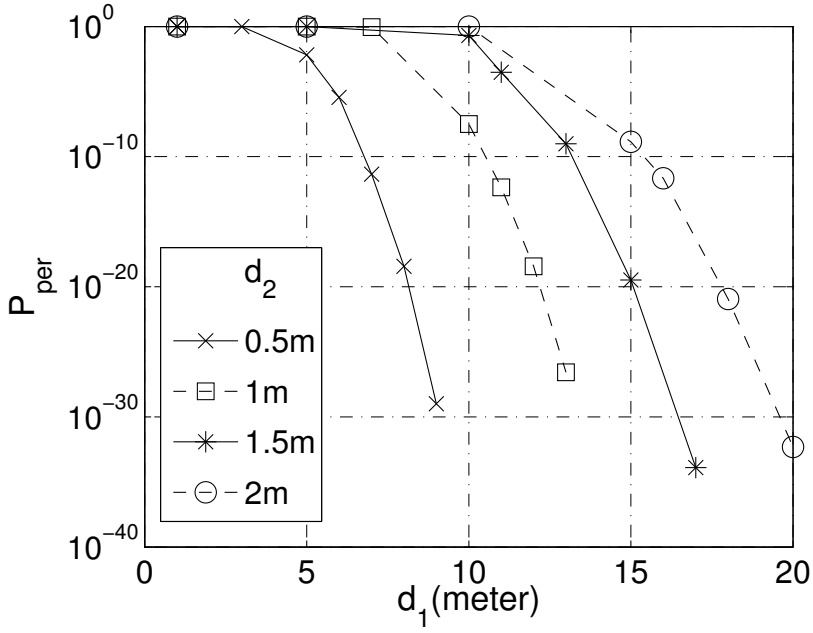


Figure 2.7: PER P_{per} of WBAN under WiFi interference. d_1 is the distance between WiFi jamming source and WBAN receiver; d_2 is the distance between WBAN transmitter and WBAN receiver (see Fig. 2.5).

Fig. 2.7 shows the WBAN PER (P_{per}) under WiFi interference when d_2 (distance from WBAN monitor to electrodes) equals 0.5m, 1m, 1.5m, and 2m respectively.

A more important metric is the whole ECG multi-parameter monitoring applica-

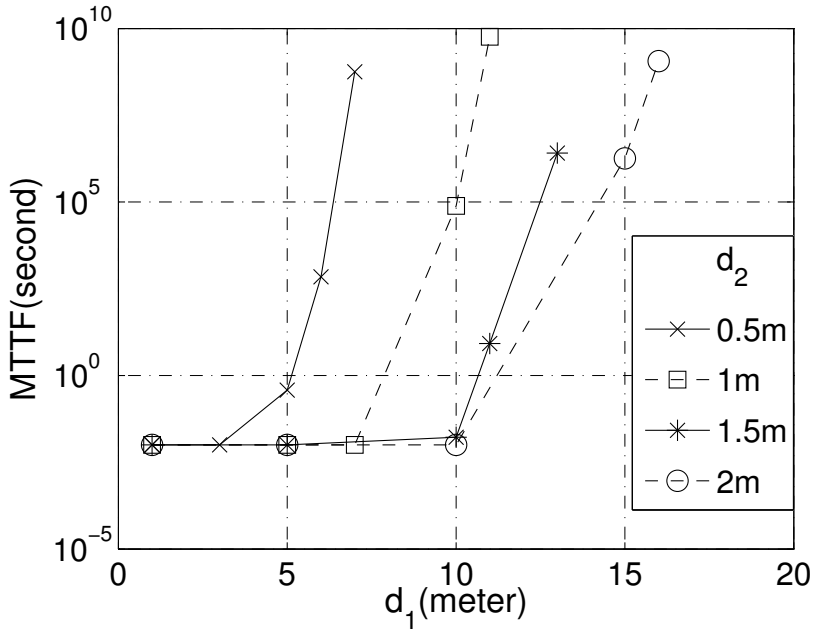


Figure 2.8: MTTF of WBAN under WiFi interference. d_1 is the distance between WiFi jamming source and WBAN receiver; d_2 is the distance between WBAN transmitter and WBAN receiver (see Fig. 2.5).

tion's MTTF. In wired ECG multi-parameter monitoring, various failures may happen. A typical scenario is electrode fall-off due to patient movement. Such failures are acceptable as long as MTTF is long enough, e.g., 3 hours.

In our WBAN case study, Fig. 2.8 plots the WBAN ECG multi-parameter monitoring MTTF under WiFi interference. According to the figure, even when $d_2 = 0.5$m (i.e., the received WBAN signal is very strong), the WiFi jamming source must be more than 6m away from WBAN receiver to guarantee an MTTF above 3 hours. When $d_2 = 2$m (i.e., the received WBAN signal is much weaker), the WiFi jamming source must be even farther away (more than 14m) from WBAN receiver. *This implies WiFi can effectively interfere WBAN*.

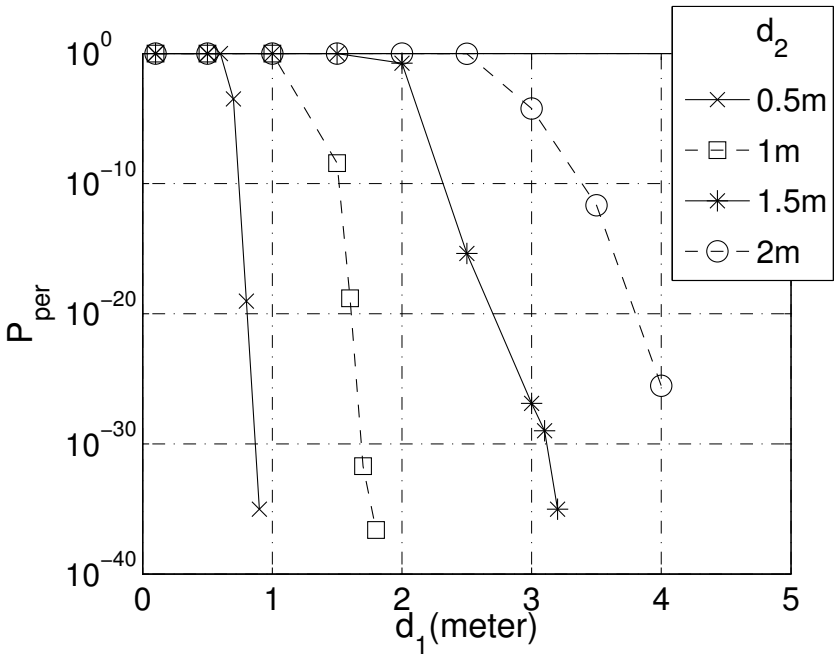


Figure 2.9: PER P_{per} of WBAN under Bluetooth interference. d_1 is the distance between Bluetooth jamming source and WBAN receiver; d_2 is the distance between WBAN transmitter and WBAN receiver (see Fig. 2.5).

2.4.5 Simulation Results on Bluetooth Interference

Fig. 2.9 shows the WBAN PER (P_{per}) under Bluetooth interference when $d_2 = 0.5$m, 1m, 1.5m, and 2m respectively. The figure shows even when $d_2 = 2$m (i.e., the received WBAN signal is weak), PER goes below 10^{-5} as long as $d_1 > 3$m.

Fig. 2.10 plots the WBAN ECG multi-parameter monitoring MTTF under Bluetooth interference. According to the figure, even when $d_2 = 2$m (i.e., the received WBAN signal is weak), the MTTF goes beyond 3 hours as long as the Bluetooth jamming source is more than 3.1m away from the WBAN receiver. When $d_2 = 0.5$m (i.e., the received WBAN signal is very strong), the Bluetooth jamming source only needs to be more than 0.7m away to achieve an MTTF more than 3 hours. *This implies Bluetooth interference is NOT a major threat to WBAN*[2].

[2]Note this conclusion assumes that downlink communications (from monitor to electrodes) always succeed (see Section 2.4.1). As the transmit power of monitor ($100 \sim 1000$mW) is much stronger than that

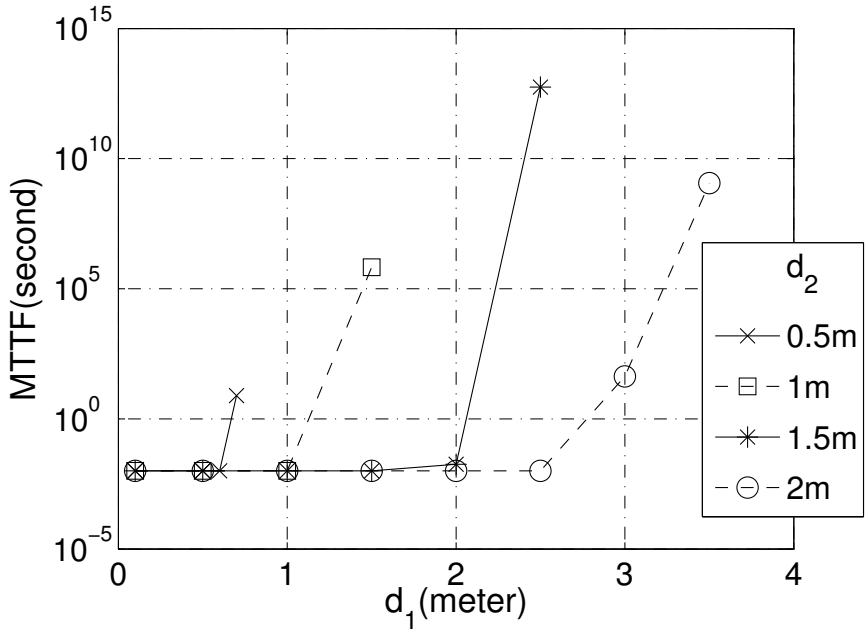


Figure 2.10: MTTF of WBAN under Bluetooth interference. d_1 is the distance between Bluetooth jamming source and WBAN receiver; d_2 is the distance between WBAN transmitter and WBAN receiver (see Fig. 2.5).

2.5 Related Work

IEEE 802.15.6 is a new standard of WBAN, so very few works study the coexistence of IEEE 802.15.6 2.4GHz standard and WiFi. Some works study the coexistence of WBAN and WiFi [21] [64] [63], but their WBAN used IEEE 802.15.4 (Zigbee). More importantly, their works only consider general requirement; while our work considers the medical requirement, such as sampling rate and MTTF. Some works [43] [24] study the performance of medical WBAN under contention, but their WBANs also use IEEE 802.15.4 and their focus is the contention *within* the WBAN devices.

The content of this chapter is published in [75].

of electrodes (1mW), through the same analysis we can see such assumption holds as long as the Bluetooth jamming source is reasonably away from the WBAN electrodes (e.g. more than 0.7m away when $d_2 = 0.5$m).

2.6 Summary

In this chapter, we evaluate the IEEE 802.15.6 2.4GHz standard under WiFi/Bluetooth interference in the context of medical multi-parameter monitoring. We conclude that WiFi poses a major threat to such application scenario, while Bluetooth does not.

CHAPTER 3

WICOP: ENGINEERING WIFI TEMPORAL WHITE-SPACES FOR SAFE OPERATIONS OF WIRELESS BODY AREA NETWORKS IN MEDICAL APPLICATIONS

3.1 Demand

In last chapter, we have shown that low power WBAN (using IEEE 802.15.6 2.4GHz standard) suffer from co-channel WiFi interference. Similar results [75] [44] [21] [24] [31] [32] [63] [25] show that WiFi poses a big threat to (Zigbee based) WBAN.

For instance, Liang et al. [44] experimentally show the performance degradation of Zigbee link under WiFi interference. In their experiments, *Packet Reception Rate* (PRR) of a ZigBee link drops below 20% when the ZigBee receiver is 15ft away from a WiFi interferer.

Though the coexistence interference may not be a major concern for low duty-cycle non-critical applications such as body temperature monitoring [19], it is not the case for WBAN applications with stringent requirements on packet delivery ratio and/or latency. One example is *Electrocardiography* (ECG) monitoring [16]. The IEEE 1073 [34] standard mandates that each ECG sample be delivered within 500ms [16]. A sample delivered after its 500ms deadline is considered lost, which means a fault happens.

To deal with the WBAN-WiFi coexistence challenge, three categories of solutions have been proposed. The first category of solutions aim to operate WBAN over RF channels sufficiently away from the active WiFi RF channels [19]. However, such solution does not cover cases where the ISM band is fully occupied (e.g., when there are two active non-overlapping IEEE 802.11n channels). The second category of solutions revise current WBAN or WiFi standards, adding intelligent coexistence schemes to

make WBAN or WiFi devices more aware of each other [44] [32]. However, the need to modify existing standards/implementations does not suit cases where *Commercially-Off-The-Shelf* (COTS) devices are used, or cases where interferers are non-cooperative. The third category of solutions try to spatially separate WBANs from WiFi networks via careful configuration-time planning. However, this does not deal with the case where WiFi networks are not under the same administration domain as WBANs. Furthermore, unintended usage of mobile WiFi devices may still cause spurious outages in WBANs[1].

In this chapter, we propose WiCop, a novel policing approach different from the aforementioned three categories of solutions. WiCop addresses the WBAN-WiFi co-existence problem by effectively controlling the temporal white-spaces (gaps) between consecutive WiFi transmissions. Though temporal white-spaces are abundant in light to medium loaded WiFi networks [44], they are scarce in heavy loaded WiFi networks and tend to be irregular. Our approach "engineers" the intervals and lengths of WiFi temporal white-spaces, and utilizes them to deliver low duty cycle WBAN traffic with minimum impacts on WiFi. WiCop exploits the *Clear Channel Assessment* (CCA) mechanisms in the WiFi standard. Two policing schemes are proposed: i) Fake-PHY-Header and ii) DSSS-Nulling. We have implemented and validated WiCop on SORA, a software defined radio platform. Experiments show that under WiFi interference, WiCop can raise WBAN packet delivery rates by up to 116%.

The rest of this chapter is organized as follows. Section 3.2 briefly introduces WiFi (IEEE 802.11) standard. Section 3.3 presents a case study showing the significance of WiFi co-channel interference on WBAN, using ECG monitoring as the medical application background. Section 3.4 proposes WiCop to engineer WiFi interference traffic's temporal white-spaces for WBAN communications. Section 3.5 analyzes the performance of WBAN under WiFi interference with/without WiCop protection. Section 3.6 evaluates WiCop through experiments. Section 3.8 summarizes this chapter.

[1] Repeated probe requests have been reported on certain WiFi devices when they are not associated with particular APs.

3.2 Background of WiFi

Before delving into the details of WiCop, we first give an overview of the WiFi (aka IEEE 802.11) standard. The WiFi standard boils down to several subtype standards, of which, most of nowadays COTS WiFi devices comply with the subtype standard of IEEE 802.11a, b, g, or n. IEEE 802.11b is the first to reach mass production, which runs *Direct Sequence Spread Spectrum* (DSSS) in the 2.4GHz ISM band. IEEE 802.11a emerges next, and runs *Orthogonal Frequency Division Multiplexing* (OFDM) in the 5GHz ISM band, a less frequently used RF band due to more stringent line-of-sight transmission constraints. IEEE 802.11g supports IEEE 802.11a-like OFDM in the 2.4GHz ISM band, meanwhile is fully backward compatible with IEEE 802.11b. IEEE 802.11n mainly enhances the previous three by adding *Multiple Input Multiple Output* (MIMO) antenna support.

In the following, we shall only look at those common features of IEEE 802.11a/b/g/n that are critical to our WiCop strategies.

Full Occupation of 2.4GHz ISM Band Every WiFi subtype standard predefines a fixed set of RF channels. Though a single WiFi network can only use one of these predefined RF channels, when several WiFi networks coexist in an area, they will try or will be configured to use non-overlapping RF channels. This can easily exhaust the whole 2.4GHz ISM band. For example, two coexisting IEEE 802.11n networks are enough to occupy the whole 2.4GHz ISM band. Such scenario is not uncommon nowadays given the ubiquitous presence of WiFi networks. When all such WiFi networks are active, jamming the whole 2.4GHz ISM band, it is hard to carry out WBAN communications, no matter the WBAN uses ZigBee, Bluetooth, or the draft IEEE 802.15.6 2.4GHz standard.

Common Packet Formats Due to backward compatibility considerations, all subtypes of WiFi running in 2.4GHz ISM band recognize the IEEE 802.11 1Mbps packet format, which is one of the basic data rates of 802.11b.

Viewing from the *Physical Layer* (PHY), we can abstract an IEEE 802.11 1Mbps packet as four consecutive segments (see Fig. 3.1): preamble, *Start Frame Delimiter* (SFD), PHY header, and PHY payload[2].

[2]which correspond to *Physical Layer Convergence Protocol* (PLCP) SYNC bits, SFD, PLCP header, and *MAC Protocol Data Unit* (MPDU) respectively according to the standard jargon [68].

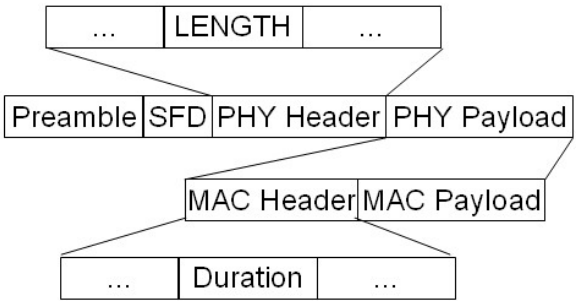


Figure 3.1: IEEE 802.11 1Mbps PHY packet format.

The preamble is for receiver carrier acquisition, made up of 128 consecutive '1'.

SFD is a 16-bit field indicating the successive PHY header.

The 48-bit PHY header contains several fields that carry control/management information. What is important is the LENGTH field, a 16-bit unsigned integer indicating the number of microseconds required to transmit the PHY payload. This implies a maximum of $2^{16} = 65535\mu$s can be reserved for PHY payload.

The PHY payload usually consists of MAC header and MAC payload. These two parts have variable length. For example, an *Ready To Send* (RTS) packet has a 160-bit MAC header and has no MAC payload. The RTS packet has a *Duration* field in MAC header to claim a sequence of WiFi transmissions, lasting up to 32767μs.

Common Receiver Diagram Due to backward compatibility considerations, all subtypes of WiFi should have a compatible receiver to decode 802.11 1Mbps DSSS signal. The receiver diagram is shown by Fig. 3.2. First, *RX Filter* retrieves chips from raw samples. Second, *slicer* detects bit timing by picking the max energy. Third, *demode* retrieves one bit from every 11 chips. Fourth, *decode* is responsible for searching and processing preamble, PHY header and MAC header.

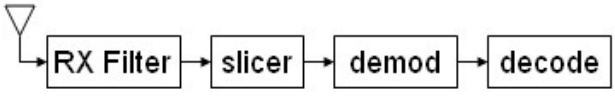


Figure 3.2: diagram on receiving and decoding 802.11 1Mbps DSSS signal

Clear Channel Assessment (CCA) All subtypes of WiFi carry out *Carrier Sense Multiple Access* (CSMA) MAC protocol. According to CSMA, an IEEE 802.11 node shall always listen to the wireless medium before transmission. Only when the wireless medium is idle will the node start transmitting. This procedure is called *Clear Channel Assessment* (CCA).

There are three types of CCA: *Energy Detection* (ED) only, *Carrier Sense* (CS) only, and ED+CS (the combination of ED and CS). ED-only CCA measures the wireless medium spectral power level; if it is greater than a threshold, the wireless medium is considered busy. CS-only CCA tries to capture WiFi PHY preambles; if a PHY preamble is successfully captured, the wireless medium is considered busy. Usually, CS-only CCA also looks into the content of the PHY header immediately following the captured PHY preamble (if there is one) to provide more accurate CCA evaluations. ED+CS CCA does both. In practice, most WiFi devices support CS-only CCA or ED+CS CCA [26] [68].

3.3 A Case Study on ECG Monitoring

In this section, we study the performance of a ZigBee WBAN for ECG monitoring under WiFi interference, so as to empirically show the necessity of addressing the WBAN-WiFi coexistence problem.

3.3.1 Experiment Setup

Fig. 3.3 shows the layout of the experiment. The ECG monitoring WBAN consists of one base station and one ECG sensor, implemented by two TMote Sky nodes (aka *motes*, a well-known ZigBee device) [82] respectively. In Fig. 3.3, the base station is denoted as *Mote-B*, and the ECG sensor is denoted as *Mote-C*; the distance between *Mote-B* and *Mote-C* is d_2. The transmission power of *Mote-B* and *Mote-C* is set to the maximum: 0dBm. *Host-Z* is a laptop connected with *Mote-B* through USB for data collection. *Host-I* is the WiFi interferer, implemented by a Linux laptop with Intel Pro/Wireless 3945ABG WiFi chip; while WiFi *Access Point* (AP) is a LinkSys WRT54GL WiFi router. *Host-I* sends packets to WiFi AP via an IEEE 802.11g link. The transmission power of *Host-I* is 30mW, a typical value adopted in practice [24]. The distance from *Host-I* to *Mote-*

B and *Mote-C* are both d_1. In addition, *Host-M* is connected to the WiFi AP to record WiFi interference traffic between *Host-I* and the WiFi AP. An additional WiFi *sniffer* is deployed which passively logs WiFi events on the wireless medium. *Host-P* runs WiCop and is not used in this experiment.

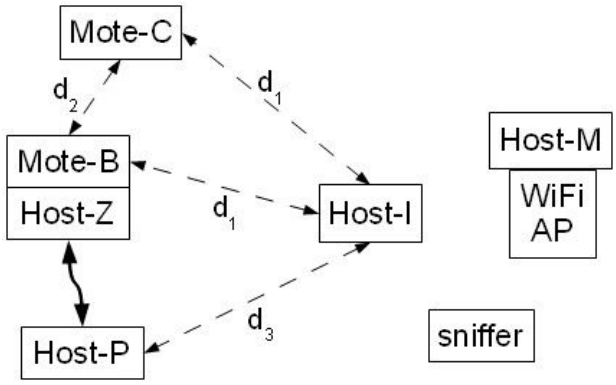


Figure 3.3: Experiment Layout

Upon reception of ECG samples from the ECG sensor, the ECG base station reconstructs the ECG signal. The sampling rate of ECG signal is 250Hz, a typical value for ECG monitoring [2]; and each sample is 8-bit. The ECG sensor (*Mote-C*) sends the base station (*Mote-B*) one packet every 100ms. Hence each packet contains 250Hz $\times$ 100ms = 25 new ECG samples, which we call an *ECG sample chunk*. In addition, to increase reliability, the ECG sensor (*Mote-C*) buffers the two immediate previous ECG sample chunks, which are sent together with the new chunk in the same packet. Therefore, each packet contains 3 ECG sample chunks, i.e., $25 \times 3 = 75$ ECG samples; and every ECG sample is transmitted 3 times. At the typical ZigBee raw bit rate of 250kbps, the transmission time cost of each packet is less than 4ms.

3.3.2 Performance Metric

To evaluate the performance of ECG monitoring under WiFi interference, we consider three metrics. The first metric is *Packet Reception Rate* (PRR), defined as the probability that a packet is successfully received.

Let $T_{polling}$ denote the ECG packet transmission period ($T_{polling} = 100$ms in our case study). As mentioned before, ECG samples are only transmitted in the grouping of

ECG sample chunks; and each ECG sample chunk is transmitted $N_{re} = 3$ times within $T_{polling} \times N_{re} = 300$ms (which is within the typical ECG sample delivery deadline [16]). An ECG sample chunk is lost iff it fails all its N_{re} transmissions. A chunk loss is defined as a failure.

With the definition of failure, we introduce the second metric, *Mean Time To Failure* (MTTF), which is the expected duration between two ECG sample chunk losses. MTTF is given by (see Section 3.5.6 for detail):

$$MTTF = \frac{T_{polling}}{PER^{N_{re}}}, \tag{3.1}$$

where $PER \stackrel{def}{=} 1 - PRR$.

The third metric is *Mean Time To Recovery* (MTTR), which is the expected duration of failures. MTTR is equal to (see Section 3.5.6 for the derivation):

$$MTTR = T_{polling}/PRR. \tag{3.2}$$

3.3.3 Experiment Results and Observations

With the layout set as Fig. 3.3, we let *Host-I* transmit at an application layer rate of 30Mbps to the WiFi AP to emulate WiFi interference.

We set d_2 to 4ft. As the distance from *Host-I* to *Mote-B* (i.e., d_1) changes from 12ft to 4ft, the PRR decreases from 98% to 67% (see Fig. 3.4). At 67% PRR, the MTTF is around 2.8s. In other words, on average every 2.8s, an ECG sample chunk may be lost, which is a serious problem. The MTTR performance shows a similar trend. As the distance from *Host-I* to *Most-B* changes from 12ft to 4ft, MTTR increases 15% (see Fig. 3.5).

3.4 Illustration of WiCop

3.4.1 Architecture

The case study in Section 3.3 identifies WiFi interference as an eminent threat to WBAN reliability. This is consistent with the conclusions of the literature on 2.4GHz ISM band

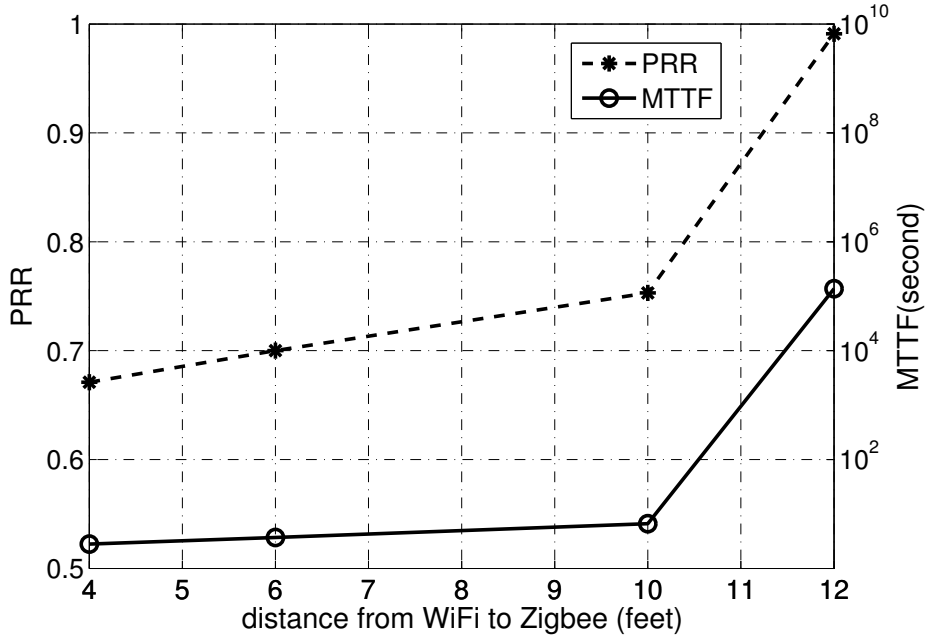


Figure 3.4: PRR and MTTF of ECG monitoring WBAN under 802.11g interference

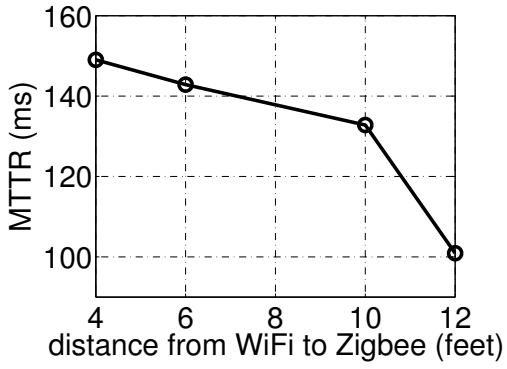


Figure 3.5: MTTR of WBAN under 802.11g interference

WBAN coexistence [44] [32] [24]. In fact, due to the low power nature of other mainstream 2.4GHz ISM band technologies (ZigBee, Bluetooth, IEEE 802.15.6 proposal etc.), and the ubiquitous presence of WiFi networks, WiFi stands out as the major threat to 2.4GHz ISM band WBAN coexistence reliability. This motivates us to devise a policing approach, called WiCop, to curb the WiFi threat. As mentioned in Section 3.1, firstly, WiCop shall force WiFi communications to pause at proper time, leaving temporal white-spaces for WBAN to communicate. Secondly, WiCop shall require no changes to COTS WiFi devices, nor COTS WBAN devices. Thirdly, to allow cross layer design, and to achieve high adaptability, WiCop shall reside upon *Software Defined Radio* (SDR) platform. In this chapter, WiCop uses the SORA platform [70], an SDR platform developed by Microsoft Research.

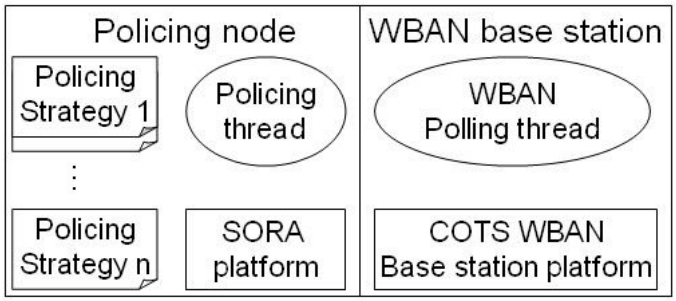


Figure 3.6: WiCop architecture: the policing node and the WBAN base station can reside in a same host, or two separate but synchronized hosts

The WiCop architecture is illustrated by Fig. 3.6. The architecture involves two core entities, the policing node and the WBAN base station. The centerpiece of the policing node is the *WiCop policing thread* running upon an SDR platform, e.g., SORA [70]. The centrepiece of the WBAN base station is the *WBAN polling thread* running upon a COTS WBAN base station platform, e.g., TMote Sky [82]. This polling thread periodically polls remote WBAN client electrode(s)/actuator(s) for data/actuation. As already mentioned in Section 3.3, we call the corresponding period the *WBAN polling period*, denoted as $T_{polling}$.

The policing node and the WBAN base station shall reside in a same host, or two well synchronized hosts. At the beginning of each WBAN polling period, the policing thread would first load the SORA platform with a specific policing strategy, which will be further explained in Section 3.4. When the policing strategy is active, the policing thread

triggers the WBAN polling thread to start polling the WBAN (for this specific WBAN polling period).

We call the temporal interval for a WBAN base station to finish one round of polling the *WBAN active interval*. Each WBAN polling period includes one WBAN active interval; the rest of the period is called *WBAN idle interval*. Usually, the WBAN polling period is much longer than the WBAN active interval, leaving enough WBAN idle interval for WiFi or other coexisting wireless schemes.

With all the above concepts in mind, we can proceed to propose various policing strategies.

3.4.2 WiCop Policing Strategies

The basic idea of all our proposed WiCop policing strategies is to exploit the WiFi *Clear Channel Assessment* (CCA) mechanisms: by sending engineered WiFi compliant signals, we can properly administrate WiFi transmissions.

Strategy I: Fake-PHY-Header

Policing Signal As mentioned in Section 3.2, at PHY layer, a WiFi packet transmission begins with a PHY preamble, followed by a PHY header, and then DATA. The PHY header carries a LENGTH field (see Fig. 3.1): a 16-digit unsigned integer specifying the number of microseconds that the WiFi packet lasts.

According to WiFi CCA specifications, when another WiFi device detects a PHY preamble and decodes the following PHY header, it will mute (i.e., refrain from transmitting) for a number of microseconds depending on the received LENGTH field and the device's specific implementation. Therefore, the LENGTH field plays the role of reserving wireless medium access for its WiFi packet.

As the LENGTH field is a 16-bit unsigned integer, in theory, a maximum of 65535μs can be reserved for the corresponding WiFi packet. However, our calibration measurements show that the actual maximum duration that can be reserved is vendor dependent, as shown in Fig. 3.7. Fortunately, Fig. 3.7 also show all WiFi devices from major

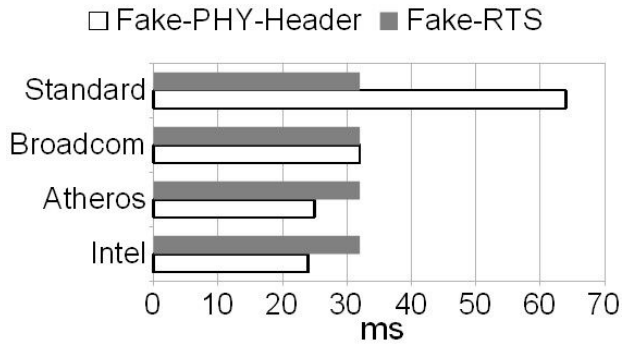


Figure 3.7: Maximum duration a WiFi device mutes upon receiving a Fake PHY Header policing packet and a Fake RTS policing packet (please see Section 3.4.2) respectively

vendors can mute for at least 24ms. This is enough for reserving temporal white-spaces for typical WBAN communications. For example, in ECG WBAN monitoring, with each WBAN packet containing 75 8-bit samples, a WBAN needs no more than 4ms to send a packet from the ECG sensor to the base station.

MAC Protocol We propose to exploit the aforementioned LENGTH field to administrate coexisting WiFi transmissions. To do this, the WiCop policing node and the WBAN base station must carry out a coordinated *Multiple Access Layer* (MAC) protocol, as explained by Fig. 3.8(a).

According to Fig. 3.8(a), each WBAN polling period starts with the policing node's broadcast of a so called *Fake-PHY-Header policing signal*: a fake WiFi packet with only PHY preamble and PHY header. Although this fake WiFi packet does not have DATA segment, its PHY header's LENGTH field claims a packet duration equivalent to the temporal length of the WBAN active interval (hence "faking"). Immediately following this Fake-PHY-Header policing signal, the WBAN active interval starts, during which the WBAN base station polls its client(s).

The intuition of Fake-PHY-Header policing is that on hearing the Fake-PHY-Header policing signal, a WiFi interferer will mute for the following WBAN active interval, creating a temporal white-space for WBAN to communicate.

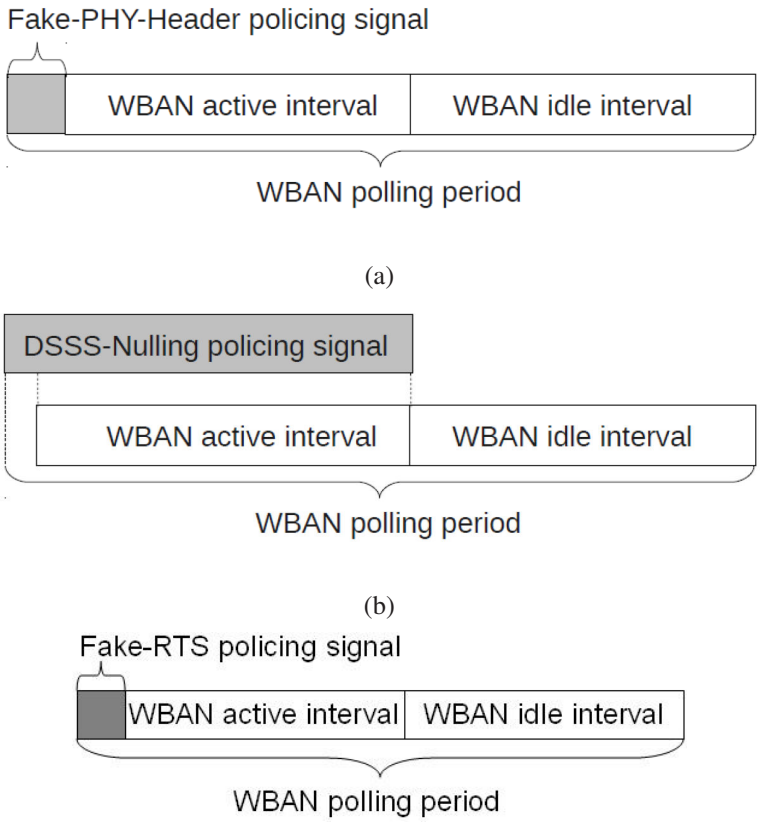


Figure 3.8: Temporal domain schemes: (a) Fake-PHY-Header policing; (b) DSSS-Nulling policing; (c) Fake-RTS policing

Strategy II: DSSS-Nulling

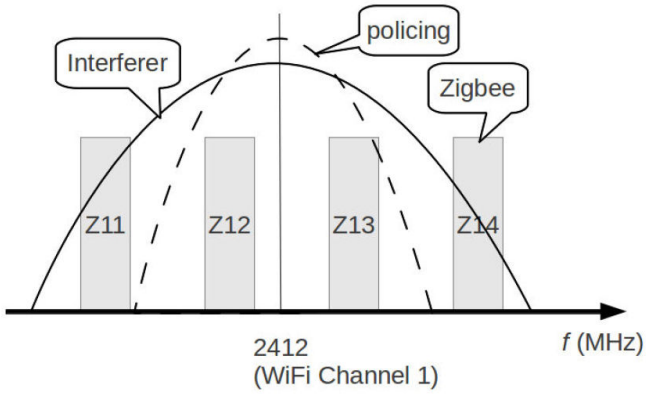


Figure 3.9: Power Spectral Density (PSD) of interferer, policing, and ZigBee

Policing Signal It is well-known that continuously sending repeated WiFi PHY preambles can jam other WiFi devices' transmissions [26] [79]. Since WiFi PHY preamble is a DSSS modulated signal, we call the continuous sending of repeated WiFi PHY preamble "*DSSS-Jamming*". We intend to use DSSS-Jamming as another means of policing. However, DSSS-Jamming not only jams WiFi devices, it also jams co-channel WBAN devices. To solve this problem, we reshape the DSSS-Jamming signal with a band-pass filter to generate the desired policing signal. We call such generated policing signal *DSSS-Nulling policing signal* (i.e., the sides of the DSSS-Jamming signal spectrum are "nulled" to create spaces for WBAN signals), and the corresponding policing scheme the *DSSS-Nulling* policing.

Fig. 3.9 compares the *Power Spectral Density* (PSD) of DSSS-Nulling signal, WiFi signal, and ZigBee signal. When a DSSS-Nulling signal is present, a WiFi device thinks the carrier is busy and backs off. In contrast, as DSSS-Nulling signal does not occupy ZigBee channel $Z11$ and $Z14$, ZigBee communications are still possible.

In our prototype implementation, the band-pass filter to reshape DSSS-Jamming signal is a raised cosine *Finite Impulse Response* (FIR) filter, which results in a DSSS-Nulling signal bandwidth of 8MHz (in comparison, WiFi signal bandwidth is 22MHz). MATLAB simulations show that the side lobe of this filter is -55dB (Fig. 3.10). In other words, we reduce the interference power to WBANs by 55dB.

Alternatively, one can use other forms of noise signal (e.g., simply a sine wave) in

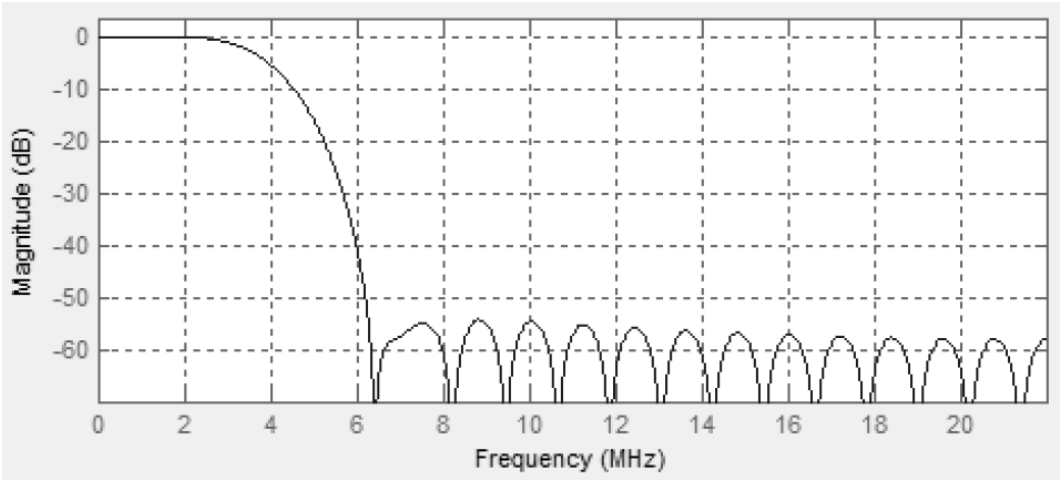


Figure 3.10: Frequency response of the FIR that reshapes DSSS-Jamming signal into DSSS-Nulling signal (baseband equivalent spectrum)

the WiFi band to jam/police WiFi transmission. However, as DSSS-Nulling signal carries repeated WiFi PHY preamble information (though distorted by the band-rejection filter), it can more effectively jam WiFi devices that support CS-only or ED+CS CCA. Based on Tanenbaum and Wetherall [71], DSSS-Nulling signal can use at least 20dB less power than any other forms of noise in jamming an ED+CS CCA WiFi device.

MAC Protocol Same as the Fake-PHY-Header policing case, DSSS-Nulling policing still assumes the WBAN runs centralized polling and the policing node resides on the same host as (or is synchronized to) the WBAN base station. But instead of *preceding* each WBAN active interval, the DSSS-Nulling policing signal persists throughout the WBAN active interval as shown by Fig. 3.8(b).

Strategy III: Fake-RTS

Policing Signal Similar to Fake-PHY-Header, we can extend the policing strategy to MAC layer. Instead of using a fake LENGTH field of PHY header, we transmit a IEEE 802.11b *Request-To-Send* (RTS) packet [65]. Similar to the LENGTH field of PHY header, the RTS packet has a ”Duration” field to claim that a sequence of WiFi packet-exchange is starting, which will last up to 32767μs. Most COTS WiFi devices respect

RTS packets (see Fig. 3.7). On receiving such an RTS packet, these WiFi devices will remain silent for the claimed duration. However, like Fake-PHY-Header policing, the RTS claim is fake: no subsequent WiFi packet-exchange will actually happen. The WiCop policing node will instead use the claimed duration as the WBAN active interval. We henceforth call this fake RTS packet the *Fake-RTS policing signal*, and this policing strategy *Fake-RTS* policing[3].

MAC Protocol Similar to Fake-PHY-Header, the temporal view of Fake-RTS policing is shown in Fig. 3.8(c).

3.4.3 Qualitative Comparisons of Policing Strategies

Every policing strategy has its pros and cons. Table 3.1 qualitatively compares the aforementioned three policing strategies.

CCA Compatibility DSSS-Nulling is the most versatile. It works with all WiFi devices, no matter they support CS-Only, ED-Only, or ED+CS CCA. In contrast, Fake-PHY-Header and Fake-RTS policing both requires the interfering WiFi devices support CS based CCA. Fortunately, most main-stream WiFi adaptors nowadays support CS based CCA [26] [44], hence ensure Fake-PHY-Header and Fake-RTS's viability.

Success Rate All three policing strategies have high success rate in suppressing interfering WiFi transmissions (see Section 3.6) when wireless channel quality is lenient.

Under poor wireless channel quality, however, DSSS-Nulling has the highest success rate in suppressing interfering WiFi transmissions. This is because DSSS-Nulling policing retransmits IEEE 802.11b preambles throughout the WBAN active interval. The

[3]It is brought to our attention recently that Hou et al. [31] is in fact the first to propose the Fake-RTS policing strategy (in the form of fake CTS to be exact), though we proposed the strategy independently. Nevertheless, we are the first to implement this strategy on an SDR platform; and by exploiting the flexibility of SDR, we integrate this strategy as one of the runtime alternatives in a holistic framework. We are also the first to compare this strategy with other strategies in the context of medical applications.

Table 3.1: Qualitative Comparisons of Policing Strategies

Policing Strategy	Fake-PHY-Header	DSSS-Nulling	Fake-RTS
CCA Compatibility	CS-Only, ED+CS	CS-Only, ED+CS, ED-Only	CS-Only, ED+CS
Success Rate	High	Highest	High
Temporal-Spectral Overhead	Lowest	Large	Low
Platform Requirement	high	high	Low

retransmissions enhance reception. In contrast, Fake-PHY-Header and Fake-RTS have no retransmission mechanisms to improve reception. For Fake-PHY-Header to work, the received policing signal's PHY layer checksum must be correct. For Fake-RTS to work, it is even harder: both of the received policing signal's PHY and MAC layer checksums must be correct.

Temporal-Spectral Overhead We define overhead ratio with

$$\rho = \frac{\text{Time-Spectrum Overhead}}{\text{Time-Spectrum Reserved for WBAN}},$$

and the ratios of each policing strategies are defined as follows.

In each WBAN polling period, there only needs to be one Fake-PHY-Header broadcast, which occupies 22MHz of spectrum (the standard WiFi PHY preamble/header spectrum bandwidth) and 0.2ms[4]. Such a broadcast allows 4 ZigBee channels to communicate throughout one WBAN active interval. Therefore, the overhead ratio of Fake-PHY-Header policing is

$$\rho_{fph} = \frac{22 \times 0.2}{4B_z \times T_{act}} = \frac{1.1}{B_z T_{act}}, \tag{3.3}$$

[4]The more exact duration of a Fake-PHY-Header policing frame is 0.192ms, assuming IEEE 802.11 1Mbps DSSS modulation and long preamble [68].

where B_z(MHz) is the bandwidth of a Zigbee channel, and T_{act}(ms) is the length of WBAN active interval.

Similarly, the overhead ratio of Fake-RTS policing is

$$\rho_{fr} = \frac{22 \times 0.4}{4B_z \times T_{act}} = \frac{2.2}{B_z T_{act}}, \tag{3.4}$$

based on the fact that a fake RTS packet takes 0.4ms[5].

Suppose the effective DSSS-Nulling policing signal needs 8MHz of spectrum [6]; and must persist throughout the WBAN active interval. This implies a DSSS-Nulling policing signal can only help reserve two Zigbee channels throughout the WBAN active interval. Therefore, the overhead ratio of DSSS-Nulling policing is

$$\rho_{dn} = \frac{8 \times T_{act}}{2B_z \times T_{act}} = \frac{4}{B_z}. \tag{3.5}$$

As T_{act} is usually 4ms $\sim$ 40ms, Formulae (3.3), (3.4), and (3.5) imply Fake-PHY-Header and Fake-RTS incur much lower overhead ratio than DSSS-Nulling, given that the policing is successful.

The overhead ratio of Fake-RTS policing is a little higher than that of Fake-PHY-Header, as Fake-RTS frame contains a MAC header in addition to the PHY header.

Platform Requirement Both Fake-PHY-Header and DSSS-Nulling requires SDR platform; while Fake-RTS only requires commercial WiFi adaptor with soft MAC function [31].

3.4.4 Impact to WiFi

WiCop does little harm to WiFi transmission due to the following reasons.

First, WiCop carries out ED CCA (see Section 3.2) before transmitting policing signals. This guarantees WiCop policing signal does not preempt existing WiFi trans-

[5]The more exact duration of a Fake RTS policing signal is 0.352ms, assuming IEEE 802.11 1Mbps DSSS modulation and long preamble [68].

[6]Note that the best bandwidth of DSSS-Nulling signal is out of the scope of this thesis.

missions[7]. Furthermore, both the Fake-PHY-Header and the DSSS-Nulling policing signal follow WiFi preamble/header formats. Therefore, from WiFi devices' perspective, a WiCop policing node behaves just like another WiFi device.

Second, medical WBAN traffic is typically of low duty-cycle and low workload [66]. For example, the WBAN polling period for ECG monitoring is typically 100ms; and during this 100ms, only 5ms is for WBAN traffic (and under WiCop policing). The remaining 95ms interval can be used for WiFi communications.

3.4.5 Implementation of Policing Thread

We implemented policing thread of WiCop upon *Microsoft Research Software Radio* (SORA) [70] platform. A SORA platform consists of the following hardware: a desktop computer (denoted as *Host-P* in Fig. 3.3), a *Radio Control Board* (RCB), and a third-party radio daughter board. The radio daughter board that we use is USRP XCVR2450.

Correspondingly, the SORA platform software mainly consists of the various software defined radio drivers and the corresponding development tools. For WiCop, we implemented the aforementioned policing strategies upon SORA Soft WiFi driver v1.0 (simplified as "*SORA driver*" in the following). The details are as follows.

As shown by Fig. 3.11, in order to transmit a policing signal, WiCop sends a policing packet down through the SORA stack, which involves five layers (including three layers in the SORA driver: *Link Layer* (LL), MAC, and PHY). Each layer carries out special processing of the policing packet.

At the application layer (denoted as "Police App" in Fig. 3.11), WiCop customizes the payload of the policing packet according to the specific policing strategy used. For Fake-PHY-Header policing or Fake-RTS policing, the policing packet payload is nulled. For DSSS-Nulling policing, the policing packet payload length is adjusted according to WBAN active interval length, and the payload digits are all set to one. At the network layer (denoted as "UDP socket" in Fig. 3.11), a special IP/MAC address is used to flag the policing packet. In the LL layer, upon detecting the flagged IP/MAC address, we add

[7] As a WiFi packet typically lasts less than 1ms [68], the incurred backoff of WiCop policing signal has little impact on WBAN performance, as the typical medical WBAN polling period is ≥ 100ms.

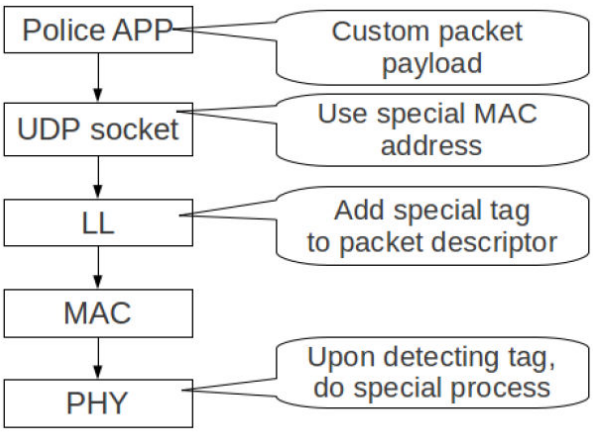


Figure 3.11: Procedure of sending a policing packet

special tags to the policing packet's descriptor (a data structure in SORA to record packet information). In the MAC layer, policing packets' backoff is deliberately shortened (to less than standard IEEE DIFS) to achieve a higher priority when contending with WiFi interferers. In the PHY layer, special processing is done according to the tag in the policing packet's descriptor. For Fake-PHY-Header policing packet, we customize the LENGTH field to cover the whole WBAN active interval. For DSSS-Nulling policing packet, we apply the band-rejection filter to null its side spectrum. To realize the WiCop policing strategies, the policing thread must work with the WBAN base station simultaneously. In our experiment set up (see Fig. 3.3), this is achieved by wiring the policing node (*Host-P*) and the WBAN base station (*Mote-B*) host (*Host-Z*) with high speed Ethernet.

3.5 Performance Analysis

In this section, we analyze the performance of ZigBee network under WiFi interference with or without policing strategies. For consistency, but without loss of generality, the analysis in this section still assumes the network layout of Fig. 3.3.

We assume WiFi interferer performs CS-CCA, so WiFi interferer cannot detect ZigBee transmissions. Similar assumption has been made in [32] (please refer to Appendix.3 for more discussion). For ease of analysis, the packet inter-arrival time and packet duration of WiFi interferer are assumed to be constant $1/\lambda$ and $1/\mu$ respectively, where $1/\lambda > 1/\mu$. More sophisticated stochastic models can be devised under general distribu-

tions but omitted in this thesis, as the objective of the analysis is just to gain insights on average performance.

A polling based MAC protocol is adopted in ZigBee networks with a polling interval $T_{polling}$. At the beginning of each polling interval, the WBAN base station broadcasts a beacon containing a transmission schedule (for guaranteed access). Upon receiving this beacon, clients upload their respective data one by one in a batch. We suppose the downlink (from base station to clients) is free of error since the WBAN base station usually has larger transmit power, while the uplink (from clients to the WBAN base station) is susceptible to WiFi interference. We denote the duration of transmitting a ZigBee (uplink) packet as T_{pkt}. According to our configurations, $T_{pkt} > 1/\lambda > 1/\mu$, which is a common scenario in practice.

In the analysis, policing signals are encoded according to 802.11b 1Mbps DSSS mode. Moreover, ED-CCA is used before channel access. Thus, we assume policing signals do not preempt existing WiFi transmissions.

The rest of this section is organized as follows. First, we give the PRR of a ZigBee WBAN under WiFi interference without policing. Second, we inspect how the preamble of policing signal delays WiFi transmissions. Next, we analyze the PRR of ZigBee WBAN under WiFi interference with the three policing strategies respectively. Finally, the analytical forms for WBAN's MTTF and MTTR are derived.

3.5.1 PRR with No Policing

The PRR of ZigBee WBAN under WiFi interference can be mainly attributed to two factors: the *Bit Error Rate* (BER) under WiFi interference, and the number of ZigBee bits interfered. For simplicity, BER in absence of WiFi interference is assumed to be 0.

Since the WiFi transmission bandwidth (denoted as B_w) is much larger than the bandwidth of ZigBee (denoted as B_z), a WiFi interferer can be viewed as a white noise source in the pass band of ZigBee [63] [75]. Let $P^z_{tx}, P^w_{tx}, P^z_{rx}, P^w_{rx}$ be the transmitted signal power and received signal power of the ZigBee transmitter and WiFi interferer (the received signal power from WiFi corresponds to the energy in the pass band of ZigBee) respectively. Let distance from the ZigBee WBAN base-station and the ZigBee client be d_2 and the distance from the WiFi interferer to the ZigBee base-station/client be d_1 (See

Fig. 3.3). The BER can be modeled by [66]

$$BER_z = \frac{8}{15}\frac{1}{16}\sum_{k=2}^{16}(-1)^k\binom{16}{k}e^{20\times SINR\times(\frac{1}{k}-1)}, \tag{3.6}$$

where the $SINR$ is *Signal Interference Noise Ratio* and $SINR \approx P_{rx}^z/P_{rx}^w$ (ignoring noise). For typical indoor environment, the large-scale path loss α along a distance of d can be modeled as [66],

$$\alpha(d)(dB) = 40.2 + 20\log_{10} d.$$

Therefore, we have $P_{rx}^z = P_{tx}^z/10^{\alpha(d_2)/10}$, and $P_{rx}^w = \frac{B_z}{B_w}P_{tx}^w 10^{\alpha(d_1)/10}$.

Once we get the value of BER_z, we can calculate the PRR of ZigBee under WiFi interference with

$$PRR_{np} = (1 - BER_z)^{n_{col}}, \tag{3.7}$$

where n_{col} is the average number of "corrupted" bits, which can be regarded as

$$n_{col} = \frac{\lambda T_{pkt}}{\mu T_{bit}}, \tag{3.8}$$

where T_{bit} is bit duration of Zigbee.

3.5.2 WiFi Interferer Random Backoff during Preamble of Policing Signals

All the policing signals consist of preamble(s). As mentioned before, a Fake-PHY-Header (or a Fake-RTS) policing signal starts with a preamble; while a DSSS-Nulling policing signal is made of repeated preambles. In this sub-section, we study how WiFi interferer behave during the preamble of policing signals.

In this sub-section, we assume WiFi interferer always has backlogged packets during the whole period of the policing signal transmission. This makes our analysis pessimistic on the WBAN side. Before the transmission starts, WiFi interferer follows a *random backoff* procedure [68]. This procedure, performed according to a temporally slotted system, where each slot is called a *Random Backoff slot* (RB-slot) and of duration $\tau_{slot} = 20\mu$s, is described as follows.

In each RB-slot, a WiFi device carries out a CCA based *Random Backoff Counter Decrement Decision Logic* (RBCDDL), which returns "yes" or "no". When a WiFi

transmitter has a packet to transmit, it first initializes its random backoff counter n_b to $n_{b0} = 1 + cw$, where cw is an integer drawn according to uniform distribution over interval $[0, CW]$ (typically $CW = 7$) [68]. The decrement of n_b depends on the per-RB-slot RBCDDL decision: decrement by 1 on "yes", and remain unchanged on "no".

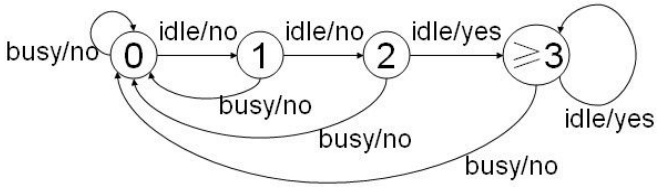


Figure 3.12: Markov Chain on RBCDDL Behavior. "0" is the initial state.

The behavior of RBCDDL follows the *discrete time Markov chain* (simplified as "Markov chain" in the following) of Fig. 3.12. The input ("idle", "busy") is the results of CCA during the corresponding RB-slot. *Let us focus on the duration when WiCop policing preamble exists on the wireless medium.* Because WiCop policing preamble is a DSSS scrambled pseudo white-noise and its duration is much longer than an RB-slot duration τ_{slot}, we can reasonably assume the probability that CCA reports "busy" in an RB-slot to be a constant P_{cca}, which can be calculated according to Appendix .1. Therefore, the probability that RBCDDL reports x times of "yes" during n_a continuous RB-slots is

$$q(x, n_a) = \binom{n_a}{x} P_{yes}^{x} (1 - P_{yes})^{n_a - x}, \tag{3.9}$$

where P_{yes} is the probability that RBCDDL reports "yes" in one RB-slot. By analyzing the Markov chain of Fig. 3.12, we have

$$P_{yes} = (1 - P_{cca})^3.$$

3.5.3 PRR with Fake-PHY-Header Policing

To determine the PRR with Fake-PHY-Header policing, we first derive P_{fph}, the probability that the WiFi interferer successfully decodes the Fake-PHY-Header policing frame.

Let t_0 be the time instance when the WiCop policing node starts transmitting a Fake-PHY-Header policing frame. Because the policing node carries out CCA before transmitting, we can assume at t_0 the WiFi interferer is not transmitting. On the other

hand, as Section 3.5.2, we still pessimistically assume the WiFi interferer is always backlogged during the whole period of the Fake-PHY-Header policing frame transmission. Hence at t_0, the WiFi interferer has a positive random backoff counter value $n_{b0} = x$, where x is uniformly distributed over $\{1, 2, \ldots, CW + 1\}$.

To successfully decode the Fake-PHY-Header policing frame, the WiFi interferer must first maintain its random backoff counter n_b above 0 in the first 6 RB-slots (which corresponds to the first 120 bits of the Fake-PHY-Header policing frame preamble [68]) after t_0. This probability is $[1 - \sum_{x=1}^{6} \frac{q(x,6)}{CW+1}]$. Then the WiFi interferer must correctly decode (just getting "yes" decisions from RBCDDL is no longer enough) the remaining 72 bits of the Fake-PHY-Header (the last 8 bits of preamble, plus 16-bit SFD, plus 48-bit PHY header), this corresponds to a probability of $(1 - P_{ber})^{72}$, where P_{ber} is the bit error rate of WiFi interferer's decoding of the policing frame (see Appendix .1 for the calculation of P_{ber}). Therefore, the probability that a WiFi interferer successfully decodes the Fake-PHY-Header policing frame is

$$P_{fph} = [1 - \sum_{x=1}^{6} \frac{q(x,6)}{CW+1}](1 - P_{ber})^{72}.$$

This implies that the PRR of ZigBee under WiFi interference with Fake-PHY-Header policing is

$$PRR_{fph} = P_{fph} + (1 - P_{fph})PRR_{np}. \tag{3.10}$$

3.5.4 PRR with Fake-RTS Policing

Similarly, to decode a Fake-RTS policing frame, the WiFi interferer needs to decode an extra 160 bit MAC header (See Section 3.2), compared to Fake-PHY-Header policing frame. Therefore, the success probability to detect and decode the Fake-RTS policing frame is $P_{fr} = P_{fph}(1 - P_{ber})^{160}$. Thus, the PRR of ZigBee under WiFi interference and Fake-RTS policing is given by

$$PRR_{fr} = P_{fr} + (1 - P_{fr})PRR_{np}. \tag{3.11}$$

3.5.5 PRR with DSSS-Nulling Policing

The effect of DSSS-Nulling on the WiFi interferer is different from the other policing strategies in two aspects. First, DSSS-Nulling is transmitted persistently along with the ZigBee transmission. Second, the DSSS-Nulling policing signal is band-pass filtered.

Let us inspect how the repeated preamble (persistently along Zigbee transmissions) delays the WiFi transmission.

First, because each WBAN polling period ends with a long WBAN idle interval for WiFi interferer to transmit, we can assume the WiFi interferer's backlog by the beginning of the next WBAN polling period is very low (depleted or nearly depleted). Under the assumption of constant WiFi inter-arrival time $1/\lambda$, WiFi packet duration $1/\mu$, and ZigBee packet duration $T_{pkt} > 1/\lambda > 1/\mu$, we can pessimistically assume during each WBAN polling period, throughout the transmission duration of the kth ($k = 1, 2, \ldots$) ZigBee packet, the WiFi interferer has at the most $N_c = \lceil \lambda k T_{pkt} \rceil$ packets to transmit.

We further pessimistically assume that to transmit each of the N_c WiFi interferer packets, the random backoff counter is always initialized to $n_{b0} = 1$, the minimum possible value (hence the most intense interference threat to ZigBee); and that each WiFi interferer packet transmission collides with $N_B = \lceil \frac{1}{\mu T_{bit}} \rceil$ bits of the ZigBee packet, where T_{bit} is the duration of a ZigBee bit.

With the above pessimistic assumptions, we obtain a lower bound of PRR of ZigBee WBAN under WiFi interference with DSSS-Nulling policing signal as

$$PRR_{dn} \geq 1 - \sum_{x=1}^{N_c} \big(q(x, N_s)(1 - (1 - BER_z)^{xN_B}) \big), \tag{3.12}$$

where $N_s = \lceil T_{pkt}/\tau_{slot} \rceil$; and BER_z is the bit error rate of ZigBee under WiFi interference (see Equation (3.6)).

Another factor about the performance of DSSS-Nulling policing is the band pass filter used to shape DSSS-Nulling policing signal. In Appendix .2, we prove that the impact from the band pass filter is minor. Please refer to Appendix .2 for more detail.

3.5.6 MTTF and MTTR of WBAN

With the above ZigBee packet reception rates PRR at hand, we can calculate the WBAN performance metric of MTTF and MTTR (see Section 3.3.2 for their definitions).

According to the description of Section 3.3.2, assuming i.i.d. ZigBee packet losses, Markov chain of Fig. 3.13 describes the state of a ZigBee client after each of its uplink packet transmission. In this Markov chain, each state is labeled by a number, which is the current number of continuous ZigBee packet transmission failures of the ZigBee client (i.e., start from current time and look back, how many ZigBee packet transmissions have continuously failed; note each transmission success resets this number to 0).

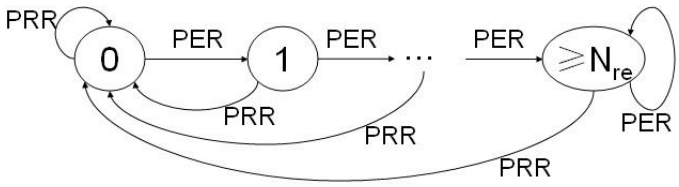


Figure 3.13: Markov chain of WBAN state: each state indicates the current number of continuous ZigBee uplink packet transmission failures; initial state is "0".

According to the description of Section 3.3.2, a WBAN failure is defined as the lost of a data chunk after its N_{re} ZigBee uplink packet (re)transmissions. Therefore, a WBAN failure happens every time the Markov chain of Fig. 3.13 enters state "$\geq N_{re}$". The Markov chain takes one input every WBAN polling period $T_{polling}$, therefore, the WBAN's *Mean Time To Failure* (MTTF) is

$$MTTF = \frac{T_{polling}}{\pi_{N_{re}}} = \frac{T_{polling}}{PER^{N_{re}}},$$

where $\pi_{N_{re}}$ is the stable probability of state "$\geq N_{re}$" in Fig. 3.13's Markov Chain.

To obtain *Mean Time To Recover* (MTTR), we define $P_f(k)$ as the probability that a WBAN failure lasts $kT_{polling}$ ($k = 1, 2, \ldots$) since it starts. This probability can be represented by:

$$P_f(k) = PRR \times PER^{k-1}.$$

With $P_f(k)$, we can calculate MTTR by:

$$MTTR = \sum_{k=1}^{\infty} P_f(k) k T_{polling} = \frac{T_{polling}}{PRR}.$$

3.6 Experiments

For interested reader, a demo for our experiment is available at YouTube [74].

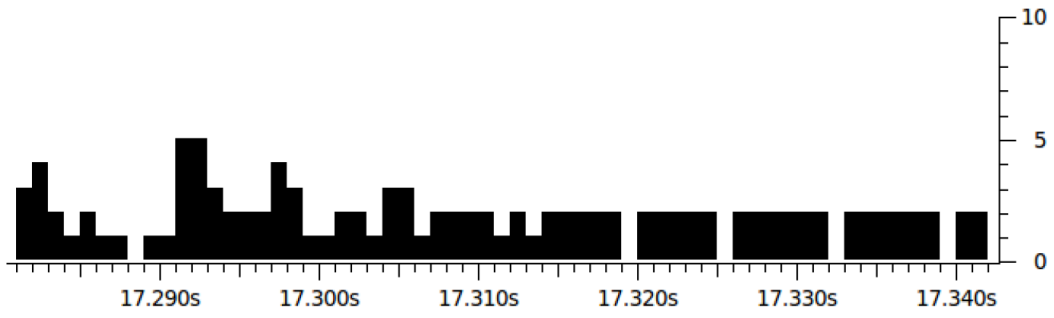
3.6.1 Effects on WiFi Temporal White-Spaces

We first illustrate the impact of WiCop on WiFi temporal white-spaces. The experiment set up reuses that of Section 3.3.1 and Fig. 3.3. *Host-I* is the WiFi interferer, which keeps sending WiFi traffic to WiFi AP at an application data rate of 10Mbps. Three feet from *Host-I* lies *Host-P*, the WiCop policing node. *Host-P* is wired/sychronized to the WBAN base station *Mote-B* (via *Host-Z*). The WBAN polling period is 10ms, and the WBAN active interval is less than 5ms. To protect such WBAN, the policing node broadcasts policing signal every 10ms, claiming a WBAN active interval of 5ms. This affects the WiFi interference traffic, which is recorded by *Host-M*, the host of WiFi AP (the WiFi interference traffic destination).

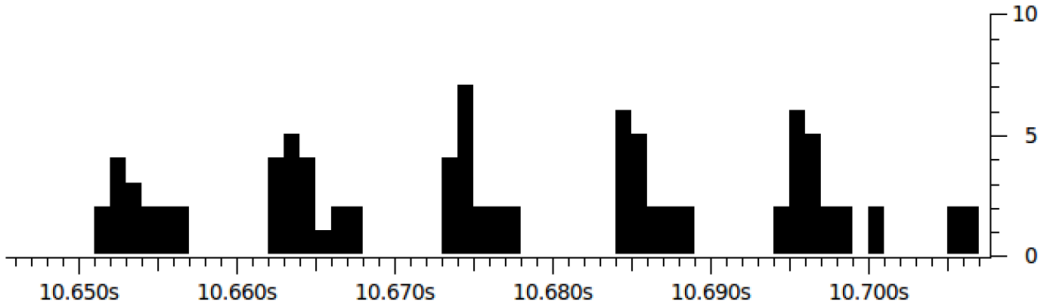
Fig. 3.14 shows two typical excerpts of the WiFi interference traffic trace, one generated under no WiCop policing, and the other generated under WiCop policing (without loss of generality, the specific policing strategy used in this example is Fake-PHY-Header).

Under no policing, there are few WiFi temporal white-spaces wide enough to allow the 5ms WBAN active intervals (see Fig. 3.14(a)). In contrast, under policing, WiFi temporal white-spaces of no less than 5ms wide emerge every 10ms, enough to allow the periodical WBAN communications.

We then illustrate the effectiveness of Fake-PHY-Header, DSSS-Nulling, and Fake-RTS policing. Fig. 3.15 compares the distributions of WiFi temporal white-space lengths under these three policing strategies. For each policing strategy, we rerun the aforementioned experiment for 25s, with a WBAN polling period of 25ms and WBAN active interval of 5ms. If policing is successful for every WBAN polling period, $25\text{s}/25\text{ms} = 1000$ WiFi temporal white-spaces of length ≥ 5ms should be created. According to Fig. 3.15: all three policing strategies result in over 600 such temporal white-spaces; with DSSS-Nulling the most effective (with the highest success rate). Note Fig. 3.15 also shows there are a large number of WiFi temporal white-spaces of length less than 2ms. This is



(a)



(b)

Figure 3.14: (a) WiFi interference traffic when there is no policing; (b) WiFi interference traffic when there is policing. The X axis is time (unit: second); the Y axis is the number of WiFi interference traffic packets received in each 1ms time slot. In case of (b), WiCop sends a Fake-PHY-Header policing packet every 10ms to claim 5ms of WBAN active interval.

because when WiFi is allowed to transmit continuously, there are short temporal white-spaces (each less than 2ms) between each consecutive WiFi packets.

It is also of interest to see how WiFi transmissions are negatively affected by WiCop. Fig. 3.16 shows the throughput of TCP and UDP connections over WiFi when there is policing. The WBAN polling period is 25ms. As the claimed length of WBAN active interval increases, the throughout decreases. However, when the claimed WBAN active interval is 5ms, the decreases of TCP/UDP throughput are both mild. This shows that our policing strategies enable the coexistence of WiFi and WBAN.

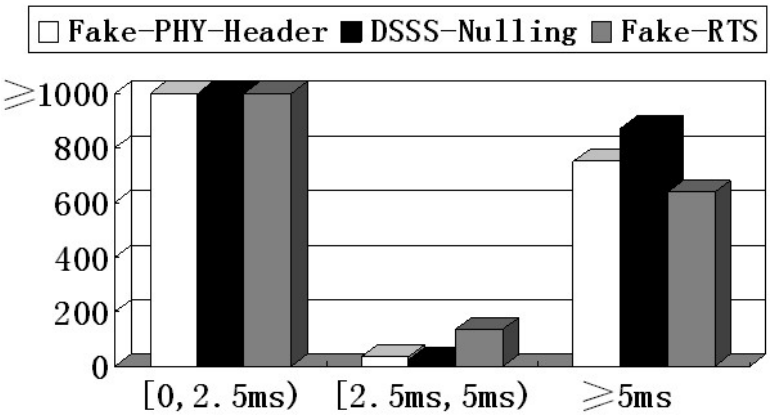


Figure 3.15: Histogram showing WiFi temporal white-space distribution under Fake-PHY-Header policing (white bar), DSSS-Nulling policing (black bar), and Fake-RTS policing (grey bar) respectively. The X axis is the range of the lengths of WiFi temporal white-spaces (granularity: 2.5ms); the Y axis is the the number of such WiFi temporal white-spaces encountered throughout the 25s experiment trial. Y axis is truncated at 1000 to save page space: temporal white-spaces in the $0 \sim 2.5$ms range are mostly those between consecutively transmitted WiFi packets. WiCop sends a policing packet every 25ms to claim 5ms of WBAN active interval.

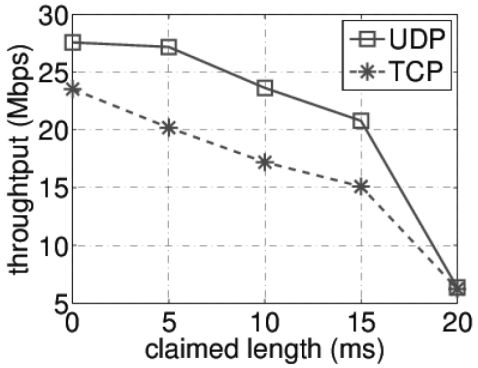


Figure 3.16: WiFi throughput degradation under WiCop policing (Without loss of generality, we use Fake-PHY-Header policing strategy in this example). X axis is the claimed length of WBAN active interval; Y axis is the throughput of WiFi interference traffic. WBAN polling period is 25ms.

3.6.2 Effects on WBAN Performance

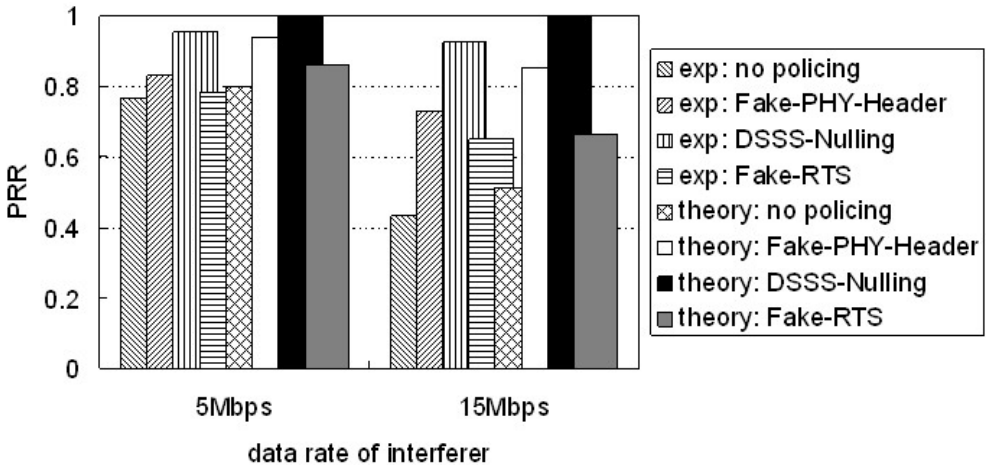


Figure 3.17: WBAN PRR under different WiFi interferer data rates. Both experimental ("exp") results and theoretical ("theory") predictions are plotted.

Now, we are in the position to evaluate the effects of WiCop on WBAN performance.

We reuse the experiment set up of Section 3.3.1 and Fig. 3.3, and deploy it in a typical indoor environment. All wireless links are *Non-Line-Of-Sight* (NLOS).

The WBAN is a centralized ZigBee WBAN, which runs a WBAN polling period of 100ms, and a WBAN active interval of 5ms. Both the WBAN base station and WBAN client transmits at 0dBm over a mutual distance of $d_2 = 4$ft [8].

The WiFi interferer (*Host-I*) runs IEEE 802.11g and transmits at power level of 30dBm. Its distances to the WBAN base station (*Mote-B*), WBAN client (*Mote-C*), and WiCop policing node (*Host-P*) are set to 6ft, 6ft, and 3ft, respectively. The (application layer) data rate of the WiFi interferer is set to 5Mbps and 15Mbps respectively. For each of the data rate, four experiment trials are carried out, respectively corresponds to no policing, Fake-PHY-Header policing, DSSS-Nulling policing, and Fake-RTS policing. Each trial lasts 600s.

[8] When evaluating wireless uplink, the downlink polling message is sent via reliable wired connection, so as to prevent errors caused by downlink wireless packet loss.

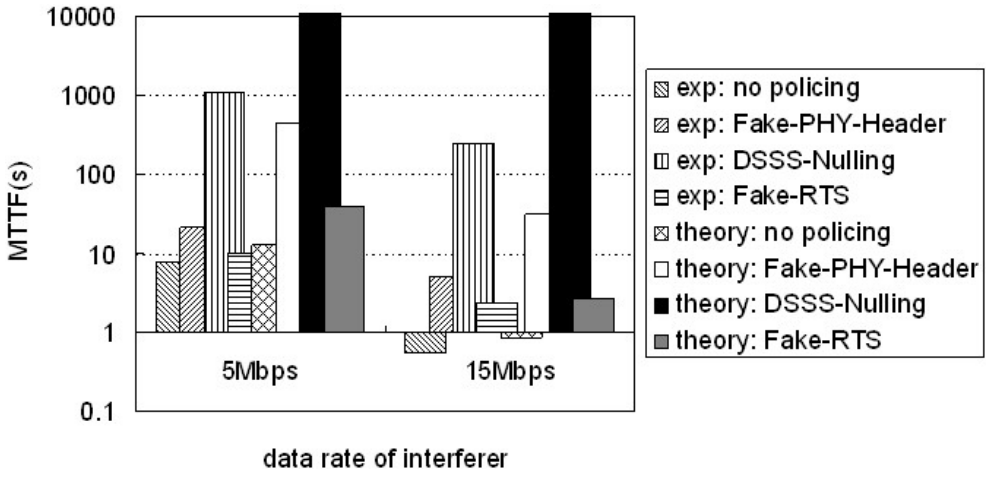


Figure 3.18: WBAN MTTF under different WiFi interferer data rates. Both experimental ("exp") results and theoretical ("theory") predictions are plotted. As theoretical values of MTTF with DSSS-Nulling policing under 5 and 15Mbps interference are 1×10^{11} and 3.7×10^{9} (seconds) respectively, we truncate Y axis at 10^{4}.

The results are summarized by Fig. 3.17, 3.18, and 3.19, respectively plotting the PRR, MTTF, and MTTR of the WBAN. Each of these figures also plots the theoretical predictions.

The setup of theoretical calculations is summarized as follows. First, the calculations use the same layout as the experiment. Second, as we use iperf to generate WiFi interference in experiment, we suppose the WiFi packet inter-arrival time and packet duration are constant in theoretical calculation. Thus, we use Equation (3.7), (3.10), (3.11), and (3.12) to calculate PRR. Third, the parameters about PHY/MAC of ZigBee or WiFi strictly follow IEEE $802.15.4$ or 802.11 standard. Last, all the other parameters in calculation use the same value of the parameters in experiment.

These figures, no matter through experimental results or theoretical predictions, lead to a number of observations. First, under heavy WiFi interference (e.g., when the WiFi interferer's data rate is 15Mbps), the WBAN PRR degrades significantly if there is no policing. Second, DSSS-Nulling policing performs better than Fake-PHY-Header and Fake-RTS policing in maintaining WBAN PRR under heavy WiFi interference. This is because DSSS-Nulling policing signal continuously repeats throughout the WBAN active interval; while Fake-PHY-Header (or Fake-RTS) policing signal is just broadcasted

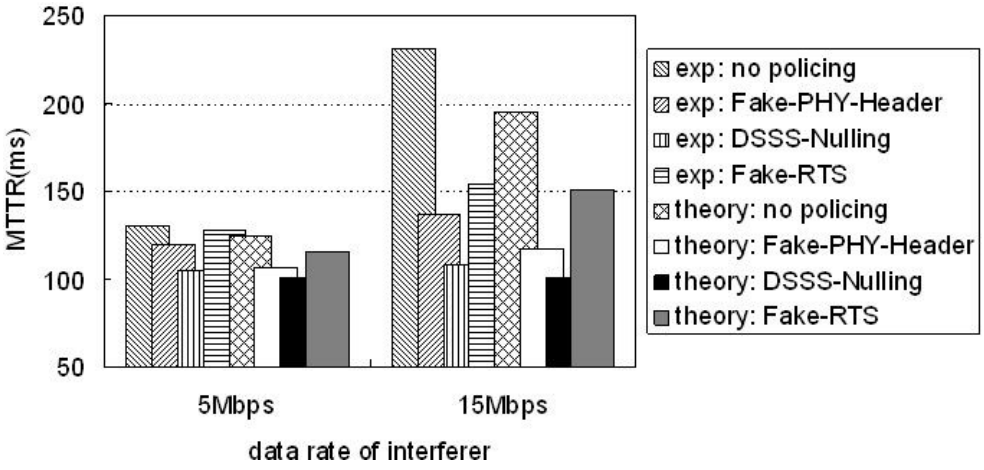


Figure 3.19: WBAN MTTR under different WiFi interferer data rates. Both experimental ("exp") results and theoretical ("theory") predictions are plotted.

once, right before each WBAN active interval. Third, WiCop can significantly improve WBAN performance under WiFi interference. For example, under heavy WiFi interference (15Mbps trials), experimentally, DSSS-Nulling policing can improve PRR by 116% (from 0.43 to 0.93), improve MTTF from 0.5s to 245.6s, and decrease MTTR from 232ms to 108ms. Fourth, the metric obtained by theoretical calculation is more optimistic than the same metric obtained by experiment (under the same data rate and with/without the same policing strategy). The reason is: there are other 'hidden' WiFi interferers around experimental environment; SORA does not have enough big power and enough good signal quality to suppress these 'hidden' WiFi interferers; 'hidden' interferers also degrade the signal quality of policing signal.

3.6.3 Case Study on ECG Signal Distortion

In this section, we utilize real-world ECG traces from the public medical database of PhysioNet [2] to evaluate the distortion of ECG signal.

The "gold standard" of measuring ECG signal distortion is the subjective metric of *Mean Opinion Score* (MOS) [3]: mean score given by medical professionals by comparing the original ECG trace and the reconstructed ECG trace.

Unfortunately, obtaining subjective metrics like MOS incur overwhelming workload. As a result, several objective metrics have been proposed in literature. Among

these objective metrics, *Wavelet based Weighted Percentage Root mean square Difference* (WWPRD) is one of the best for two reasons. First, it quantifies the significance of ECG signal components in frequency domain. Second, it can be mapped to MOS in some range. Therefore, in our experiments, we choose WWPRD as the distortion metric.

According to Al-Fahoum et al. [3], the way to calculate WWPRD is as follows.

First, we use *Cohen-Daubechies-Feauveau* (CDF) $9/7$ *Wavelet Transform* (WT) [46] [3] to obtain the sub-band coefficients of the original signal and the reconstructed signal respectively. Let the coefficients of the jth sub-band of original signal be $\{c_{j,1}, c_{j,2}, \ldots, c_{j,n_j}\}$, where $j = 0, 1, 2, 3, 4, 5$. Denote define the coefficients of the jth sub-band of reconstructed signal as $\{\widetilde{c}_{j,1}, \widetilde{c}_{j,2}, \ldots, \widetilde{c}_{j,n_j}\}$. The *Wavelet Percentage Root mean square Difference* (WPRD) of the jth sub-band is given by

$$WPRD_j = \sqrt{\frac{\sum_{i=1}^{n_j}(c_{j,i} - \widetilde{c}_{j,i})^2}{\sum_{i=1}^{n_j} c_{j,i}^2}},$$

where $c_{j,i}$ is the ith coefficient of the jth sub-band of original signal, $\widetilde{c}_{j,i}$ is the ith coefficient of the jth sub-band of reconstructed signal. Last, we calculate WWPRD by

$$WWPRD = \sum_{j=0}^{5} w_j \times WPRD_j,$$

where w_j is the weights of the jth sub-band. The weights are $6/27$, $9/27$, $7/27$, $3/27$, $1/27$, $1/27$ respectively [3].

Clearly, the smaller value of WWPRD, the less the distortion of the received signal.

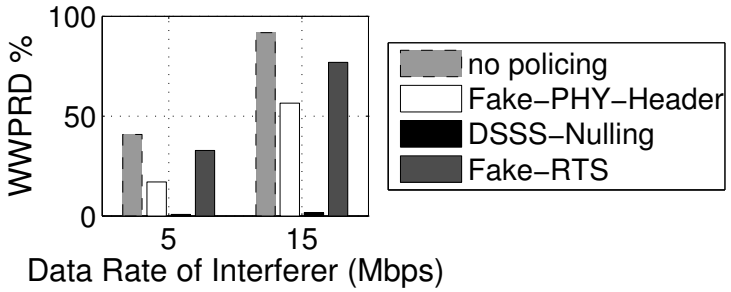


Figure 3.20: WWPRD of ECG signal under different WiFi interference source end data rates

In our evaluation, we overlay the real-world ECG traces from PhysioNet [2] onto the packet reception traces in Section 3.6.2. That is, for the experiments in Section 3.6.2, we emulate the ECG sensor (i.e., the WBAN client *Mote-C* in Fig. 3.3) readings by reading from PhysioNet ECG traces. Fig. 3.20 shows the WWPRD of the ECG traces received at the WBAN base station (i.e., *Mote-B* in Fig. 3.3). From this figure, we can make two observations: First, the WWPRD under no policing is at least 40%, this way exceeds the empirical acceptable limit of 15% [46]. Therefore, WiFi interference indeed distorts ECG signal. Second, policing strategy can reduce the distortion. For example, with DSSS-Nulling policing, the WWPRD is less than 2% even under heavy WiFi interference (when WiFi interferer data rate is 15Mbps).

3.7 Related Work

In this section, we provide a brief overview of related work pertaining to our work in the areas of

- WBAN and WiFi coexistence
- Experiment in medical units
- Denial of Service attack (DoS) to WLANs

3.7.1 Coexistence between Low power wireless schemes and WiFi

It is widely accepted that WiFi can severely interfere ZigBee communications [32] [63] [21]. Huang et al. [32] argued that the performance degradation of ZigBee in the presence of WiFi interference is caused by two main reasons, namely power asymmetry and carrier sense based CCA. The experiments in [21] showed that WiFi might interfere ZigBee transmission significantly under certain conditions even with a center frequency offset of 18MHz. Recently, many researchers found that ZigBee transmitters might impact WiFi performance under certain conditions [57] [44] [27]. Most of these works use packet loss rates to measure the performance of WBAN. However, in our work, applying ZigBee to delay sensitive applications, we use application level performance metrics, such as MTTF and MTTR.

Some researchers give analytical solution to evaluate the performance of ZigBee network under WiFi interference. Shin et. al. [63] conducted numerical analysis and simulations to evaluate the PER of ZigBee communication under the interference of WiFi. Shin argued that WiFi would not impact ZigBee communication if the separation of their center frequency is bigger than 7MHz. Zhang et. al. [84] analyzed the collision probability of WiFi and Zigbee, under two assumptions. One is that WiFi uses ED-CCA; the other is that inter-arrival time of WiFi packets is exponentially distributed. Our analytical framework gives another solution to calculate the corruption probability when collision occurs, by considering the impact from WiFi packet duration (this impact was also revealed by the experiment of Liang [44]). Further, our work is the only one comparing theoretical result and experimental result (for our best knowledge).

Our analytical framework uses the random back off model of WiFi. Similar model has been studied by Bianchi [11]. Bianchi supposes every WiFi device can always detect the transmission of other WiFi devices, unless collision happens [11]. We suppose WiFi interferer uses CS-CCA, so WiFi interferer does not always detect our poling signal (though it is WiFi compliant signal), but with a probability. We derive the probability that a WiFi interferer detects policing signal.

Some researchers propose to passively exploit the temporal or spectral white-spaces in WiFi transmissions to enable coexistence of WiFi and other wireless schemes. Huang et al. [32] designed a MAC protocol to detect and use the idle time slice (temporal white-spaces) in WiFi sessions. Liang [44] proposed a mechanism to detect and estimate the temporal white-spaces in WiFi transmission and designed an MAC protocol to utilize temporal white-spaces of different lengths. Arkoulis [5] proposed a simple and efficient method to detect a single operational frequency channel that guarantees satisfactory communications. However, in some cases, white-spaces in time and frequency domain may not exist or are insufficient. WiCop, in contrast, proactively enforces temporal white-spaces on demand to support WBAN traffic.

It is brought to our attention recently that Hou et al. [31] is in fact the first to propose the Fake-RTS policing strategy (in the form of fake CTS to be exact), though we proposed the strategy independently. Nevertheless, we are the first to implement this strategy on SDR platform; and by exploiting the flexibility of SDR, we integrate this strategy as one of the runtime alternatives into a more holistic framework. We are also the

first to compare this strategy with other strategies in the context of an SDR platform.

3.7.2 Evaluation of the Performance of Medical WBAN

Many researchers design, and deploy wireless medical systems in hospital units [61] [53] [52] [7] [36] [19] [42] [22] [45] [78]. Paksuniemi et. al. [53] reveal problem areas in patient monitoring when applying Bluetooth, ZigBee and UWB to vital sign monitoring in ICU and operating rooms. Chipara et. al. [19] use over-sampling to increase reliability of a patient monitoring system, the main applications of which include temperature and heart beat monitoring. Ko et. al. [42] design a hop-by-hop retransmission scheme to enhance the wireless medical emergency detection system. Garudadri [22] applies Compressed Sensing to ECG. This approach uses the redundancy in periodic ECG trace, to mitigate distortion under high packet losses. CodeBlue [45] and AlarmNet [78] use dynamic power to enhance mote based medical care systems. Most of these works propose general methods to increase the reliability in miscellaneous wireless links. These general enhancements are orthogonal to WiCop and can be used in conjunction with WiCop to further improve the robustness of wireless medical systems. Few of these works consider the co-channel interference from WiFi.

3.7.3 Denial of Service Attacks against WiFi

A few work has investigated mechanisms for jamming WiFi transmissions from a security point of view. Karhima [38] evaluated WiFi's tolerance to wide-band and narrow-band jamming. Park [54] and Mishra [48] studied partial-band jamming to WiFi. Gummadi et.al. [26] found that some WiFi cards were sensitive to beacon losses. Thus, jamming periodic beacon is an effective means to attack WiFi. Wullems [79] used the DSSSTEST-MODE of a WiFi device to jam WLANs. In this optional working mode, a WiFi device will transmit continuous DSSS preambles, so that the other WiFi devices in range will sense the channel as busy. Bellard [10] used commercial hardware to carry out de-authentication and virtual carrier-sense attack. They found that the later was not as effective as the former. Thuente [72] studied several intelligent jamming methods with

the requirement of low power and low detection probability, including DIFS waiting jamming, ACK corruption jamming, fake RTS jamming, etc..

All these works exploit the defect of current IEEE 802.11 standards. However, our work aims to provide co-existence between WLANs and WBANs. Thus, malicious attacking methods, such as jamming beacon and fake death packet, are not considered.

The main content of this chapter was published in our conference paper [76].

The content of this chapter was accepted for journal publication in [77].

3.8 Summary

Our analytical and empirical study confirm that for safety-critical WBAN medical applications (such as ECG) with stringent temporal requirements, co-channel WiFi interference is an eminent threat. To address this WBAN-WiFi coexistence challenge, we can exploit WiFi's CCA mechanisms to propose WiCop. By deploying Fake-PHY-Header, DSSS-Nulling and Fake-RTS policing strategies, WiCop can effectively engineer the temporal white-spaces of WiFi transmissions, reserving enough resource for WBAN communications without significantly affecting WiFi performance. We implemented and validated WiCop on SORA, a software defined radio platform. Experiments show that with the assistance of the proposed WiCop policing strategies, even under heavy WiFi interference, the packet reception rate of a ZigBee-based WBAN can increase by up to 116%. Another case study on the medical application of WBAN ECG monitoring shows WiCop can bound ECG signal distortion within 2% even under heavy WiFi interference. Besides empirical implementations and evaluations, we also propose an analysis framework. This framework explores the details of WiFi CCA mechanisms to model WBAN PRR and WiFi backoff behavior in fine-grain. Based on this fine-grained model, we derive closed-form formulae on the performance of Fake-PHY-Header, DSSS-Nulling, and Fake-RTS policing. The predictions made by these theoretical models/formulae very-well match our experiment data.

CHAPTER 4

SELF-TUNED DISTRIBUTED MONITORING OF MULTI-CHANNEL WIRELESS NETWORKS USING ANNEALED GIBBS SAMPLER

4.1 Problem Description

Besides the proactive policing approach proposed in Chapter 3, to build a robust WBAN, we also need to give profiles of the subjective network [85] [56] [60].

The premise of profiling is to monitor the subjective network. Traditional monitoring, using Simple Network Management Protocol (SNMP), is actually wired side monitoring, unfortunately having several drawbacks. First, most existing SNMP products provide very limited visibility to PHY/MAC behaviors [80]. For instance, SNMP logs do not record some special MAC packets, such as beacon, RTS, and acknowledgement. Second, SNMP products (implemented protocol stacks) usually have a SNMP polling interval (typically 5 minutes) [80]. Thereafter, a long polling interval may cause missing some event with a short duration [29] [80]. Third, wired side monitoring usually occupy some resources, such as CPU time, and a port in Ethernet switch. Thus, wired side monitoring is impractical if the monitored network and interested monitor are not in the same administrative domain.

All these drawbacks can be naturally overcome by wireless side monitoring, conducted by deploying passive wireless sniffer to collect information. First, wireless side monitoring can provide detailed information in PHY/MAC, such as signal strength, collision, and back off. It is widely agreed that wireless side monitoring is a necessary complement of SNMP and base station log [14, 15, 18, 59, 80, 81]. Second, wireless sniffers are supposed to continuously work (collect information). Third, the deployment of sniffers is flexible.

However, the wireless sniffers also have two limitations. First, the number of

sniffers is limited. Second, due to hardware limitation, typically, a sniffer is only capable of collecting information at one channel at a time [14]. With these two limitations, it is a challenging issue to assign different sniffers to different channels, so as to maximize the collected information. This Sniffer Channel Assignment (SCA) problem is challenging, due to the limitations of monitoring resources (sniffers), and thus it is always impossible to monitor all the wireless users on all the channels.

Traditionally, SCA problem may be solved by below methods (though without achieving optimal result).

- **Deterministic**: an iterative solution, where at each step each sniffer independently chooses the channel that maximize the coverage of its neighborhood.

- **Greedy**: all the sniffers choose the channel in turn. The choosing strategy is that each sniffer should not monitor the user(s) that is (are) already monitored by a preceding sniffer. Therefore, **Greedy** is a centralized algorithm. On the contrary, **Deterministic** is a distributed algorithm.

- **LP-UP**: this method gives the upper bound of SCA by solving an equivalent Linear Programming problem. Obviously, LP-UP is NOT a real algorithm, but only providing the upper bound and favoring evaluations of other algorithms.

Besides these traditional methods, Arora and Zheng [6] propose an annealed Gibbs sampler based algorithm to solve SCA problem. Arora's solution achieve nearly optimal result, but suffering from parameter tuning. The proposed algorithm depends a lot on the parameter value, but parameter tuning costs a lot of efforts. In this Chapter, we propose several methods to enhance Arora's work. Actually, our work is an extension of Arora's work.

The rest of the chapter is organized as follows. The problem formulation is presented in Section 4.2. The annealed Gibbs sampler, the proposed distributed algorithm, and several of its variants are detailed in Section 4.3. Next, we evaluate their performance in Section 4.4 through extensive simulation, review the related work in Section 4.5, and summarize this chapter in Section 4.6.

4.2 Problem Formulation

Note that in this chapter, we only the give a simple formulation. Interested reader can refer [6] for more details.

We use undirected bipartite graph $G_b(S, U, L)$ to represent the system consisting of n sniffers, m users, K channels. Now let us introduce sniffer set S, user set U, and edge set L one by one. For sniffer set S with n sniffers, each sniffer s is assigned one channel $a(s) \in \{1, 2, ..., K\}$. This channel assignment of sniffer s is also denoted with $z_{s,k} = \mathbb{I}_{\{a(s)=k\}}$, where $\mathbb{I}_{\{\cdot\}}$ is indicator function. Clearly, $z_{s,k}$ is a binary variable indicating whether sniffer s is assigned channel k. Also, we use $N(s)$ to represent the user neighbors of sniffer s. Physically, $N(s)$ is the set of users that locates in the range of sniffer s. For user set U with m users, each user u chooses a channel $c(u) \in \{1, 2, ..., K\}$ (in our simulation, each user chooses the channel with the biggest signal strength). Each user u also has a weight $p_u \in [0, 1]$ [1]. Similarly, we denote the neighbor sniffers of user u as $N(u)$, such that $N(u) = \{s | u \in N(s)\}$. For edge set L, an edge $l(s, u)$ exists, if $a(s) = c(u)$.

With definition of G_b, we still need define the "collected information". In this chapter, to represent collected information, we use Quality of Monitoring (QoM) introduced by [17] (also used in [6]). QoM is defined as the expected number of weighted users monitored by sniffers, denoted by $\sum_{u \in U} p_u y_u$, where $y_u = \mathbb{I}_{\{\exists s \in N(u), s.t., a(s)=c(u)\}}$. Note that binary variable y_u is not a decision variable.

Now the SCA problem is formulated as [6],

$$\begin{array}{lll} \max & \sum_{u \in U} p_u y_u & \\ \text{s.t.} & \sum_{k=1}^{K} z_{s,k} \leq 1 & \forall s \in S \\ & y_u \leq \sum_{s \in N(u)} z_{s,c(u)} & \forall u \in U \\ & y_u, z_{s,k} \in \{0, 1\} & \forall u, s, k. \end{array} \tag{4.1}$$

This problem (equivalent to MEC problem in [17]) is proved to be NP-hard in [17].

Theorem 1. *The* MEC *problem is NP-hard with respect to the number of sniffers, even for* $K = 2$ *[17].*

[1] for instance, we can define weight to be the transmission probability of a user.

4.3 A Distributed Algorithm based on Annealed Gibbs Sampler

4.3.1 Introduction to General Annealed Gibbs Sampler

In this subsection, we briefly introduce the general annealed Gibbs sampler. Interested reader can refer [12] for details.

We define a system $\mathcal{S}$, consisting of n nodes (denoted by $s_1, s_2, ..., s_n$). The state of these n nodes is represented with a vector $\mathbf{a} = (a_1, a_2, ..., a_n)$, where a_i $(1 \leq i \leq n)$ belongs to a finite set $\mathcal{A}$. State vector $\mathbf{a}$ determines the system energy $\mathcal{E}(\mathbf{a})$ ($\mathcal{E}(\mathbf{a}) \in \mathbb{R}$). Besides global energy, every node s_i has its local energy $\mathcal{E}_i(a_i, a_{-i})$, where a_{-i} is the states of neighbor nodes of s_i.

The objective of the above general problem is to find a state vector $\mathbf{a}$, minimizing the system energy $\mathcal{E}(\mathbf{a})$. Such problem can be solved by annealed Gibbs sampler, if the local energy function $\mathcal{E}_i(a_i, a_{-i})$ can be represented by the sum of a set of potential function $V(\mathcal{B})$ [12]:

$$\mathcal{E}_i(a_i, a_{-i}) = \sum_{\mathcal{B} \in \mathcal{C}: s_i \in \mathcal{B}} V(\mathcal{B}), \tag{4.2}$$

where $\mathcal{B}$ is a subset of system $\mathcal{S}$ and $\mathcal{C}$ is clique set. The potential function $V(\mathcal{B})$ implies that $V(\mathcal{B}) \equiv 0$ if $\mathcal{B}$ is not a clique (For a better understanding of clique and potential function, please refer [12]). Similarly, global energy is of the form [12],

$$\mathcal{E}(\mathbf{a}) = \sum_{\mathcal{B} \in \mathcal{C}} V(\mathcal{B}). \tag{4.3}$$

The annealed Gibbs sampler is an iterative procedure where temperature T decreases at each step. Each node s_i, at each step, samples the next state according to the following distribution on $\mathcal{A}$ [12].

$$\pi(a_i) = e^{-\frac{\mathcal{E}_i(a_i, a_{-i})}{T}} \Big/ \left(\sum_{a_i' \in \mathcal{A}} e^{-\frac{\mathcal{E}_i(a_i', a_{-i}')}{T}} \right), a_i \in \mathcal{A}. \tag{4.4}$$

4.3.2 The Base Algorithm

In this section, we briefly introduce the base algorithm using annealed Gibbs sampler to solve SCA problem. Please refer [6] for details.

To apply annealed Gibbs sampler to SCA problem, we need define the clique, the potential function, global energy, and the local energy.

Recall the system (graph) $G_b(S, U, L)$ in Section 4.2. The energy of such system depends on the states (channel assignment) of sniffers. The state of sniffer s_i $(1 \leq i \leq n)$ is a_i (or $a(s_i)$), where $a_i = 1, 2, ..., K$; the state of sniffer set S is $\mathbf{a}$.

In this system, we define clique $\mathcal{B}$ to be the neighborhood of one user, but any sniffer of $\mathcal{B}$ does not monitor the user. In another word, $\mathcal{B} = \{s|\exists u, \text{ s.t. } s \in N(u) \text{ and } a(s) \neq c(u)\}$. Then, we define potential function for a subset of sniffers $\mathcal{B}$ by [6]

$$V(\mathcal{B}) = \sum_{u \in \{u|N(u)=\mathcal{B}\}} p_u \times \left(1 - \mathbb{I}_{\{\exists s \in \mathcal{B}, s.t., a(s)=c(u)\}}\right). \tag{4.5}$$

In another word, the potential function can be represented by the sum of unmonitored user weights (transmission probability). Thereafter, we rewrite the objective function in (4.1) with an energy function [6]:

$$\mathcal{E}(\mathbf{a})) = \sum_{u} p_u \cdot \left(1 - \mathbb{I}_{\{\exists s \in N(u), s.t., a(s)=c(u)\}}\right) = \sum_{\mathcal{B} \in \mathcal{C}} V(\mathcal{B}).$$

In another word, we transform the maximization problem in (4.1) to the minimization of the system energy.

Also, we define local energy of sniffer s_i by [12]

$$\mathcal{E}_i(a_i, a_{-i}) = \sum_{\mathcal{B} \in \mathcal{C}: s_i \in \mathcal{B}} V(\mathcal{B}), \tag{4.6}$$

where a_{-i} is the channel assignment of the neighbor sniffers of s_i (the neighbor sniffers $s_{-i} = \{s|\exists u \text{ s.t. } s_i \in N(u) \text{ and } s \in N(u)\}$). Substituting (4.5) into (4.6) yields [6],

$$\mathcal{E}_i(a_i, a_{-i}) = \sum_{u \in N(s_i)} p_u \times \left(1 - \mathbb{I}_{\{\exists s \in N(u), s.t., a(s)=c(u)\}}\right). \tag{4.7}$$

Now, with the above definitions of clique, potential function and energy functions, we are ready to introduce the distributed algorithm using annealed Gibbs sampler (to solve SCA problem). We suppose every sniffer has an independent identical exponentially distributed timer. Upon the timer firing (of any sniffer), the sniffer runs a procedure (identical to the others), which has the following steps:

1. Compute the current temperature $T(t) = f_T(t)$ $(t > 0)$, where $f_T(t)$, the cooling schedule, is a monotonically decreasing function of t [12]. Common cooling schedules include exponential cooling or logarithmic cooling. $f_T(t)$ of exponential cooling is generated according to $T(t) = T_0\alpha^t$, where T_0 is initial temperature and $\alpha \in [0.90, 0.99]$ [41]. $f_T(t)$ of logarithmic cooling is generated according to $T(t) = D/log(1+t)$, where D is constant [30].

2. With the stored channel assignment of neighbor sniffers, use (4.7) to compute the local energy $\mathcal{E}(k, a_{-i})$ (on channel k).

3. Select the next state according to a multinomial distribution, the probability vector of which is denoted as $(\pi_1, \pi_2, ..., \pi_K)$, where K is the number of channels. One element of this vector is obtained by [12]:

$$\pi_k = \frac{e^{-\mathcal{E}_i(k,a_{-i})/T}}{\sum_{c=1}^{K} e^{-\mathcal{E}_i(c,a_{-i})/T}},$$

   where $k = 1, 2, ..., K$.

4. Broadcast the current state to its neighbors.

5. Repeat above steps, until the changes in the global energy and temperature are small enough.

Theorem 2 (sufficient condition of convergence)**.** *The base algorithm using logarithmic cooling $T(t) = N\Delta/\ln(1+t)$ will converge to global optima, where N is the number of states and Δ is the maximum energy gap.*

Proof. The proof is similar to that of Example 8.8 p.311 in [12] □

Difficulty in choosing cooling schedule The base algorithm with logarithmic cooling scheme is proven to converge to the global optima if the parameters are chosen properly. However, logarithmic cooling schedule is impractical because of its slow cooling rate [50]. In contrast, exponential cooling schedule has fast convergence, but is often stuck at local optima [50]. Furthermore, exponential cooling schedule is highly sensitive to parameter selection. To obtain fast convergence and (nearly) optimal result, one needs to manually tune the initial temperature T_0 and base α.

4.3.3 Variants of the Annealed Gibbs Sampler

To address the above limitations of the base algorithm, we need consider several variants. The design focus is to not only achieve optimality but also have fast convergence and low parameter sensitivity. In this section, we consider three variants to the base algorithm – parallel execution, distorted objective function, and thermodynamic cooling.

*Parallel execution (*PARALLEL*)*

Parallel execution of anneal Gibbs samplers has been studied in literature and shown to accelerate the convergence speed [8]. There are several ways to realize parallelism: multiple runs of the base algorithm with different seeds and the same initial condition, multiple runs of the base algorithm with different initial conditions, and partition of the configuration space, etc. In this chapter, we adopt the first approach, which is most suitable for distributed implementation.

Consider M instances of the base Gibbs sampler running on each sniffer with randomly chosen seeds. The parameters of the cooling schedule are selected such that the temperature cools M times faster. In each slot, a node exchanges with its neighbors the state of all M instances. Each instance essentially runs independently in the same fashion as the base algorithm. State updates are limited within the respective instance. However, different instances can share one inter-sniffer packet, by letting one packet carry multiple pieces of state information belonging to different instances. As the length of a typical packet header is much larger than state information, the overhead of including multiple copies of state information is not much higher. At the end of the procedure, the assignment that gives the best result among the M instances is chosen.

*Distorted Objective Function (*DISTORTION*)*

Using monotonic functions to produce a distortion of the energy function has been shown to improve the performance of simulated annealing algorithms [8]. A properly chosen distortion function is beneficial for two reasons. First, it accelerates the convergence of the algorithm by changing the slope of energy function. Second, it reduces the likelihood of trapping in local minima by accentuating the differences between local minima and the global minimum. We consider the following two commonly used distortion functions:

- the logarithmic distortion: $\mathcal{E}_d(\mathbf{s}) = \ln(\beta\mathcal{E}(\mathbf{s}) + 1)$,
- the exponential distortion: $\mathcal{E}_d(\mathbf{s}) = -\exp(-\beta\mathcal{E}(\mathbf{s})) + 1$,

where $\beta > 0$.

*Thermodynamic Cooling schedule (*THERMODYNAMIC*)*

Both exponential and logarithmic cooling schedules are problem-independent and do not take into account the specific structure of the objective function to be optimized. As a result, careful parameter tuning is needed. Unfortunately, there is little guideline as to how to tune parameters such as the initial temperature and cooling factor given specific objective functions.

To address these issues, we consider the adaptive thermodynamic cooling schedule [50] [4] [51]. In a comparative study carried out in [50] on linear, exponential, logarithmic and thermodynamic cooling schedules, thermodynamic cooling schedule is shown to converge very fast while having the least overall dissipation (a measure for the efficiency of the algorithm). Unlike fixed cooling schedules, thermodynamic cooling adapts itself to the structure of the objective function during its progression defined by,

$$\frac{dT}{dt} = \frac{-vT}{\epsilon(T)\sqrt{C(T)}},$$

where v is the constant thermodynamic speed, $C(T)$ is the heat capacity of the system and $\epsilon(T)$ is the relaxation time of the system. Both quantities are functions of the current temperature T.

Both $C(T)$ and $\epsilon(T)$ depend the problem instance. Next we introduce the procedure to obtain $C(T)$ and $\epsilon(T)$ using the lumped energy algorithm proposed by Anderson and Nulton *et al.* [4, 51].

First, we need to learn the structure of the state space (Markovian field). This is done through Gibbs sampling at infinite temperature. When the temperature is infinite, from (4.4), the transition probabilities of transferring to a different channel or staying in the current channel are equal. During the sampling process, we record the energy obtained over time. Next, all the energy values are quantized to N_{el} energy levels; and the energy trace is quantized to a lumped energy trace $\{EL_i\}$, where EL_i is one of the

lumped energy levels, $i = 1, 2, ..., N_{el}$. From the lumped energy trace, we retrieve the transition probability matrix P. P_{ij} is the transition probability from energy level EL_i to energy level EL_j. P can be used to estimate the steady state distribution of the lumped process. The quantization level N_{el} affects the accuracy and complexity of the estimation of $C(T)$ and $\epsilon(T)$.

Next, we define the associated partition function $Z(T)$ and mean energy $E(T)$. $Z(T)$ is given by:

$$Z(T) = \sum_j m_j e^{-EL_j/T}, \tag{4.8}$$

where m_j is the stationary probability of EL_j obtained through transition probability matrix P. With $Z(T)$, the mean energy $E(T)$ is given by:

$$E(T) = T^2 \frac{\partial \ln Z(T)}{\partial T}. \tag{4.9}$$

Finally, $C(T)$ is computed as:

$$C(T) = \frac{dE(T)}{dT}. \tag{4.10}$$

As its name suggests, the thermal capacity $C(T)$ characterizes the amount of heat (energy) dissipation required to change the system's temperature by one unit.

The final step of obtaining thermodynamic cooling schedule is to calculate the relaxation time $\epsilon(T)$. Firstly, we apply the Boltzmannization operation to P to get $G(T)$ defined as:

$$G_{ij}(T) = \begin{cases} P_{ij} e^{-\Delta E/T}, & \text{if} \quad \Delta E > 0, i \neq j \\ P_{ij}, & \text{if} \quad \triangle E \leq 0, i \neq j \\ 1 - \sum_{k \neq j} G_{ik}, & \text{if} \quad i = j \end{cases}, \tag{4.11}$$

where $\triangle E = EL_j - EL_i$ (the delta energy from energy level i to energy level j), and $i, j = 1, 2, ..., N_{el}$. With $G(T)$, ϵ is obtained by:

$$\epsilon(T) = -1/\ln \lambda_2, \tag{4.12}$$

where λ_2 is the second largest eigenvalue of $G(T)$.

To this end, we summarize the procedure to obtain heat capacity $C(T)$ and relax time $\epsilon(T)$ as follows:

1. Run the infinite temperature Gibbs sampling to obtain an energy trace.

(a) Sample the next state randomly.

(b) Calculate the delta local energy and update global energy.

(c) Broadcast the updated global state information and global energy to all the sniffers (maybe through a spanning tree).

(d) Record the energy obtained.

2. Quantize the energy trace and calculate the lumped energy transition probability matrix P.

3. Calculate associated partition function $Z(T)$, mean energy $E(T)$ and $C(T)$ according to (4.8), (4.9) and (4.10) respectively.

4. Calculate $G(T)$ using (4.11) and obtain $\epsilon(T)$ from (4.12).

The key part of this algorithm is the infinite temperature Gibbs sampling. The sampling procedure "sounds" the system and allows us to gain knowledge of the structure of the objective function and the state space. The more we explore the state space, the more we know about its structure. However, the size of the state space is in general astronomical. For an instance, in a network with 16 sniffers and 3 channels, the size of the state space is 3^{16}, and the size of the transition matrix is $3^{16} \times 3^{16}$. Anderson's algorithm reduces the computation complexity by lumping similar energy levels together, resulting in a much smaller transition matrix. We postulate that the cooling schedule only needs to be computed once as long as the layout of the network and mean user active probability remain the same. In Section 4.4, we will carry out sensitivity analysis when individual user active probabilities change.

4.4 Numerical Simulation Result

In this Section, to evaluate the performance of the proposed algorithms (implemented in MATLAB), we consider two kinds of layout/trace: both synthetic traces and real-world traces we collected from a network test bed.

Before delving into more complicated layouts or traces, let us first look at a toy example, which shows the drawback of DETERMINISTIC and GREEDY (we introduced them in Section 4.1).

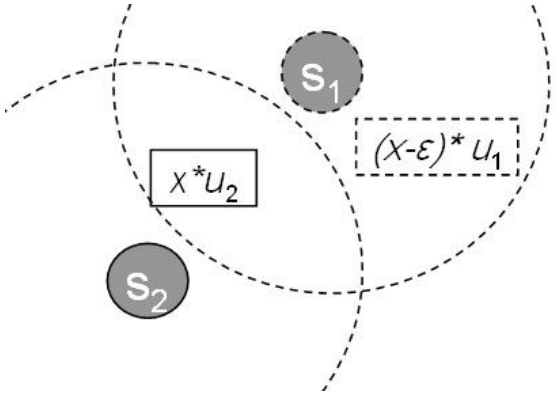


Figure 4.1: A toy example showing the layout of users and sniffers. User u_1 (with weight $x - \epsilon$) operates in channel 1; while user u_2 (with weight x) operates in channel 2. In the GREEDY or DETERMINISTIC algorithm, sniffer s_1 will choose channel 2 to maximize its output, instead of the global optimal choice (channel 1).

A Toy Example In Fig. 4.1 , there are only two sniffers (s_1 and s_2), two channels (1 and 2) and two users (u_1 and u_2). u_1, with an active probability $x - \epsilon$ ($1 \geq x > \epsilon > 0$), operates on channel 1; while u_2, with an active probability x, operates on channel 2. The optimal assignment puts s_1 and s_2 on channel 1 and 2, respectively, yielding QoM of $2x - \epsilon$. In GREEDY algorithm, s_1 will be chosen first and it will select channel 2, as this channel maximizes its coverage. This selection leaves s_2 with nothing to monitor, as u_1 is outside the range of s_2. In the DETERMINISTIC algorithm, both s_1 and s_2 choose channel 2, which leave u_1 unmonitored. Table 4.1 gives the simulation results of running different algorithms on this toy example (with $x = 1$ and $\epsilon = 0.01$).

Table 4.1: Results of Different Algorithms on the Toy Example

($x = 1$, $\epsilon = 0.01$)

	Deterministic	Greedy	LP-UP	Gibbs sampler
QOM	1	1	1.99	1.99

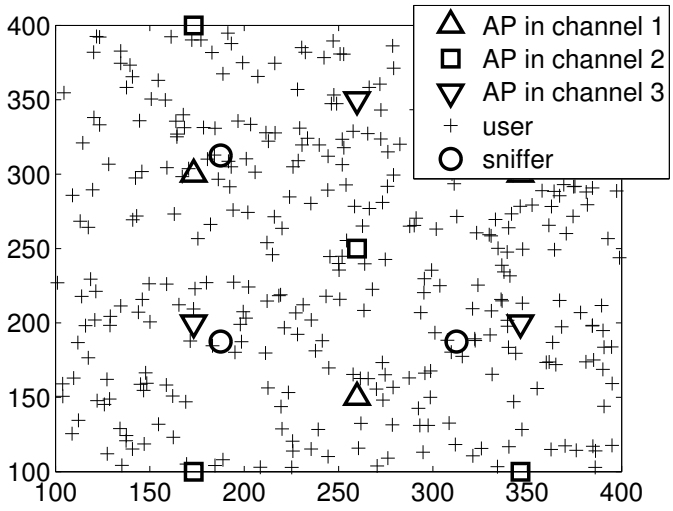


Figure 4.2: Simulation layout with APs (in three channels), users, and sniffers

4.4.1 Synthetic Traces

In this set of simulations, we use the layout depicted by Fig. 4.2. In this $500m \times 500m$ area, there are 36 hexagon cells, each with a radius of 86 meters. In these 36 cells, there are 1000 users randomly deployed. The weight of a user is defined to be the transmission probability (or active probability) of the user. The transmission probability of each user is uniformly chosen from $[0, 0.06]$. These 1000 users (and 36 APs) may work in 3, 6, or 9 channels respectively. To monitor such network, we deploy 16 sniffers separated by a distance of 156m. The coverage radius of each sniffer is 120m.

Base Gibbs Sampler

Under the layout of Fig. 4.2, we run the proposed annealed Gibbs sampler (**Gibbs**). The cooling schedule of **Gibbs** is exponential cooling with $\alpha = 0.95$ and $T_0 = 1.4$ (costing a lot of tuning efforts). We also consider 3 reference algorithms – **Deterministic**, **Greedy**, and **LP-UP** (as we introduce them in Section 4.1, **LP-UP** only gives the upper bound, which is NOT indeed an algorithm). The simulation result (the average of 30 runs) is shown by Fig. 4.3, which gives several observations. First, as the channel number increases (from 3 to 9) and sniffer number keeps, the QoM value (monitored information) decreases. Second, **Gibbs** is comparable to **LP-UP**. As the latter is NOT always feasible, the former gives the (nearly) optimal result. Third, **Gibbs** always performs better than

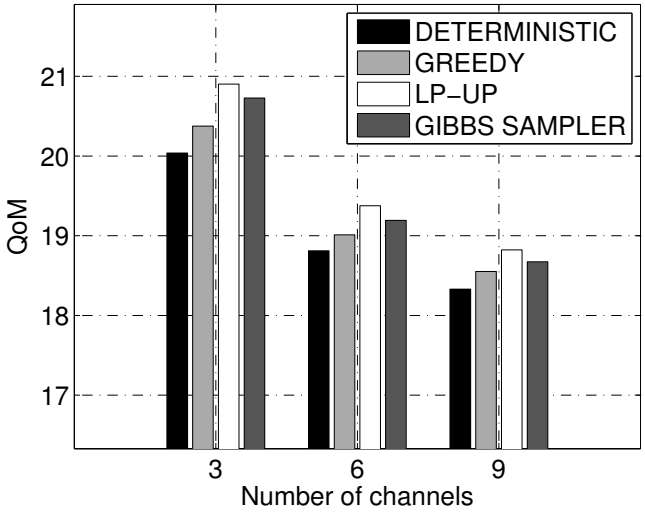


Figure 4.3: QoM with synthetic traces. QoM value 1 stands for about 33 users on average, if the the average active probability of each user is 0.03.

Greedy or **Deterministic**. Note that the real opponent of distributed **Gibbs** is distributed **Deterministic**, instead of centralized **Greedy**.

The convergence results are shown by the 3 subplots of Fig. 4.4 when the number of channels equals 3. The first subplot is for the logarithmic cooling schedule with $T_n = N\Delta/\ln(n)$, where Δ is estimated by running Greedy algorithm. The second subplot is for the exponential cooling with $\alpha = 0.9$. The third subplot is for the exponential cooling with $\alpha = 0.95$. The second and the third subplots (for cooling schedules) use the same initial temperature. With logarithmic cooling schedule, the algorithm converges very slowly. As can be seen from the first sub figure, the scope of QoM does not reduce clearly as the temperature decreases. This slow convergence makes the algorithm using logarithmic cooling schedule infeasible. With exponential cooling schedule, we have three observations. First, the algorithm converges faster than that using logarithmic cooling schedule. Second, it may stop at local optima (at least the one with $\alpha = 0.95$). Third, the performance of exponential cooling depends on the choice of the parameter(α). We can clearly see the trade-off between faster convergence and optimality in fixed cooling schedules.

We use the number of iterations to represent convergence time. The actual time consumed can be broken down into computation and communication time. On a computer

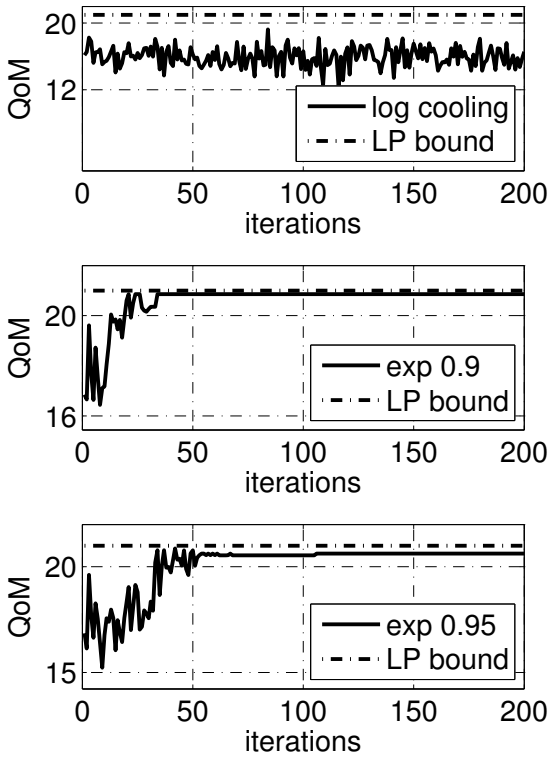


Figure 4.4: Convergence of Gibbs sampler based methods with logarithmic and exponential cooling schedules with different parameters. The layout is the same as that of Fig.4.2, and the total number of available channels is 3.

with 2GHz CPU, every 100 iterations take 5 seconds. Communication latency depends on network conditions, the underlying MAC protocol and the network size. In each iteration, each sniffer needs to broadcast its channel assignment to its neighbors, which in turn apply the proposed algorithms to make channel assignment decision.

Gibbs Sampler Variants

In the previous section, we have shown that the base algorithm using exponential cooling schedule performs better than the greedy or deterministic algorithms with carefully tuned parameters. Next, we will evaluate the performance of the base algorithm and its variants. For comparison, we introduce two metrics, *the convergence time* and *the relative error* in the achieved value of the objective function. Convergence time t^C is defined as the time (iteration numbers) when the standard deviation of energy values of the past N_w (set to 30) iterations is smaller than γ (set to 10^{-5}). The relative error σ is computed as

$$\sigma = \sqrt{\frac{1}{N_w}\sum_{m=0}^{N_w-1}(\frac{q_{t^c-m}-q^*}{q^*})^2},$$

where q^* is obtained by the LP-UP (note that it may not be achievable), and q_{t^C-m} is the value of the objective function achieved at time $t^C - m$ (m iterations priori to the convergence time). The reason for averaging over the past N iterations is because in a Gibbs sampler, due to its stochastic nature, the achievable value may still vary slightly over time.

Fig. 4.5 compares the convergence time and relative errors among 5 algorithms, each of which is the average over 30 trials with the same setup (the setup in Fig. 4.3 where the available channel number equals 3). LOG uses a single instance, no distortion, and logarithmic cooling schedule satisfying the convergence condition. EXP uses a single instance, no distortion, and exponential cooling with $\alpha = 0.95$ [41]. PAR uses two instances ($M = 2$), no distortion, and exponential cooling with $\alpha = 0.95$. DISTOR uses the logarithmic distortion with $\beta = 100$, single instance and exponential cooling with $\alpha = 0.95$. Finally, THERM uses single instance, no distortion, and the thermodynamic cooling with $v = 0.1$. From Fig. 4.5, we make the following observations:

- Logarithmic cooling schedule converges extremely slowly. This gives a poor per-

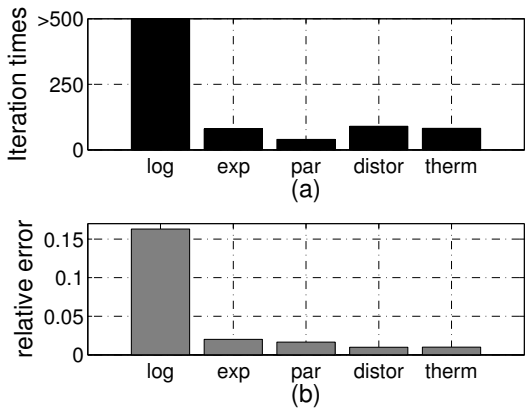


Figure 4.5: Performance of Variants of Gibbs Sampler (a) iteration times (b) relative error by the 500th iterations. Please note that given enough number of iterations ($\gg 500$) LOG can converge to global optimum.

formance within 500 iterations. However, variant algorithms achieve better performance within 100 iterations.

- PARALLEL with two instances can roughly half the convergence time with no loss of quality. In fact, the achieved utility is even better than that of the base.

- DISTORTION can improve the utility achieved with similar convergence speed. However, we find that the performance is sensitive to β. Tuning β is time consuming, requiring more than 1000 trials. Such tuning efforts make distortion method impractical.

- The use of thermodynamic cooling schedule achieves better utility without increasing iteration time.

Thermodynamic cooling schedule To gain some intuition of THERMODYNAMIC, we compare in Fig. 4.6 changes in the temperature over time in the exponential cooling schedule and thermodynamic cooling schedule. For ease of comparison, we set both schedules to start from the same initial temperature. Also included in Fig. 4.6 are the $-log(\cdot)$ of both schedules. As shown in Fig. 4.6, the thermodynamic cooling schedule behaves like the exponential cooling schedule initially and eventually flatten out. Taking the $-log(\cdot)$,

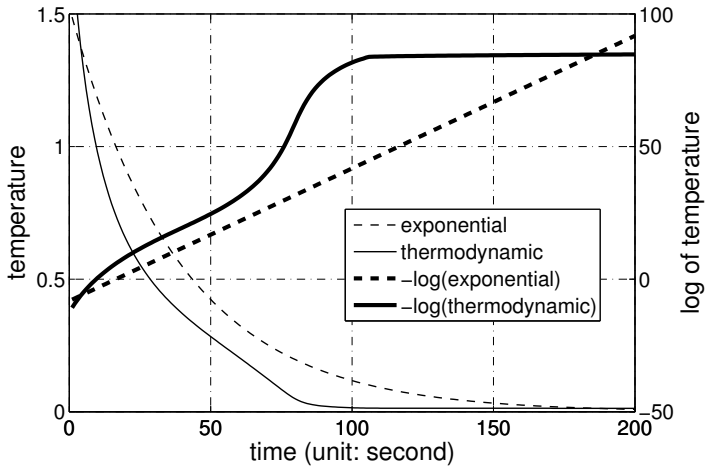


Figure 4.6: Exponential and thermodynamic cooling schedules

we observe that the "exponent" in the thermodynamic cooling schedule is small initially. It then gets bigger first and decreases later, eventually reaching zero. What this implies is that at the initial stage, more "exploration" is needed so the state space is sufficiently explored. Later, searches are done around the region that is known to contain good solutions and thus more "exploitation" is needed for fast convergence. In contrast, with the exponential cooling schedule, the exponent remains constant. Consequently, the trade-off between "exploration" and "exploitation" is uniform over time. This explains the importance of proper parameter tuning in exponential cooling schedules. If the exponent is too large, there is not sufficient exploration. Otherwise, the algorithm may explore for too long.

Sensitivity analysis From the results in Section 4.4.1, we find that annealed Gibbs samplers with thermodynamic cooling schedule performs well and incur smaller relative errors than those with exponential cooling schedule. However, computing the thermodynamic schedule itself can be time consuming, while the exponential cooling schedule depends the choice of α and T_0. In this section, we evaluate how sensitive these two methods are when the underlying structure of the problem changes moderately.

The computation complexity of determining the thermodynamic cooling schedule (or training) depends on two factors: the number of rounds to execute Gibbs sampling at infinite temperature and the number of lumped energy levels. In the simulation, the number of rounds and lumped energy levels are set to be 1000 and 100, respectively.

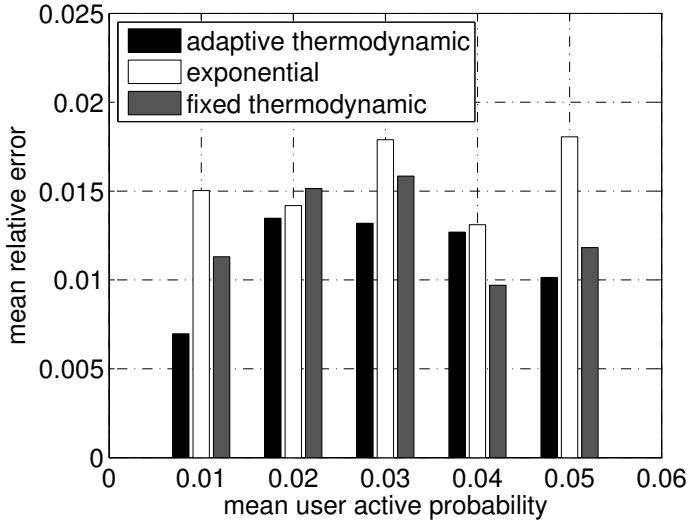


Figure 4.7: sensitivity of user active probability

Under such a setting, the time to conduct the training is roughly ten times longer than running 200 iterations of the annealed Gibbs sampler algorithm given the cooling schedule. Although the time consumed is much less than that of tuning α and T_0 in exponential cooling (running the annealed Gibbs sampler algorithm for more than 1000 times), it is still undesirable. However, the cost of training can be amortized if the cooling schedule does not change significantly when the changes in the structure of the problem are moderate. While here the "structure" of a problem is vaguely defined and "moderate" is hard to quantify, we observe a certain intrinsic nature of the optimization problem is preserved (e.g., hardness and approximability) even as the input parameters vary. For instance, in the MEC problem in (4.1), changes in the bipartite graph and the user active probability do not fundamentally change the hardness of the problem.

To test our hypothesis, we conduct experiments where the mean user active probabilities vary from 0.01 to 0.05. In each case, we run 30 trials. We compare the performance of the exponential cooling schedule (with fixed α and T_0), the thermodynamic cooling schedule with and without training. In case of exponential cooling, we carefully tune the parameters ($\alpha = 0.9$ and $T_0 = 1.4$) in the setup where the mean user probability equals 0.03. In case of the thermodynamic cooling schedule without training, we use the same cooling schedule for each mean user active probability (regardless of the variations in individual users' active probability).

Fig. 4.7 gives the relative errors of the three schemes. As shown in Fig. 4.7, performance of exponential cooling schedule with fixed parameters appears to be sensitive to the mean user active probability. Especially, when the mean user active probability equals 0.01 or 0.05, the gap between thermodynamic cooling and exponential cooling is larger than the gap when the probability equals 0.02, 0.03, or 0.04. This is because the parameters of exponential cooling are carefully tuned for the case where the mean user active probability equals 0.03. For exponential cooling, to reduce the relative error, one attempt is to tune the values of α and T_0 in the exponential cooling function; however, it is much more time consuming than calculating thermodynamic cooling schedule. Another observation is, the differences between fixed thermodynamic schedule and adaptive thermodynamic schedule are less pronounced. This supports our hypothesis that a fixed thermodynamic schedule may suffice when the structure of the problem only changes moderately. Another possible solution is to have **pre-computed** thermodynamic schedules for different time of the day, weekdays and weekends as it has been observed that WLAN traffic exhibits salient diurnal patterns.

Finally, we provide a qualitative comparison in Table 4.2 summarizing the pros and cons of different algorithms. In Table 4.2, LOGARITHMIC is the base algorithm using logarithmic cooling; EXPONENTIAL is the base algorithm using exponential cooling; both PARALLEL and DISTORTION use exponential cooling.

Table 4.2: Summary of Improved Algorithms

	Convergence rate	Optimality	Sensitivity
LOGARITHMIC	very slow	very good	fine
EXPONENTIAL	fast	parameter dependent	high
PARALLEL	very fast	algorithm dependent	high
DISTORTION	fast	good	very high
THERMODYNAMIC	fast	good	low

4.4.2 Real Traces

In this section, to evaluate the proposed algorithm, we reuse the real traces adopted by [6], which was collected in the campus network at the University of Houston [2].

As we observe from the previous section, PARALLEL and THERMODYNAMIC perform well in terms of convergence speed and optimality. Note that the two variants are orthogonal to one another. Thus, we consider one variant of the base Gibbs sampler algorithm that utilizes thermodynamic cooling schedule with two parallel instances on each node.

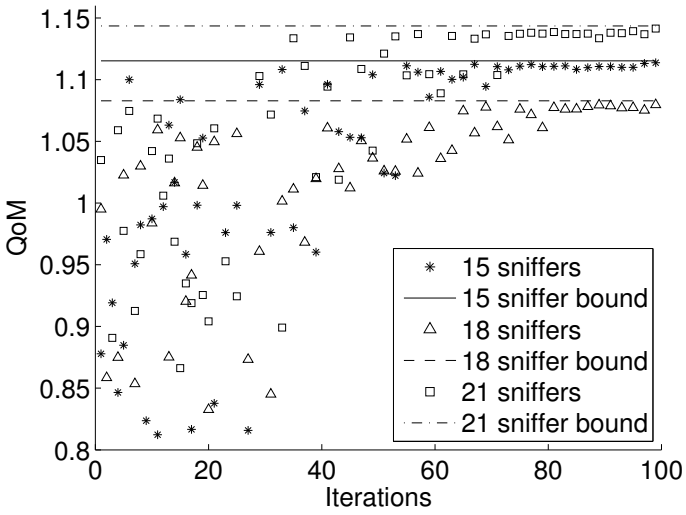
Fig. 4.8 shows the convergence of three trials using 15, 18, and 21 sniffers respectively. As expected, the combination of thermodynamic cooling schedule and parallel instances outperforms the base algorithm (using exponential cooling). The base algorithm hardly converges at round 80, while the combined one converges around 30. Additionally, with 18 and 21 sniffers, the combined one has higher utility.

4.5 Related Work

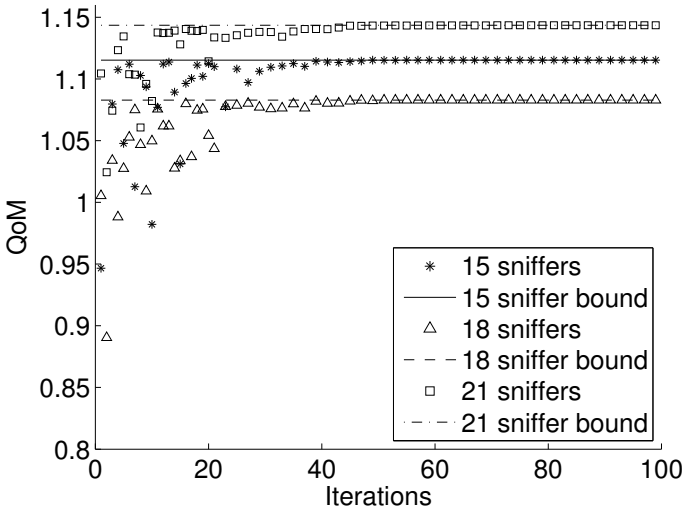
Many system works are conducted on the area of *Wireless Side Monitoring* [9, 15, 29, 59, 80, 81]. The works in [9, 29] extends the function of *Wired Side Monitoring* to analyze WiFi traffic, through *Access Point* (AP) logs or SNMP logs. Yeo is the first to deploy passive sniffer network for wireless side monitoring [80, 81]. The results of these papers are mostly experimental.

Some works focus on the allocation of sniffers and the channel selection of sniffers. Shin and Bagchi [62] study monitoring a wireless mesh network using sniffers, focusing on the selection of sniffers and their associated frequency channels. Chekuri and Kumar [13] study the allocation of sniffers, by formulating a maximum coverage problem with group budget constraints. Both of these two works study the SCA problem, by proposing centralized algorithm, without fine scaling as network size grows. In contrast, we propose a distributed algorithm using annealed Gibbs sampler for SCA problem.

[2] In this setup, we deploy sniffers in campus to capture user packets. Through inspecting and analyzing packet headers, we retrieve the knowledge of the bipartite graph and user weights.



(a) Base algorithm using exponential cooling schedule



(b) Thermodynamic cooling schedule with two parallel instances

Figure 4.8: Convergence of two annealed Gibbs sampler based algorithms for real traces

Through accelerating convergence and increasing optimality, the algorithm adopted in this chapter is an extension of Arora's work [6]. The methodology of Arora's work is motivated by Kauffmann's work [39]. All these works use Gibbs Sampler based Simulated Annealing (a.k.a. annealed Gibbs sampler) to optimize channel selection in distributed settings. Kauffmann's work is to optimize the AP channel assignment so as to minimize the interference of individual user; while our work (including Arora's work) is to maximize the user activity monitored by sniffer network.

We are also inspired by the study on Simulated Annealing. Bremaud [12] analyzed the convergence conditions of annealed Gibbs Sampler. Nourani [50] compared the linear, exponential, logarithmic and thermodynamic cooling schedules, showing that thermodynamic cooling schedule converges very fast while keeping the least dissipation. Anderson [4] gave a lumped energy algorithm to efficiently calculate the thermodynamic cooling schedule.

This chapter is an extension of [6].

The content of this chapter is under review for journal publication.

4.6 Summary

In this chapter, we study the annealed Gibbs sampler for distributed solving SCA problem, which was firstly proposed by Arora [6]. However, the base algorithm proposed by Arora does not scale well due to high parameter sensitivity. To enhance Arora's work, we study three methods – parallel, distortion, and thermodynamic cooling. The simulation result show that the combination of parallel and thermodynamic cooling is self tuned, at the same time with faster convergence rate and higher chance to reach global optima.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

ISM band WBAN are rapidly gaining popularity in e-healthcare, due to its many advantages, such as low cost, low power, and abundant supply of COTS devices/components. However, WBAN have to deal with the co-channel WiFi interferences problem. To address this problem, in this thesis, we propose a profiling-policing framework. Under this framework, we investigated several challenging issues.

- We evaluate the performance of medical WBAN under WiFi/Bluetooth interference. The result shows that WiFi is a major threat to WBAN, while Bluetooth is not.
- We propose a policing solution, WiCop, to regulate and control the WiFi interferences. Our experiment evaluations show that WiCop improves WBAN performance under heavy WiFi interferences. For instance, under heavy WiFi interferences, WiCop can increase the PRR of a Zigbee WBAN by up to 116%. A further case study on ECG monitoring shows that WiCop can bound WWPRD (a metric for evaluating ECG signal distortion) within 2%.
- We also propose a theoretical framework to analyze the performance of WBAN under WiFi interference, with/without WiCop. The theoretical analysis very well matches the experiment results.
- We propose annealed Gibbs sampler using thermodynamic cooling to solve the frequency channel selection problem in sniffer networks, so as to enhance WiFi monitoring quality. The proposed method can automatically adapt itself, and has faster convergence rate and higher chance to achieve global maxima.

5.2 Future Work

The work presented in this thesis can be extended in different directions in the future.

- In Chapter 2, we suppose the wireless channel is AWGN. In later work, we will consider more comprehensive indoor channel model for medical units, such as the channel models in [40] [83] [49] [33] [20] [86].
- Besides ECG monitoring WBAN, in the future, we will consider more medical scenarios.
- We propose a policing design and a profiling mechanism, so as to protect and optimize WBAN. In later work, we will consider integrating policing and profiling functions into WBAN base station.

Appendices

.1 Derivation of P_{cca}

To derive P_{cca}, let us briefly introduce the CCA mechanism [68].

WiFi PHY layer measures and reports CCA every RB-slot τ_{slot}. Typically, for a 802.11 1Mbps DSSS compatible WiFi receiver, $\tau_{slot} = 20\mu$s.

Conceptually, we shall regard the WiFi receiver carries out CCA and RF demodulation in parallel. The CCA works according to the automaton A_{cca} described in Fig. 1. A_{cca} has two states: "rx_idle" and "rx_busy". Whenever the RF demodulation circuit acquires a WiFi packet's preamble and successfully demodulates the subsequent SFD, a "SFD detected" event is triggered. The RF demodulation circuit then goes on to demodulate the WiFi packet. When the packet demodulation is fully completed or aborted due to check sum errors, a "WiFi packet reception ended" event is triggered. Correspondingly automaton A_{cca} is switched between the "rx_idle" and "rx_busy" states.

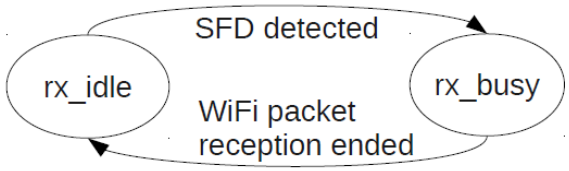


Figure 1: CCA Automaton A_{cca}. The initial state is "rx_idle".

When A_{cca} is in "rx_idle", in every RB-slot (each lasts for $\tau_{slot} = 20\mu$s), if the demodulator circuit decodes 8 consecutive bits of '1's in the first 15μs (which corresponds to 15 bit-time of demodulation), a "busy" CCA decision is made; otherwise an "idle" CCA decisoin is made.

When A_{cca} is in "rx_busy", however, in every RB-slot, a "busy" CCA decision is always made.

Therefore, when a WiCop policing node is broadcasting preamble (which consists

of continuous bits of '1's), the the probability that a WiFi interferer reports CCA "busy" in an RB-slot is

$$\begin{aligned} P_{cca} &= \Sigma_{k_1=8}^{14} 2(1-P_{ber})^{k_1} P_{ber} \\ &+ \Sigma_{k_2=8}^{13} (14-k_2) P_{ber}^2 (1-P_{ber})^{k_2} \\ &+ (1-P_{ber})^{15}, \end{aligned} \tag{1}$$

where P_{ber} is the bit error rate for the WiFi interferer's demodulation. According to [66] [68],

$$P_{ber} = Q\big((11 \times \frac{2P_{rx}^p}{N_0 B_w})^{\frac{1}{2}}\big), \tag{2}$$

where $N_0/2$(W/Hz) is the noise power spectral density [28], and P_{rx}^p is the received policing signal power. The calculation of P_{rx}^p is similar to that of P_{rx}^z (see Section 3.5.1).

.2 Impact of DSSS-Nulling Band-Pass Filtering

The reduced bandwidth of the policing signal only affects the output of the RX filter (See Fig. 3.2). Thus, we first derive the output of the RX filter upon receiving a DSSS-Nulling policing signal.

For normal WiFi signal, we define the Fourier transform of the chip signal is $kG_c(f)$, where $k = \pm 1$. The transfer function of a perfect RX filter is given by,

$$H_{opt}(f) = G_c^*(f)exp(-j2\pi fT_c),$$

where $G_c^*(f)$ is the complex conjugate of $G_c(f)$, and T_c is the chip duration [28]. The Fourier transform of the RX filter output is.

$$\begin{aligned} G_o^{normal}(f) &= H_{opt}(f)kG_c(f) \\ &= k|G_c(f)|^2 exp(-j2\pi fT_c). \end{aligned}$$

Then, the output of the RX filter at time $t = T_c$ is

$$\begin{aligned} g_o^{normal}(T_c) &= \int_{-\infty}^{\infty} G_o^{normal}(f)exp(j2\pi fT_c)df \\ &= k\int_{-\infty}^{\infty} |G_c(f)|^2 df \\ &= kE_c, \end{aligned}$$

where E_c is also known as the chip energy.

For DSSS-Nulling signal, we denote the Fourier transform of DSSS-Nulling policing chip signal as $kG_c(f)H_x(f)$, where $H_x(f)$ is the transfer function of the band-pass filter. The Fourier transform of the RX filter output is thus,

$$\begin{aligned} G_o(f) &= H_{opt}(f)kG_c(f)H_x(f) \\ &= k|G_c(f)|^2 H_x(f)exp(-j2\pi fT_c). \end{aligned}$$

Then, the output of RX filter at time $t = T_c$ is,

$$\begin{aligned} g_o(T_c) &= \int_{-\infty}^{\infty} G_o(f) exp(j2\pi f T_c) df \\ &= k \int_{-\infty}^{\infty} |G_c(f)|^2 H_x(f) df \end{aligned}$$

We suppose $H_x(f)$ is an ideal rectangular filter, such that

$$H_x(f) = \begin{cases} A & -f_x \leq f \leq f_x < f_{cut} \\ 0 & \text{otherwise} \end{cases},$$

where A is a constant, f_x is the cut off frequency of $H_x(f)$, and f_{cut} is the cut off frequency of H_{opt}. Therefore, the key observation is that the band pass filter only reduces the chip energy at the output of the RX filter by a constance factor a constant $A_x (0 < A_x < 1)$ such that $g_o(T_c) = kA_xE_c$.

To counter the negative effect of A_x, we can properly tune A, such that $A_x = 1$. In the subsequent analysis, we suppose $A_x = 1$. This implies that at the output of the RX filter, a WiFi receiver can not differentiate a DSSS-Nulling signal from a regular 802.11 frame. Therefore, we still use P_{ber} in (2) to denote the BER for WiFi interferer to decode DSSS-Nulling signal.

.3 Visibility of WBAN to WiFi

One signal (say the signal of WBAN) is visible to a WiFi device means the WiFi device can back off to this signal.

Visibility of WBAN to WiFi has been widely studied by researchers. Some researchers argue that WBAN is invisible to WiFi [32]. However, other researchers experimentally prove that WBAN transmission can make WiFi back off, when the distance between two kinds of transceivers is small enough [57] [27]. To clarify this difference, we quantitatively discuss the visibility of WBAN to WiFi (in the rest of this section, we simply use *visibility* to present *the visibility of WBAN to WiFi*).

First, the visibility depends on the CCA a WiFi device uses. If a WiFi device uses CS-CCA or CS+ED CCA, WBAN signal is invisible to WiFi. If a WiFi device uses ED-CCA, WBAN signal is visible to WiFi when the distance between two kinds of transceivers is small enough. For our best knowledge, most of WiFi device use CS-CCA or CS+ED CCA [26].

Second, the visibility generally depends on the receiver sensitivity of a WiFi device. For an instance, for WiFi compliant signal, the required received sensitivity of a $802.11g$ device is -76dBm [68]. In another word, if a WiFi signal with a strength of -76dBm (or greater) is present, a WiFi device should conclude the channel is busy. Note that the premise of reporting busy channel is decoding the WiFi signal (at least decoding the PLCP header) successfully. However, in case of missing preamble, a WiFi can not decode WiFi signal. In such case (missing preamble), if the present signal is greater than -56dBm, a WiFi device may conclude the channel is busy [68]. Thereafter, in the case of missing preamble, there must exist a threshold distance d_x (suppose the distance between

WBAN and WiFi is d), such that a WiFi device concludes the channel is busy if $d < d_x$.

Now let us calculate d_x. Recalling Equation (2.2) and the parameter values in [47], the path loss is represented by

$$\alpha(d) = 35.6901 + 10 \times 1.81199 \lg d/0.1.$$

With this equation, supposing that the transmit power of WBAN is 0dBm (typical value in WBAN [47]), we have $d_x = 1.32$m. Such small distance shows that WBAN is almost invisible to WiFi, even in the case of missing preamble.

To sum up, we conclude that WBAN is invisible to WiFi in most of the cases.

REFERENCES

[1] *National Health Care Expenditures Data*. Centers for Medicare and Medicaid Services, Office of Actuary, National Health Statistics Group, `http://www.cms.gov/nationalhealthexpenddata`, January 2010.

[2] *PhysioNet*. `http://www.physionet.org`, 2011.

[3] A.S. Al-Fahoum. Quality assessment of ecg compression techniques using a wavelet-based diagnostic measure. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):182 –191, jan. 2006.

[4] Bjarne Anderson, Karl Heinz Hoffmann, Klaus Mosegaard, Jim Nulton, Jacob Morch Pedersen, and Peter Salamon. On lumped models for thermodynamic properties of simulated annealing problems. *J. Phys.*, (49):1485–1492, Sep 1988.

[5] Stamatios Arkoulis, Dimitrios-Emmanuel Spanos, Socrates Barbounakis, Anastasios Zafeiropoulos, and Nikolas Mitrou. Cognitive radio-aided wireless sensor networks for emergency response. *Measurement Science and Technology*, 21, December 2010.

[6] Pallavi Arora, Na Xia, and Rong Zheng. A gibbs sampler approach for optimal distributed monitoring of multi-channel wireless networks. In *IEEE Global Communications Conferences. Exhibitions and Forum (GLOBECOM)*, 2011.

[7] H.H. Asada, P. Shaltis, A. Reisner, Sokwoo Rhee, and R.C. Hutchinson. Mobile monitoring with wearable photoplethysmographic biosensors. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3):28 – 40, may-june 2003.

[8] R. Azencott. *Simulated annealing: parallelization techniques*. Wiley, 1992.

[9] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. Characterizing user behavior and network performance in a public wireless LAN. *SIGMETRICS Perform. Eval. Rev.*, 30(1):195–205, 2002.

[10] John Bellardo and Stefan Savage. 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In *Proceedings of the 12th conference on USENIX Security Symposium*, volume 12, 1994.

[11] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535 –547, march 2000.

[12] Pierre Bremaud. *Markov Chains, Gibbs Field, Monte Carlo Simulation and Queues*. Springer-Verlag, 1999.

[13] Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *APPROX*, pages 72–83, 2004.

[14] Xian Chen, Yoo-Ah Kim, Bing Wang, Yuan Song, Hieu Dinh, and Guanling Chen. Sniffer channel selection for monitoring wireless lans. *Computer Communications*, 35(16):1994 – 2003, 2012.

[15] Yu-Chung Cheng, John Bellardo, Péter Benkö, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *SIGCOMM*, 2006.

[16] Nicolas Chevrollier and Nada Golmie. On the use of wireless network technologies in healthcare environments. In *Proc. IEEE 5th Workshop on Applications and Services in Wireless Networks (ASWN 2005)*, pages 147–152, June 2005.

[17] Arun Chhetri, Huy Nguyen, Gabriel Scalosub, and Rong Zheng. On quality of monitoring for multi-channel wireless infrastructure networks. In *The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.

[18] Arun Chhetri and Rong Zheng. WiserAnalyzer: A passive monitoring framework for wlans. In *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2009.

[19] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 155–168, New York, NY, USA, 2010. ACM.

[20] R. de Francisco. Indoor channel measurements and models at 2.4 ghz in a hospital. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1 –6, dec. 2010.

[21] R. de Francisco, Li Huang, and G. Dolmans. Coexistence of wban and wlan in medical environments. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1 –5, sept. 2009.

[22] H. Garudadri, P.K. Baheti, S. Majumdar, C. Lauer, F. Masse and, J. van de Molengraft, and J. Penders. Artifacts mitigation in ambulatory ecg telemetry. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pages 338 –344, july 2010.

[23] N. Golmie et al. Interference evaluation of bluetooth and ieee 802.11b systems. *Wireless Networks*, 9(3):201–211, 2001.

[24] N. Golmie et al. Performance analysis of low rate wireless technologies for medical applications. *Computer Communications*, 28(10):1266–1275, 2009.

[25] N. Golmie, R.E. Van Dyck, A. Soltanian, A. Tonnerre, and O. Rbala. Interference evaluation of bluetooth and ieee 802.11b systems. *Wireless Networks*, 9:201–211, 2003. 10.1023/A:1022821110023.

[26] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 385–396, New York, NY, USA, 2007. ACM.

[27] Jan-Hinrich Hauer, Vlado Handziski, and Adam Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *Lecture Notes in Computer Science*, volume 5432, pages 17–32, 2009.

[28] Simon Haykin. *Communications Systems*. Wiley, third edition, 1994.

[29] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 187–201, New York, NY, USA, 2004. ACM.

[30] K H Hoffmann and P Salamon. The optimal simulated annealing schedule for a simple model. *J. Phys. A: Math. Gen.*, 23(15), Aug 1990.

[31] James Hou, Benjamin Chang, Dae-Ki Cho, and Mario Gerla. Minimizing 802.11 interference on zigbee medical sensors. In *Proceedings of the Fourth International Conference on Body Area Networks*, BodyNets '09, pages 5:1–5:8, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[32] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 305 –314, oct. 2010.

[33] L. Huang, R. de Francisco, and G. Dolmans. Channel measurement and modeling in medical environments. In *Proc. International Symposium on Medical Information and Communication Technology*, February 2001.

[34] ISO/IEEE. *IEEE Standard 1073*. 1998.

[35] Mark JB. *Atlas of Cardiovascular Monitoring*. Number 130. Churchill Livingstone, New York, August 1998.

[36] E. Jovanov, A. O'Donnell Lords, D. Raskovic, P.G. Cox, R. Adhami, and F. Andrasik. Stress monitoring using a distributed wireless intelligent sensor system. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3):49 – 55, may-june 2003.

[37] Toshiko Kaneda. *China's Concern over Population Aging and Health*. Population Reference Bureau, `http://www.prb.org/Articles/2006/ChinasConcernOverPopulationAgingandHealth.aspx`, 2006.

[38] T. Karhima, A. Silvennoinen, M. Hall, and S.-G. Haggman. Ieee 802.11b/g wlan tolerance to jamming. In *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, volume 3, pages 1364 – 1370 Vol. 3, oct.-3 nov. 2004.

[39] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-based self organization of interfering 802.11 wireless access networks. In *IEEE INFOCOM'07*, pages 1451–1459, 2007.

[40] Seong-Cheol Kim, Henry L. Bertoni, and Miklos Stern. Pulse propagation characteristics at 2.4GHz inside buildings. *IEEE Transactions on Vehicular Technology*, 45(3), August 1996.

[41] S. Kirkpatrick, C. Gelatt Jr., , and M. Vecchi. Optimization by simulated annealing. *Science, 220(4598):498-516*, May 1983.

[42] JeongGil Ko, Răzvan Musăloiu-Elefteri, Jong Hyun Lim, Yin Chen, Andreas Terzis, Tia Gao, Walt Destler, and Leo Selavo. Medisn: medical emergency detection in sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 361–362, New York, NY, USA, 2008. ACM.

[43] Changle Li, Huan-Bang Li, and Ryuji Kohno. performance evaluation of ieee 802.15.4 for wireless body area network (wban). In *ICC Workshops*, 2009.

[44] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 309–322, New York, NY, USA, 2010. ACM.

[45] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *Proc. International Workshop on Wearable and Implantable Body Sensor Networks*, April 2004.

[46] M. Sabarimalai Manikandan and S. Dandapat. Wavelet energy based diagnostic distortion measure for ecg. *Biomedical Signal Processing and Control*, 2(2):80 – 96, 2007.

[47] D. Miniutti. *Narrowband on body to off body channel characterization for ban IEEE P802.15-08-0559-00-0006*. IEEE 802.15 WG, August 2008.

[48] Arunesh Mishra, Vivek Shrivastava, Suman Banerjee, and William Arbaugh. Partially overlapped channels not considered harmful. *SIGMETRICS Perform. Eval. Rev.*, 34(1):63–74, June 2006.

[49] Aleksandar Neskovic, Natasa Neskovic, and George Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *Communications Surveys Tutorials, IEEE*, 3(3):2 –12, quarter 2000.

[50] Y. Nourani and B. Andresen. A comparison of simulated annealing cooling strategies. *JOURNAL OF PHYSICS -LONDON- A MATHEMATICAL AND GENERAL*, 31(41):8373–8386, 1998.

[51] Peter Nulton, James D.; Salamon. Statistical mechanics of combinatorial optimization. *Physical Review A - General Physics, 3rd Series*, 37:1351–1356, Feb 1988.

[52] N. Oliver and F. Flores-Mangas. Healthgear: a real-time wearable system for monitoring and analyzing physiological signals. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4 pp. –64, april 2006.

[53] M. Paksuniemi, H. Sorvoja, E. Alasaarela, and R. Myllyla. Wireless sensor and data transmission needs and technologies for patient monitoring in the operating room and intensive care unit. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5182 –5185, jan. 2005.

[54] Jeongho Park, Dongkyu kim, Changeon Kang, and Daesik Hong. Effect of partial band jamming on ofdm-based wlan in 802.11g. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 4, pages IV – 560–3 vol.4, april 2003.

[55] M. Patel and Jianfeng Wang. Applications, challenges, and prospective in emerging body area networking technologies. *Wireless Communications, IEEE*, 17(1):80 –88, february 2010.

[56] N. Patwari, III Hero, A.O., M. Perkins, N.S. Correal, and R.J. O'Dea. Relative location estimation in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 51(8):2137 – 2148, aug. 2003.

[57] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai. Harmful coexistence between 802.15.4 and 802.11: A measurement-based study. In *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, pages 1 –6, may 2008.

[58] John G. Proakis and Masoud Salehi. *Communication Systems Engineering*. Prentice Hall, second edition, 2002.

[59] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, and John Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, pages 5–10, 2005.

[60] C. Savarese, J.M. Rabaey, and J. Beutel. Location in distributed ad-hoc wireless sensor networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 4, pages 2037 –2040 vol.4, 2001.

[61] P.A. Shaltis, A. Reisner, and H.H. Asada. Wearable, cuff-less ppg-based blood pressure monitor with novel height sensor. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 908 –911, 30 2006-sept. 3 2006.

[62] Dong-Hoon Shin and Saurabh Bagchi. Optimal monitoring in multi-channel multi-radio wireless mesh networks. In *MobiHoc*, pages 229–238, 2009.

[63] Soo Shin, Sunghyun Choi, Hong Park, and Wook Kwon. Lecture notes in computer science:packet error rate analysis of ieee 802.15.4 under ieee 802.11b interference. *Wired/Wireless Internet Communications*, 3510:618–618, 2005.

[64] Soo Young Shin et al. Packet error rate analysis of zigbee under wlan and bluetooth interferences. *IEEE Wireless Comm.*, 6(8), August 2007.

[65] IEEE Computer Society. *IEEE Standard 802.11*. 1997.

[66] IEEE Computer Society. *IEEE Standard 802.15.4*. 2003.

[67] IEEE Computer Society. *IEEE Standard 802.15.1*. 2005.

[68] IEEE Computer Society. *IEEE Standard 802.11*. 2007.

[69] IEEE Computer Society. *IEEE Standard 802.15.6*. 2012.

[70] Kun Tan, He Liu, Jiansong Zhang, Yongguang Zhang, Ji Fang, and Geoffrey M. Voelker. Sora: high-performance software radio using general-purpose multi-core processors. *Communications of the ACM*, 54(5), January 2011.

[71] Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Prentice Hall PTR, 5th edition, 2010.

[72] David J. Thuente and Mithun Acharya. Intelligent jamming in wireless networks with applications to 802.11 b and other networks. In *Proc. of IEEE MILCOM*, 2006.

[73] Andrew J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Prentice Hall, April 1995.

[74] Yufei Wang. *WiCop Demo*. `http://www.youtube.com/watch?v=xVy5FtTNzw8`.

[75] Yufei Wang and Qixin Wang. Evaluating the ieee 802.15.6 2.4ghz wban proposal on medical multi-parameter monitoring under wifi/bluetooth interference. *International Journal of E-Health and Medical Communications (IJEHMC)*, 2(3):48–62, 2011.

[76] Yufei Wang, Qixin Wang, Zheng Zeng, Guanbo Zheng, and Rong Zheng. Wicop: Engineering wifi temporal white-spaces for safe operations of wireless body area networks in medical applications. In *IEEE 32nd Real-Time Systems Symposium (RTSS)*, pages 170 –179, nov. 29 - dec. 2 2011.

[77] Yufei Wang, Qixin Wang, Guanbo Zheng, Zheng Zeng, Rong Zheng, and Qian Zhang. Wicop: Engineering wifi temporal white-spaces for safe operations of wireless personal area networks in medical applications. *(accepted for publication) in IEEE Transactions on Mobile Computing (TMC)*, 2013.

[78] A. Wood, J. Stankovic, G. Virone, L. Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 22(4):26 –33, july-aug. 2008.

[79] C. Wullems, K. Tham, J. Smith, and M. Looi. A trivial denial of service attack on ieee 802.11 direct sequence spread spectrum wireless lans. In *Wireless Telecommunications Symposium, 2004*, pages 129 – 136, may 2004.

[80] Jihwang Yeo, Moustafa Youssef, and Ashok Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 70–79, 2004.

[81] Jihwang Yeo, Moustafa Youssef, Tristan Henderson, and Ashok Agrawala. An accurate technique for measuring the wireless side of wireless networks. In *the 2005 workshop on Wireless traffic measurements and modeling*, pages 13–18, 2005.

[82] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[83] H.-J. Zepernick and T.A. Wysocki. Multipath channel parameters for the indoor radio at 2.4 ghz ism band. In *Vehicular Technology Conference, 1999 IEEE 49th*, volume 1, pages 190 –193 vol.1, jul 1999.

[84] Xinyu Zhang and Kang G. Shin. Enabling coexistence of heterogeneous wireless systems: case for zigbee and wifi. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 6:1–6:11, New York, NY, USA, 2011. ACM.

[85] Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 275–283, New York, NY, USA, 2000. ACM.

[86] M. Zorzi, R.R. Rao, and L.B. Milstein. Error statistics in data transmission over fading channels. *Communications, IEEE Transactions on*, 46(11):1468 –1477, nov 1998.