

**TACKLING THE CHALLENGES OF WIRELESS
INTERFERENCE AND COEXISTENCE**

By

Jun Huang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science

2012

ABSTRACT

TACKLING THE CHALLENGES OF WIRELESS INTERFERENCE AND COEXISTENCE

By

Jun Huang

Recent years have witnessed the phenomenal penetration rate of wireless networks in our daily lives, ranging from 802.11-based wireless LANs that provide ubiquitous Internet access, to 802.15.4-based wireless sensor networks that carry out various mission-critical tasks such as security surveillance and patient monitoring. However, despite the advances in the field of wireless networking, how to design high-performance wireless networks remains an open problem because of the fundamental challenges of wireless interference and coexistence.

Interference is the fundamental factor that limits the link concurrency of wireless networks. Due to the broadcast nature of wireless channel, concurrent transmissions on the same frequency interfere with each other over the air, resulting in lower throughput and higher delivery delay. Handling interference in wireless networks is difficult because of the hidden terminal problem and the exposed terminal problem. Although the former is well studied in existing literature, the later is not, especially in networks where multiple bit rates are available.

Another challenge is that interference significantly hinders the coexistence of different wireless technologies. In particular, recent studies show that wireless coexistence is becoming a growing issue due to the unprecedented proliferation of wireless devices in the unlicensed 2.4GHz band. When devices of heterogeneous physical layer operating on the same frequency, interference is more difficult to resolve as devices cannot decode the signals of each other. Moreover, co-existing devices commonly transmit at different powers, which leads to unfair

channel usage. The issue is particularly critical to lower power wireless devices.

This thesis tackles the fundamental challenges of wireless interference and coexistence to the link layer design of wireless networks. In particular, we identify two problems in the design of existing link layer protocols, and advance the state-of-the-art by offering practical solutions: (1) *the rate-adaptive exposed terminal problem* where link concurrency cannot be fully exploited by conventional link layers because of they are oblivious to the bit rate diversity; and (2) *the blind terminal problem* where existing link layer protocols fail to work in co-existing environments due to the heterogeneous physical layer and power asymmetry of co-existing devices. We motivate this research by showing that existing link layer protocols are surprisingly ineffective in handling these problems. Our experiments pinpoint the fundamental reasons of such ineffectiveness and reveal their implications on the design of link layer protocols. We then develop practical solutions to tackle the identified problems. Extensive testbed-based experiments validate the design of proposed solutions, and demonstrate their significant performance gains over existing link layer protocols.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Wireless Interference	2
1.2 Wireless Coexistence	3
1.3 Contributions	4
1.3.1 Measurement-based Modeling of Interference and Coexistence	4
1.3.2 Harnessing Rate-Adaptive Exposed Terminals	5
1.3.3 Tackling the Blind Terminal Problem	6
1.4 Organization of Thesis	6
Chapter 2 Background and Motivation	8
2.1 The Wireless Channel	8
2.2 Link Layer Protocols	11
2.2.1 Bit Rate Adaptation	11
2.2.1.1 Background on Bit Rate	11
2.2.1.2 Rate Adaptation protocols	15
2.2.2 Medium Access Control	16
2.2.2.1 CSMA-based protocols	17
2.2.2.2 TDMA-based protocols	18
2.2.2.3 Multi-channel protocols	18
2.3 Motivation	19
Chapter 3 Measurement-based Modeling of Interference and Coexistence	21
3.1 Related Work	23
3.2 The Rate-adaptive Exposed Terminal Problem	24
3.2.1 Experimental Methodology	25
3.2.2 Large-scale Quantification of RETs	26
3.2.3 Micro-benchmarking RETs	28
3.3 The Blind Terminal Problem	30
3.3.1 An Empirical Study of ZigBee and WiFi Coexistence	32
3.3.2 Analysis of Results	34
3.3.3 Modeling White Space in Real-life WiFi Networks	35
3.3.3.1 WiFi Frame Clustering	36
3.3.3.2 Self-Similarity of WiFi Frame Clusters	36

3.3.3.3	Pareto Model of WiFi White Space	39
3.3.4	Modeling ZigBee Link Performance	41
3.4	Summary of Chapter	46
Chapter 4	TRACK: Unleashing Link Concurrency in Enterprise WLANs	48
4.1	Chapter Introduction	48
4.2	Related Work	49
4.3	TRACK Design	50
4.3.1	Overview	51
4.3.2	Link Admission Control	52
4.3.3	Packet Batching	56
4.3.4	Interference Estimation	57
4.3.5	Coexistence with Non-Scheduled Traffic	59
4.4	TRACK Implementation	63
4.5	EVALUATION	64
4.5.1	Experiment Setting	65
4.5.2	Selective CCA	67
4.5.3	Performance of Rate Selection	67
4.5.4	Performance on Two-AP topologies	69
4.5.5	Throughput, Fairness and Overhead	71
4.5.6	Impact of Non-scheduled Traffic	74
4.5.7	Impact of Topology	75
4.6	Summary of Chapter	76
Chapter 5	WISE: Exploiting WiFi White Space for ZigBee Performance Assurance	77
5.1	Chapter Introduction	77
5.2	Related Work	78
5.3	WISE: White Space-Aware Frame Adaptation	80
5.3.1	Overview of WISE	80
5.3.2	Optimizing Sub-Frame Size	81
5.3.3	Frame Session Management	82
5.3.4	Sub-frame Retransmission	84
5.4	Implementation	85
5.4.1	White space sampling	86
5.4.2	White space modeling	86
5.4.3	Sub-frame Adaptation	87
5.4.4	Discussion	87
5.5	Evaluation	88
5.5.1	Accuracy of the Performance Model	90
5.5.2	Impact of Sampling Frequency	91
5.5.3	Performance Comparison	92
5.5.4	Impact of Real-life WiFi Traffic	94
5.5.5	Impact of Sub-Frame Retransmission	97

5.6	Summary of Chapter	98
Chapter 6	Conclusion and Future Work	99
6.0.1	Contributions	99
6.0.2	Open Problems	101
Bibliography	104

LIST OF TABLES

2.1	Bit rates defined in 802.11g.	15
3.1	Throughput measured on the topology of benchmark 1.	28
3.2	Throughput measured on the topology of benchmark 2.	29
3.3	Throughput measured on the topology of benchmark 3.	29
3.4	The role of WiFi interferer.	34
3.5	Notation used in white space modeling.	43
4.1	Per-client traffic load vs throughput fairness.	73
4.2	Impact of non-scheduled traffic on throughput ratio.	74
4.3	The UDP throughput improvement ratio of HET nad TRACK measured on three topologies. Each topology is characterized by the percentages of ET and HT among all link pairs.	76
4.4	The UDP throughput improvement ratio of TRACK measured on three topologies.	76

LIST OF FIGURES

2.1	Channel fading varies with time and frequency. The results are measured on an 802.11n link that consists of two laptops equipped with Intel WiFi 5300 adapters. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.	9
2.2	Constellation diagrams of BPSK, QPSK, QAM16, and QAM64.	13
2.3	The effects of noise on QPSK and BPSK.	14
3.1	The PRRs of two links running CT and CTRO. The results are measured using 104 link pairs.	27
3.2	Benchmark topologies. Data and interference links are marked with solid and dash lines. Average signal strength is labeled on links.	28
3.3	Channel utilization trace of a WiFi network consisting of 2 APs and 18 users.	31
3.4	WiFi channel state trace.	31
3.5	The blind terminal problem.	33
3.6	Self-similarity of 802.11 frame cluster arrival process.	38
3.7	Statistical test of self-similar.	39
3.8	Goodness-of-fit tests of Pareto model for real-life WiFi traces.	42
4.1	The architecture of TRACK.	52
4.2	Downlink admission control of TRACK.	55
4.3	The subcarrier with least variation in our testbed.	58
4.4	The subcarrier of most variation in our testbed.	58

4.5	Prediction error of different measurement period.	59
4.6	Update overhead of reporting CSI to the controller.	59
4.7	A case study of coexistence of scheduled concurrent downlinks (A and B) with unscheduled uplink transmission. In this case, downlink A and the uplink are out of carrier sense range of each other, while B and the uplink can hear each other. The objective of selective CCA is to disable carrier sense among scheduled downlinks, while preserving the channel contention between downlinks and the uplink. In this case, the transmitter of B (AP 2) is required to detect the signal of unscheduled transmitter (client 3) before accessing the channel, under the interference of scheduled on-going transmissions (from AP 1 to client 1). In our deployment, the measured signal strength from AP 1 to AP 2 is 15 dB stronger than the link from client 3 to AP 2.	61
4.8	Trace of selective CCA at AP 2 of the link deployment in Fig. 4.7. Selective CCA samples the received signal strength at 250KHz. The size of CCA window is set to 5. A z-score threshold of 90% confidence is used. Uplink starts transmission around 600 samples. AP 2 senses a free channel with high probability when AP 1 is transmitting. After the uplink becomes active, AP 2 detects busy channel reliably.	62
4.9	The SINR-PRR model profiled using 12 Mbps rate.	64
4.10	Noise floor measured at clients and their associated APs.	64
4.11	Testbed link quality vs production WLAN.	66
4.12	CCA sensitivity.	67
4.13	CCA Delay.	67
4.14	CDF of PRR for TRACK and packet SINR based rate selection.	68
4.15	CDF of throughput reduction caused by sub-optimal rate selection.	68
4.16	Aggregate throughput on 2-AP topology.	70
4.17	Throughput improvement vs interference.	70

4.18	Throughput analysis of 11 clients. The aggregate system throughput is shown after the name of each protocol.	71
4.19	Per-link UDP throughput	73
4.20	Per-link TCP throughput.	73
4.21	Overhead.	74
5.1	Evaluation of model.	91
5.2	Sampling frequency.	91
5.3	Broadcast frame delivery ratio.	93
5.4	Unicast Frame delivery ratio.	93
5.5	ZigBee throughput vs WiFi throughput.	94
5.6	Trace of the performance of WISE and B-MAC under the interference of real-life WiFi traffic.	95
5.7	Performance under the interference of real-life WiFi traffic.	97
5.8	Impact of sub-frame retransmission on WISE performance.	98

Chapter 1

Introduction

Recent years have witnessed the phenomenal penetration rate of wireless networks in our daily lives, ranging from 802.11-based wireless LANs (WLANs) that provides ubiquitous Internet access, to 802.15.4-based wireless sensor networks (WSNs) that carry out various mission-critical tasks such as security surveillance and patient monitoring. Many applications running over wireless networks impose stringent performance requirements including high system throughput and low communication delay. For instance, mobile video streaming over WLAN should achieve required bandwidth and delay to enhance user experience. Similarly, the ECG sensors for wireless patient monitoring must reliably report the cardiac rhythm data at desired rate for real-time diagnosis. However, despite the advances in the field of wireless networking, how to design high-performance wireless networks remains an open problem because of the fundamental challenges of wireless interference and coexistence.

1.1 Wireless Interference

Interference is the fundamental factor that hinders link concurrency of wireless networks. Due to the broadcast nature of wireless channel, concurrent transmissions on the same frequency interfere with each other over the air. At the receiver, collision occurs as the intended signal is distorted by simultaneously arrived interfering signals, impeding packet decoding and resulting in lower system throughput and higher delivery delay. In wireless networks, interference is typically handled at the link layer using medium access control (MAC) protocols. For example, in the most popular carrier sense multiple access (CSMA) protocol, a sender listens to the channel and verifies the absence of other traffic before transmission. If no ongoing transmission is detected, the sender transmits its packet immediately. Otherwise, the sender defers, waiting for a random time before next channel assessment.

Despite the significant research efforts made on link layer protocols, little attention has been paid on the interplay between medium access and bit rate diversity. Recent years have witnessed the phenomenal surge of bit rate in wireless communications. For example, the state-of-the-art 802.11n offers 31 bit rates in physical layer (PHY), ranging from 6.5 Mbps to 600 Mbps. Induced by different amount of redundancy in modulation and coding, different bit rates offer different trade-offs between transmission efficiency and reliability. Generally, higher bit rate is more efficient as less redundancy is used to deliver the data, while lower bit rate is more robust against noise and interference.

The increasing diversity of bit rates has enabled opportunities to unleash link concurrency in wireless networks. Specifically, links that are considered to be conflicting in conventional wisdom may successfully share the channel without interfering each other, if their bit rates are carefully selected. Therefore, by jointly optimizing medium access and bit rate selection, it

is possible to allow more links to transmit concurrently, leading to higher aggregate system throughput. However, being oblivious to the bit rate diversity, existing link layer design cannot fully exploit the link concurrency to improve the network performance.

1.2 Wireless Coexistence

Recent years have witnessed the unprecedented proliferation of wireless devices in the unlicensed 2.4GHz band. This band is open for a wide range of wireless technologies, including WiFi (802.11), ZigBee (802.15.4), and Bluetooth (802.15.1). Recent empirical studies [23] [9] [42] show that wireless coexistence is becoming a critical issue that affects the performance of wireless networks. When devices using different wireless technologies are operating on the same frequency, they will interfere with each other and lead to poor communication performance. Unfortunately, coexistence-induced interference cannot be resolved easily as devices cannot decode the signals of each other due to the heterogeneous PHY.

Another challenge of handling wireless coexistence stems from the power asymmetry of co-existing devices. For example, WiFi enabled laptops that provide ubiquitous Internet access typically operate using a transmitting power of tens dBm. In comparison, the maximum transmitting power of ZigBee-based devices, such as wireless sensors, is only 0 dBm. Power asymmetry leads to unfair channel usage among co-existing devices. This issue is particularly critical to low-power devices, such as ZigBee-based devices, as they must compete for the limited spectrum resources with WiFi devices that have much higher transmitting powers. Although this issue can be addressed by assigning orthogonal channels to ZigBee and WiFi, such a solution is often infeasible as the 2.4 GHz band is populated by ever increasing number of WiFi devices.

1.3 Contributions

This thesis tackles the challenges of wireless interference and coexistence to the design of link layer protocols. Motivated by the concurrency opportunities enabled by bit rate diversity and the challenge of wireless coexistence, this thesis investigates two problems in the design of existing link layer protocols, and advances the state-of-the-art by presenting the design, implementation and evaluation of practical solutions. The contributions of this thesis are summarized as follows.

1.3.1 Measurement-based Modeling of Interference and Coexistence

Based on extensive experiment studies, this thesis identifies *the rate-adaptive exposed terminal* (RET) problem and *the blind terminal* problem. We show that existing link layer protocols are surprisingly ineffective in handling these problems. Our experiments pinpoint the fundamental reasons of such ineffectiveness, and reveal their implications on the design of link layer protocols.

- *The Rate-Adaptive Exposed Terminal Problem.*

It is well known that CSMA-based wireless networks suffer the exposed terminal problem, where a sender is prevented from transmitting due to the contention of a nearby sender, although the interference is not strong enough to corrupt packet reception. Exposed terminals significantly hinder link concurrency of wireless networks. Recently, various protocols [59] [67] have been proposed to address the exposed terminal problem. However, a major drawback of existing solutions is that they study the ET problem

without considering the bit rate diversity of underlying PHY. Therefore they cannot fully exploit link concurrency to improve network performance.

In this thesis, we identify the RET problem, where the quality of a link is strong enough for successful packet delivery at certain bit rate, while its sender is prevented from transmitting by CSMA due to the contention of other sender(s). Our large-scale measurements on the topology of a production 802.11 a/g enterprise WLAN demonstrate the abundance of RETs, and quantify their significant impacts on the performance of wireless networks. Our micro-benchmarking experiments characterize the properties of RETs, and reveal the challenges of harnessing RETs in practice.

- *The Blind Terminal Problem.*

Our experiments reveal the blind terminal problem where existing link layer protocols fail to work in co-existing environments due to the heterogeneous PHY and power asymmetry of co-existing devices. Based on an empirical study of WiFi and ZigBee coexistence, we show that WiFi blind terminals are the major cause of poor ZigBee performance in co-existing environments. Using real-life WiFi traffic trace, we statistically characterize the interference caused by WiFi blind terminals. Moreover, an analytical framework is developed to model the performance of a ZigBee link co-existing with WiFi transmitters.

1.3.2 Harnessing Rate-Adaptive Exposed Terminals

We present the design, implementation and evaluation of TRACK, a novel protocol of *Transmission Rate Adaptation for Colliding linKs*, to harness RETs in enterprise WLANs leveraging the bit rate diversity of 802.11. Exploiting the backend wired LAN that con-

nects access points (APs) as a control channel, TRACK jointly configures the bit rate and medium access of concurrent links. The scheduling algorithm of TRACK is designed to optimize different metrics of system performance such as link fairness and aggregate throughput. We implement TRACK on commodity 802.11 devices and evaluate its performance through extensive experiments on a WLAN testbed of 17 nodes. Our results show that TRACK improves system throughput by 67% and 35% over 802.11 CSMA and a state-of-the-art protocol, without compromising link fairness.

1.3.3 Tackling the Blind Terminal Problem

We develop WISE - a WhItE Space aware framE adaptation protocol for ZigBee devices to tackle the WiFi blind terminals. Exploiting the white space in WiFi traffic, WISE allows ZigBee networks co-existing with WiFi to achieve assured performance. Specifically, WISE adopts a model-driven approach to predict the length of white space in WiFi traffic, and intelligently adapts frame size to maximize the throughput efficiency while achieving assured packet delivery ratio. WISE is implemented in TinyOS and ZigBee-based wireless sensors. Our extensive experiments on a testbed of 802.11 laptops and 802.15.4 TelosB motes show that WISE achieves 4x and 2x performance gains over the default MAC protocol of 802.15.4 and a recent opportunistic transmission scheme, respectively, while only incurs 10.9% and 39.5% of their overhead.

1.4 Organization of Thesis

The rest of this thesis is organized as follows. Chapter 2 introduces the background and motivation of this thesis. Chapter 3 experimentally identifies the problems of rate-adaptive

exposed terminals and blind terminals. Chapter 4 discusses TRACK, a novel protocol to harness rate-adaptive exposed terminals. Chapter 5 presents WISE to tackle the blind terminal problem in wireless coexistence. Chapter 6 concludes this thesis.

Chapter 2

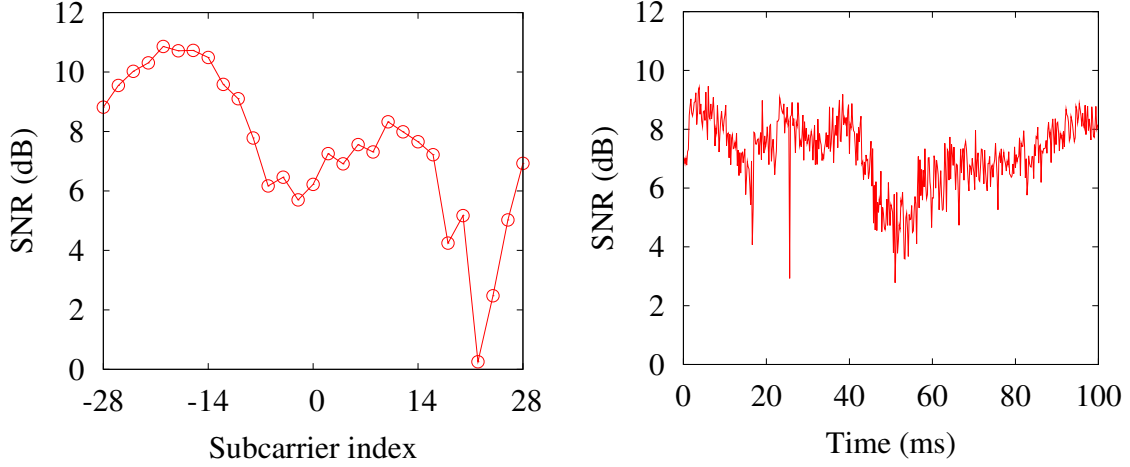
Background and Motivation

In this chapter, we present the background of this thesis. We first discuss the basic characteristics of wireless channel that make the design of link layer protocols challenging, and then introduce the background of link layer protocols.

2.1 The Wireless Channel

Communication over wireless channel is challenging because of the effects of *fading* and *interference*. In the following, we discuss these effects respectively.

In wireless communications, fading is deviation of the attenuation affecting a signal over certain propagation media. Due to reflections occurred in the environment surrounding the sender and receiver, a transmitted signal traverses multiple paths during propagation. At the receiver, this leads to a superposition of multiple copies of the signal, which differ in amplitude, phase and frequency. Their superposition will result in either constructive or destructive interference. Such a phenomenon is referred to as fading. The fading may vary with frequency or time, as discussed in the following.



(a) The measured SNR varies with frequency in a channel of 20MHz. The channel is divided into 56 subcarriers of 312.5KHz.

(b) The measured SNR varies during a period of 100 *ms*.

Figure 2.1: Channel fading varies with time and frequency. The results are measured on an 802.11n link that consists of two laptops equipped with Intel WiFi 5300 adapters. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

1. *Flat versus frequency-selective fading.*

As the signal traverses multiple paths of different length, its copies arrive at the receiver at different times. Depending on how the signal copies are spread out in time, channel fading varies across the frequency band of the signal. The fading is frequency selective, if different frequency components of the signal experience uncorrelated fading.

2. *Slow versus fast fading.*

When the sender, the receiver or any of the objects surrounding the sender and receiver is moving, the velocity of the moving object causes Doppler shift. Signal copies travelling along different paths suffer different amounts of Doppler shift, resulting in different rates of change in phase. Since fading depends on how signal copies add up at the receiver, it varies with the changes in the phases of signal copies. The channel

is fast fading if it becomes uncorrelated to its previous value within a short period of time.

Fig. 2.1 illustrates channel fading by showing the variations of signal-to-noise ratio (SNR) with time and frequency. The results are measured using an 802.11n link that consists of two laptops equipped with Intel WiFi 5300 adapters. Fig. 2.1(a) plots the effect of frequency selective fading in a channel of 20 MHz. The channel is divided into 312.5 KHz bands called subcarriers. We observe that SNR of the channel varies significantly with subcarriers. The most faded subcarrier has a SNR of 0 dB, which is 11 dB lower than the peak. Fig. 2.1(b) plots the SNR trace of a fast fading channel. The results are measured in a time window of 100 *ms*. We observe that the SNR varies significantly with time. Specifically, the SNR measured around 50 *ms* is 6 dB lower than the peak.

Interference is critical to the performance of wireless networks. Due to the broadcast nature of wireless channel, concurrent transmissions on the same frequency interfere with each other over the air. At the receiver, the intended signal is distorted by simultaneously arrived interfering signals. Error occurs if the distortion is too significant that the original signal cannot be recovered. In wireless communication, Signal to Interference plus Noise Ratio (SINR) is commonly used to quantify the intensity of interference. Given an intended signal α and a group of interfering signals $\{\beta_1, \dots, \beta_k\}$, the SINR observed at receiver can be calculated as follows,

$$\text{SINR}_{dB} = 10 \times \log_{10} \frac{\mathcal{P}(\alpha)}{\eta + \sum_i^k \mathcal{P}(\beta_i)} \quad (2.1)$$

where $\mathcal{P}(\cdot)$ gives the signal power measured in *mw*; η is the noise floor at receiver.

In theory, the packet reception ratio (PRR) of a link depends on the SINR at the receiver.

However, due to variations in the design and implementation of commodity transceivers, the mappings from SINR to PRR usually differ across hardware. To predict the link performance using SINR, calibration is usually required to build accurate SINR-PRR models.

2.2 Link Layer Protocols

Designing the link layer of wireless networks is challenging because of the fading and interference in wireless channels. First, since the fading in wireless channel varies with time and frequency, link layer protocols must adaptively tune themselves to maintain satisfactory performance. Moreover, the broadcast nature of wireless channel calls for the design of efficient MAC protocols to arbitrate conflicting transmissions that result in interference. In the following, we introduce how these challenges are addressed in existing link layer protocols.

2.2.1 Bit Rate Adaptation

In wireless networks, the common practice of combating the variation of channel fading is to adapt the *bit rates* of packet transmissions. In the following, we first introduce the background on bit rate, and then discuss rate adaptation protocols.

2.2.1.1 Background on Bit Rate

In wireless communication, bit rate is the number of bits transmitted per unit of time. Induced by different amount of redundancy used in data transmission, different bit rates offer different trade-offs between transmission efficiency and reliability. Generally, higher bit rate is more efficient as less redundancy is used in transmission, while lower bit rate is more robust against fading and interference. Specifically, bit rate is determined by the *forward*

error correction (FEC) code and the modulation scheme used in the physical layer. In the following, we introduce the background of FEC and modulation, respectively.

1. *FEC codes.*

FEC is the common practice for controlling errors in transmitting messages over unreliable and noisy channels. The basic idea of FEC is to encode the original message into a coded message by injecting some redundant bits. If some of the coded bits are flipped during transmission, the decoder tries to utilize the redundancy to recover the faulty bits without retransmission. Assuming that a FEC code encodes an original message of m bits into a coded message of n bits, the rate of the FEC code is the ratio of the size of the original message to the size of coded message, denoted by m/n . The correcting capability of a FEC code depends on its rate. The lower the rate, the more robust the transmission against noise and interference.

There are two main categories of FEC codes, including *block codes* and *convolutional codes*. Block codes work on fixed-size blocks of bits of predetermined size. A rate m/n block code encodes each m -bit symbol in the original message into an n -bit *codeword*. At the receiver, the decoder maps each received codeword to the ‘closest’ valid codeword, where the distance between two codewords can be measured using various metrics such as Hamming distance.

On the other hand, convolutional codes work on bit streams of arbitrary length, where each m -bit information symbol in the original message is encoded into an n -bit coded symbol. Each coded symbol is the function of last k information symbols, where k is the *constraint length* of the code. The basic idea of convolutional decoding is to find a sequence of bits, which can be encoded into a coded message that is ‘closest’ to the

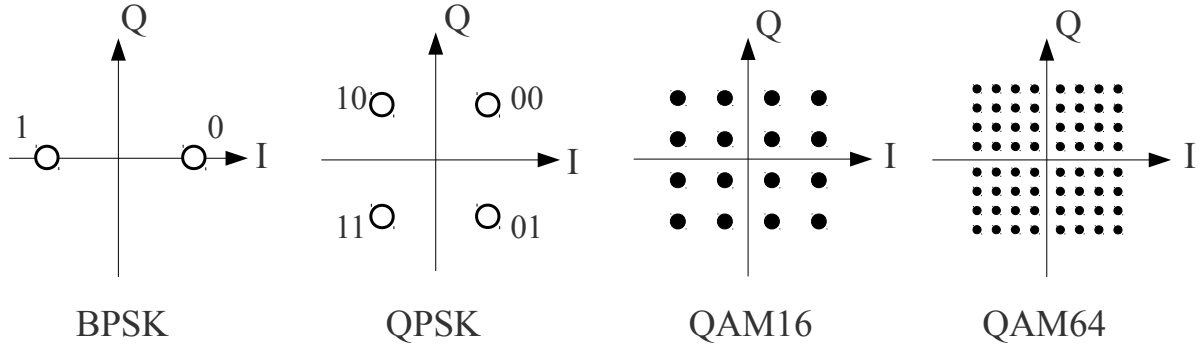


Figure 2.2: Constellation diagrams of BPSK, QPSK, QAM16, and QAM64.

received message.

2. Modulation.

To transmit the coded message over wireless channel, the transmitter maps a sequence of coded bits into a complex symbol using different modulation schemes. The complex symbol is represented by a position on a 2D complex plane called *constellation diagram*. Fig. 2.2 shows the constellation diagrams of four modulation schemes, including BPSK, QPSK, QAM16, and QAM64. These modulation schemes differ in the number of bits carried by each symbol. For example, in BPSK, two symbols are defined on the constellation diagram, at $+1+0j$ and $-1+0j$, respectively. In modulation, bit '1' and bit '0' are mapped to $+1+0j$ and $-1+0j$, where each symbol carries one bit. In comparison, QPSK defines four symbols on the constellation diagram. The bits '00', '01', '10', and '11' are mapped to $+1/\sqrt{2} + 1/\sqrt{2}j$, $-1/\sqrt{2} + 1/\sqrt{2}j$, $+1/\sqrt{2} - 1/\sqrt{2}j$, and $-1/\sqrt{2} - 1/\sqrt{2}j$, where each symbol carries two bits. Therefore QPSK is more efficient than BPSK in transmission.

Since the channel distorts the transmitted symbols, received symbols get dispersed around their ideal positions. The receiver demodulates the received symbol by mapping it to the closest valid symbol on the constellation diagram. Let s and r to be the

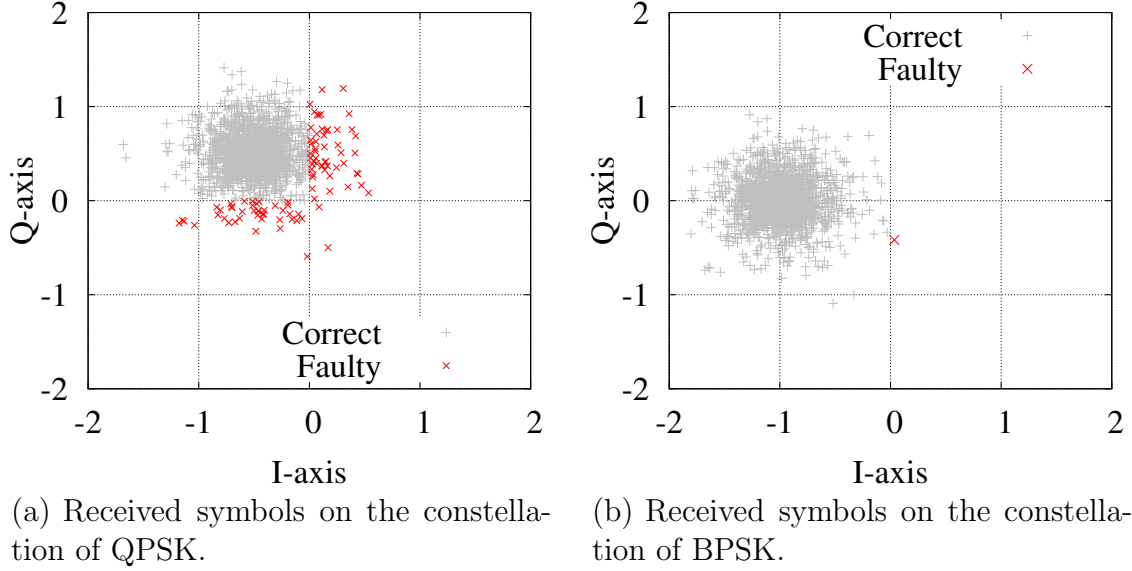


Figure 2.3: The effects of noise on QPSK and BPSK.

positions of transmitted and received symbols, error occurs if the channel-induced dispersion is too large that r is closer to the position of another valid symbol than s . Therefore modulation schemes that use sparse constellation diagrams are more robust against noise and interference.

Fig. 2.3 illustrates the effects of noise on QPSK and BPSK. The results are measured using a USRP/GNURadio link deployed in an office. In our measurement, the sender transmits 5000 identical symbols using QPSK. All the symbols are positioned at $-1/\sqrt{2} + 1/\sqrt{2}j$. Due to the effects of channel-induced distortions, the received symbols are dispersed around the ideal position, as shown in Fig. 2.3(a). We observe significant demodulation errors as a lot of received symbols are closer to another symbol than $-1/\sqrt{2} + 1/\sqrt{2}j$. However, if the symbols are transmitted using BPSK, the same level of distortions can be tolerated with significantly reduced demodulation errors, as shown in Fig. 2.3(b). The results demonstrate the need of modulation adaptation based on channel conditions.

Table 2.1: Bit rates defined in 802.11g.

Bit rate (Mbps)	Modulation	Coding rate
6	BPSK	1/2
9	BPSK	3/4
12	QPSK	1/2
18	QPSK	3/4
24	QAM16	1/2
36	QAM16	3/4
48	QAM64	2/3
54	QAM64	3/4

In practice, modulation and FEC together determine the bit rate of transmission. For, example, 802.11g offers 8 bit rates, ranging from 6 Mbps to 54 Mbps. Tab. 2.1 lists the combinations of modulation scheme and the rate of FEC used by 802.11g.

2.2.1.2 Rate Adaptation protocols

The fundamental objective of bit rate adaptation is to tune the redundancy used in packet transmissions, such that the throughput of wireless link is maximized given the current condition of channel fading. To this end, rate adaptation protocols need a responsive metric that accurately quantifies the quality of wireless channel at runtime. Based on the types of employed metric, existing protocols can be divided into two categories.

- *Packet delivery statistics based bit rate adaptation.*

Previous protocols [6] [10] [32] [64] adapts bit rate using packet delivery statistics. Specifically, PRR measured at the transmitter is commonly employed to characterize channel condition. Packet delivery statistic is simple to measure on commodity hardwares. However, it suffers the problem of less responsive to the dynamic of wireless channels. Recent study [30] shows that in wireless channels with fast fading, PRR-

driven rate adaptive protocols usually lead to poor communication performance.

- *PHY measurement based bit rate adaptation.*

Recently, various rate adaptation protocols have been proposed to exploit PHY measurement to quantify channel quality. For example, to estimate the channel condition, CHARM [30] and SGRA [72] use SNR measured at runtime; SoftRate [66] uses bit error rate (BER) estimated using per bit decoding confidence; AccuRate [56] computes channel-induced symbol dispersions; eSNR [26] advances SNR-based protocols by measuring the channel state information (CSI) to account for frequency-selective fading. Compared with the protocols driven by packet delivery statistics, PHY metric driven protocols are more responsive to channel variations. However, PHY metrics are difficult to measure on commodity hardware.

Existing rate adaptation protocols exclusively focus on handling channel fading without considering the effect of interference. For each packet transmission, the adaptation protocol picks one bit rate that is guessed to work best for an *interference-free* channel, relying on MAC protocols to arbitrate all conflicting transmissions. In Section 3.2, our experiments show that existing approaches significantly hinder the link concurrency of wireless networks.

2.2.2 Medium Access Control

In wireless networking, interference is typically controlled at the link layer using MAC protocols. The fundamental objective of MAC protocol design is to balance interference and link concurrency. Existing MAC protocols can be divided into three categories.

2.2.2.1 CSMA-based protocols

CSMA is the standard MAC protocol in most popular wireless technologies, including 802.11 and 802.15.4. In CSMA, a sender listens to the channel and verifies the absence of other traffic before transmission. If no ongoing transmission is detected, the sender transmits its packet immediately. Otherwise, the sender defers, waiting for a random time before next channel assessment.

Despite its efficiency and simplicity, CSMA is well known to be an imperfect solution. The fundamental reason is that CSMA decides whether a node should transmit or not using the channel condition measured at the sender, but the successful packet reception depends on the channel condition at the receiver. Problem arises when the sender and receiver observe different interference conditions, leading to *the hidden terminal problem* and *the exposed terminal problem*. Given a link l , where S and R are the sender and receiver, respectively, the hidden terminal and exposed terminal are defined as follows.

Definition 1 The hidden terminal (HT) problem. *A node T is referred to as a hidden terminal of link l , if T 's transmission cannot be sensed by S , but interferes with the packet reception of R .*

Definition 2 The exposed terminal (ET) problem. *A node T is referred to as an exposed terminal of link l , if T 's transmission can be sensed by S , but does not interfere with the packet reception of R .*

Addressing the ET problem is critical in wireless networks as ETs significantly hinder link concurrency. Recent empirical studies show that ETs are common in production WLANs. In [59], 41% of the links in a WLAN suffer from the ET problem, whose throughput can be

doubled when concurrent transmissions were allowed. Various protocols have been proposed to address the ET problem. For example, CMAP [67] harnesses ETs by opportunistically disable carrier sense when non-conflicting links are transmitting. CENTAUR [59] periodically measures conflict graph [7], and leverages centralized scheduling to mitigate downlink HTs and ETs in enterprise WLANs. However, as we will discuss in Section 2.3, a major drawback of existing solutions is that they either assume a uniform bit rate across the network [67] or only adapt bit rates of ETs in interference free scenarios [59]. Without accounting for the bit rate diversity, they cannot fully exploit link concurrency in wireless networks.

2.2.2.2 TDMA-based protocols

TDMA-based protocols divide the time into slots, and assign conflicting transmissions into different slots to avoid interference. For example, Bluetooth (802.15.1) employs a TDMA-based MAC layer where the transmissions of slaves are assigned to different time slots. Rao et al. [53] develop an overlay MAC layer (OML) for 802.11 networks. OML divides the time into slots using loosely-synchronized clocks, and employs a distributed algorithm to allocate the slots among competing nodes. Raman et al. [52] propose 2P, a TDMA-based protocol for long-distance 802.11 mesh networks. WiLDNet [46] leverages and builds on top of 2P, using loose time synchronization to determine the TDMA schedule in lossy environment. Similarly, various TDMA-based protocols [16] [58] [39] [34] [15] [48] [51] [33] have been developed for WSNs.

2.2.2.3 Multi-channel protocols

Multi-channel protocols mitigate interference by assigning different channels to concurrent links. For example, Bahl et al. [8] propose *slotted seeded channel hopping* (SSCH) for capacity

improvement in 802.11 networks. In SSCH, each node switches channels such that nodes desiring to communicate share the same channel, while disjoint communications occur on orthogonal channels, and hence do not interfere with each other. So et al. [60] utilize multiple channels to mitigate multi-channel hidden terminals. The problem of channel assignment is extensively studied in the context of WLANs. In [43], authors model a wireless LAN as a weighted graph with the set of vertices denoting the access points. They present centralized and distributed algorithms for performing channel assignment via graph coloring. Mishra et al. [44] present a model to characterize interference in overlapping channels. The model is then incorporated in existing multi-channel protocols to improve network performance. Similarly, various multi-channel protocols [35] [36] [70] have been developed for wireless sensor networks to mitigate interference.

Recent work [11] [45] [54] propose to exploit adaptive channel width to improve network performance. Chandra et al. [11] propose SampleWidth to improve the throughput of a single link by adapting the channel width over which transmitters spread their signals. Motivated by this result, Moscibroda et al. [45] study the problem of load-balancing in WLANs by leveraging dynamic-width channels, where every AP adaptively adjusts their spectrum usages based on their traffic loads. Similarly, FLUID [54] improves system throughput of enterprise WLANs by exploiting flexible channelization, where both of the channel width and center frequency of each transmission are optimized to minimizing interference.

2.3 Motivation

This thesis is motivated by the challenges of wireless interference and coexistence to the design of link layer protocols.

1. *Unleashing Link Concurrency*

Interference is the fundamental factor that limits the link concurrency of wireless networks. Despite the extensive research efforts made for balancing the interference and link concurrency using MAC protocols, the increasing diversity of bit rates has enabled great opportunities for further unleashing link concurrency in wireless networks. Specifically, links that are considered to be conflicting using fixed bit rates may successfully share the channel without interfering each other, if their bit rates are carefully selected. However, existing MAC protocols cannot fully exploit link concurrency as they are oblivious to the bit rate diversity of underlying PHY.

2. *Tackling Wireless Coexistence*

Wireless coexistence poses several challenges to the design of link layer protocols. First, existing link layer protocols are exclusively designed for homogenous PHY. However, when devices of heterogeneous PHY working on the same frequency, their coordination is difficult as they cannot decode the signals of each other. Moreover, even if this issue is addressed (e.g., by adopting an energy-based approach to detect conflicting transmissions), there is still a large region in which low-power devices can sense high-power transmitters but not vice versa because of the power asymmetry. Therefore, wireless coexistence calls for a link layer design that is fundamentally different with conventional link layer protocols.

In Chapter 3, we conduct extensive experiments to demonstrate the ineffectiveness of existing link layer protocols in exploiting link concurrency and handling wireless coexistence. Motivated by the results, in Chapter 4 and Chapter 5, we present the design, implementation, and evaluation of practical solutions to the above problems.

Chapter 3

Measurement-based Modeling of Interference and Coexistence

Based on extensive experimental studies on the performance of existing link layer protocols, this chapter identifies two problems that are not addressed in existing literature, including (1) *the Rate-Exposed Terminal* (RET) problem where link concurrency cannot be fully exploited by conventional link layer protocols because of their oblivious to the bit rate diversity; and (2) *the blind terminal* problem where existing link layer protocols fail to work in co-existing environment due to the heterogeneous PHY and power asymmetry of co-existing devices. We show that existing link layer protocols are surprisingly ineffective in handling these problems. Our experiments pinpoint the fundamental reasons of such ineffectiveness, and reveal their implications on the design of link layer protocols. The contributions of this chapter are summarized as follows.

1. We quantify RETs on the topology of a large-scale production 802.11 a/g enterprise wireless LAN (WLAN). Based on extensive measurements involving 23 access points

(APs) and 104 pairs of links, we demonstrate the abundance of RETs in real-life WLAN deployment. Moreover, our experiment shows that link concurrency can be boosted by 131% through harnessing RETs using optimized bit rates.

2. We present micro-benchmarking results to characterize RETs. Our measurements reveal that exploiting RETs introduces new challenges that have not been addressed before. Different from the classical exposed terminal problem, simply allowing concurrent transmissions of RETs may lead to unfair channel usage and link starvation.
3. Based on an empirical study of ZigBee and WiFi coexistence, we reveal that WiFi nodes are often *blind terminals* of ZigBee nodes due to the heterogeneous PHY and transmit power asymmetry between ZigBee and WiFi. A WiFi blind terminal fails to detect ZigBee signals and hence can easily corrupt ongoing ZigBee packet reception, which is a major cause of poor ZigBee performance in coexisting environments.
4. We characterize the interference of blind terminals through extensive statistical analysis of data traces captured in real-life WiFi networks. We show that, in a channel shared by a group of WiFi blind terminals, WiFi frames that interfere with ZigBee transmissions are highly clustered and the arrival process of clusters has the feature of *self-similarity*. We then present a Pareto model that accurately characterizes the white space between WiFi frame clusters.
5. We propose an analytical framework that models the performance of a ZigBee link in the presence of WiFi blind terminals. Based on the white space model, we derive the expected frame collision probability of ZigBee. The results will help a network designer analyze and predict the performance of ZigBee links coexisting with WiFi networks.

The rest of this chapter is organized as follows. Section 3.1 reviews related work. Section 3.2 studies the rate-adaptive exposed terminal problem. Chapter 3.3 identifies the blind terminal problem. Section 3.4 summarizes this chapter.

3.1 Related Work

Carrier sense multiple access (CSMA) is the standard MAC protocol in most wireless technologies, including 802.11 and 802.15.4. Despite its popularity, CSMA is well-known for its imperfection, yielding the problems of hidden terminals (HTs) and exposed terminals (ETs). Recent empirical studies show that hidden terminals and exposed terminals are common in production WLANs. Cheng et al. [13] presented a detailed performance study in a building-wide WLAN. It shows that HTs are the dominant cause of packet losses. Shrivastava et al. [59] demonstrated that 41% of the links in a commodity WLAN suffer from ET problem, whose throughput can be doubled when concurrent transmissions were allowed. Despite the extensive studies on hidden terminals and exposed terminals, few efforts have been made on studying the interplay between channel access and bit rate diversity. In comparison, our measurements presented in Section 3.2 show the significant impact of bit rates on link concurrency, and demonstrate the great opportunity of exploiting bit rate diversity to improve link concurrency in WLANs.

Interference measurement and modeling is critical to the performance of ZigBee-based low-power wireless networks. Maheshwari et al. [40] characterized the impact of interference on the performance of ZigBee links. The signal to interference and noise ratio (SINR) and packet reception ratio (PRR) model is empirically built for predicting link performance. Son et al. [62] showed that the SINR-PRR model depends on both received signal strength

and the number of interferers. Different with existing work, the measurements presented in Section 3.3 is conducted in the context of WiFi and ZigBee coexistence. We present a statistical study to characterize WiFi interference and develop an analytical framework to model ZigBee performance.

Traffic modeling [47] [37] [14] is a fundamental problem in the Internet community. The self-similarity of Internet traffic has also been observed [37] [14]. Different from existing studies on Internet traffic modeling, we focus on characterizing the white space in link-level traffic of WiFi channel shared by a group of users. The most similar work to our study is [20]. Compared to our work, the empirical white space model proposed in [20] is built for specific applications, such as FTP, VOIP and Skype. However, in real scenarios with diverse applications, the traffic is highly bursty at a wide range of time scales, which is not considered in [20]. In Section 3.3.3, we build white space model based on real traffic traces, and examine the modelability of white space in different time scales.

3.2 The Rate-adaptive Exposed Terminal Problem

It is well known that CSMA-based wireless networks suffer from the exposed terminal (ET) problem. Previous work [59] [67] attempts to harness ETs to improve link concurrency. However, the impact of bit rates on ETs has not been studied. Modern WLANs offer rate diversity for trading off the efficiency and reliability of wireless communication. Generally, higher rate is efficient in modulation and coding, while lower rate is robust against noise and interference. In this section, we revisit the classical ET problem in the context of rate-diverse WLANs. We define the *Rate-adaptive Exposed Terminal (RET)* problem as follows

Definition 3 Rate-adaptive exposed terminal (RET). *Given two links $l_0 = \langle s_0, r_0 \rangle$ and $l_1 = \langle s_1, r_1 \rangle$, where s_i and r_i are sender and receiver respectively, node s_0 is a rate-adaptive exposed terminal (RET) of l_1 , if (1) s_1 can deliver its packets to r_1 at some bit rate R , but (2) s_1 is prevented from transmission due to the contention of s_0 .*

In this section, we experimentally study RETs based on the topology of a production 802.11a/g WLAN. In the following, we first introduce our experiment methodology, and then present and discuss measurement results.

3.2.1 Experimental Methodology

The goal of our measurement study is to quantify and characterize RETs in rate-diverse WLANs. Since the existence of ET and RET depends on the choice of MAC protocol, we employ three baseline methods in this study, including 802.11 CSMA, concurrent transmission (CT) and concurrent transmission with rate optimization (CTRO).

- *802.11 CSMA* is the default MAC protocol of 802.11-based WLAN. The bit rate is controlled by a state-of-the-art algorithm [26], which is designed for handling fading in interference-free scenarios.
- *Concurrent transmission (CT)* simply disables carrier sense on both links to enable simultaneous channel access. CT also employs the algorithm proposed in [26] for rate control, which selects the rate learned in an interference-free channel for concurrent transmissions. We note that allowing concurrent transmissions without rate adaptation is the general method of harnessing ETs [59] [67]. For a pair of *contending* links where the two senders lie in the carrier sense range of each other, loss-free delivery of CT indicates the existence of ETs.

- *Concurrent transmission with rate optimization* (CTRO) extends previous solutions by harnessing ETs using the bit rates that optimize the aggregate throughput of concurrent links. In order to study the potential of bit rate diversity, we exhaustively try all possible combinations of bit rates, and measure the resulted optimal throughput. By Definition 3, for a pair of contending links, loss-free packet delivery of CTRO indicates the existence of RETs.

3.2.2 Large-scale Quantification of RETs

We quantify downlink RETs based on the topology of a large-scale 802.11a/g WLAN deployed in a four-story office building. Our measurement involves 23 APs deployed on the 3rd floor. We focus on studying downlink RETs as most traffic in enterprise WLANs is downlink, i.e., transmitted from AP to clients [18]. We evaluate and compare the performance of 802.11 CSMA, CT and CTRO to study the impact of RETs. The implementation of CT and CTRO requires disabling carrier sense at senders. Unfortunately, it was not feasible to modify the driver of production APs. To address this challenge, we use our own 802.11 nodes at both ends of a link, and deploy them to mimic the topology of production WLAN. For each measurement, we deploy two links with the senders placed close to the positions of production APs. Each node is a laptop equipped with 802.11 radio using Atheros AR928x chips, running mac80211 driver [5] with ath9k [1]. As RETs only exist among contending links (see Definition 3), the two APs in each measurement are selected such that they lie in the carrier sense range of each other. Our results are measured on total 104 link pairs. For each link pair, we measure the packet reception ratio (PRR) at the receivers. To minimize the effect of channel variation, the experiments is performed in a round-robin fashion, where

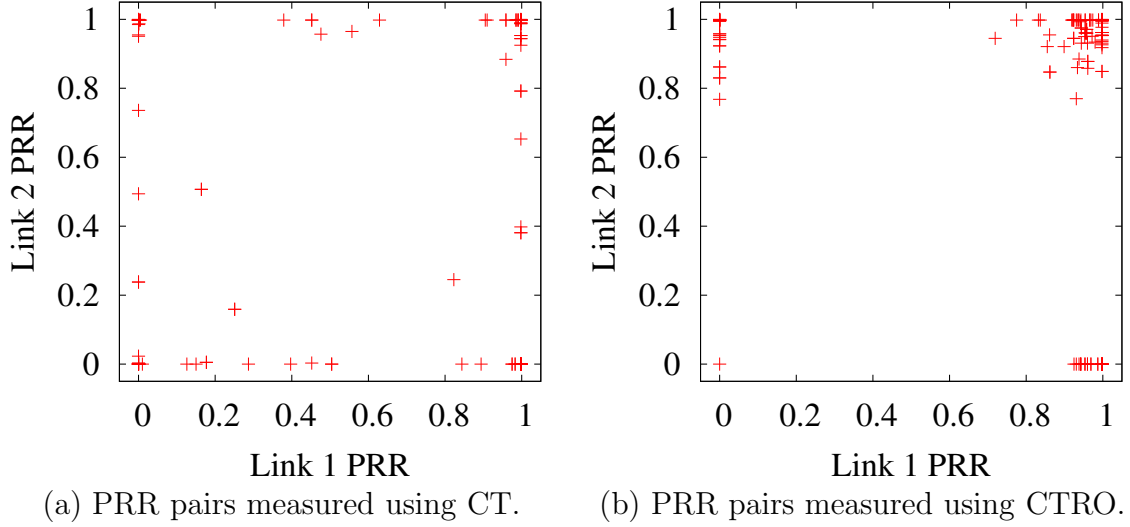


Figure 3.1: The PRRs of two links running CT and CTRO. The results are measured using 104 link pairs.

each baseline algorithm runs for 20 *ms* in each round. The reported result is an average over 1000 rounds of measurements.

We quantify RETs by studying PRR of CT and CTRO. As we discussed in Section 3.2.1, when both links achieve high PRR using CT, it implies that the two links are ET of each other. However, since the rate control algorithm of is designed for interference-free channels, it is oblivious to the concurrency opportunities enabled by bit rate diversity. In contrast, CTRO searches for the best combination of bit rates to tolerate interference, therefore allows more links to transmit concurrently. Fig 3.1(b) plots the measurement results. We observe that, compared with CT, CTRO significantly improves link concurrency. In particular, using CTRO, 67 link pairs in our measurement can achieve a PRR higher than 90%, indicating that $67/104=64.4\%$ of link pairs in our study are RETs of each other. In contrast, only $29/104=27.8\%$ of link pairs in our study can maintain a PRR higher than 90% when CT is used. Therefore, CTRO boosts link concurrency by 131% over CT. In other words, it enables concurrency on 50.6% of non-ET links by optimizing their bit rates.

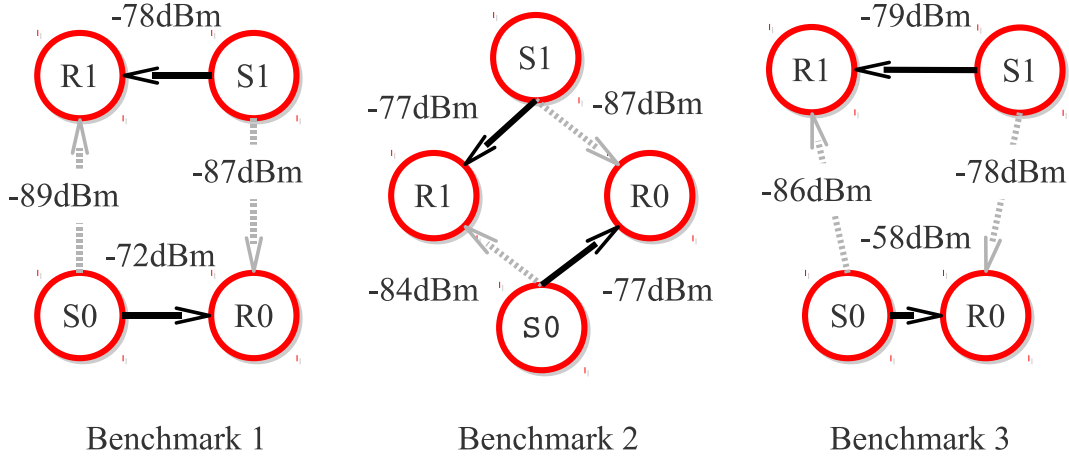


Figure 3.2: Benchmark topologies. Data and interference links are marked with solid and dash lines. Average signal strength is labeled on links.

3.2.3 Micro-benchmarking RETs

We further characterize the properties of RETs using three micro-benchmarking topologies, as shown in Fig. 3.2. We evaluate the throughput of the three baseline algorithms introduced in Section 3.2.1. As shown in the following, these micro-benchmarks represent three typical cases of RETs with different levels of interference between concurrent links.

Table 3.1: Throughput measured on the topology of benchmark 1.

Throughput (Mbps)	802.11 CSMA		CT		CTRO	
	Mean	STD	Mean	STD	Mean	STD
$s_0 \rightarrow r_0$	11.53	2.36	2.76	1.17	16.96	2.59
$s_1 \rightarrow r_1$	11.43	2.34	3.12	1.46	17.37	2.76
Aggregate	22.96	1.77	5.88	1.59	34.33	2.87

- *Benchmark 1.* Tab. 3.1 shows the throughput measured on the topology of benchmark 1. In benchmark 1, although concurrent transmission brings interference and results in significant packet loss under sub-optimal bit rate, the qualities of two links is good enough to support concurrent packet delivery at an alternative lower rate. Thus the two

links are RETs of each other. The result shows that CTRO outperforms 802.11 CSMA by more than 50%, while CT performs the worst as it is oblivious to the concurrency opportunities provided by bit rate diversity. The result of benchmark 1 demonstrates the potential of harnessing RETs for enhancing aggregate system performance.

Table 3.2: Throughput measured on the topology of benchmark 2.

Throughput (MBps)	802.11 CSMA		CT		CTRO	
	Mean	STD	Mean	STD	Mean	STD
$s_0 \rightarrow r_0$	10.06	2.77	1.20	0.59	7.61	1.20
$s_1 \rightarrow r_1$	10.11	2.72	0.91	0.31	7.87	1.29
Aggregate	20.27	1.61	2.11	1.18	15.49	1.35

- *Benchmark 2.* Tab. 3.2 shows the throughput measured on the topology of benchmark 2. Benchmark 2 gives another example of mutual RETs. Different with benchmark 1, we observe that concurrent transmissions results in strong interference on both links. Although both links can find a lower rate for reliable packet delivery, the aggregate throughput even decreases compared with 802.11 CSMA. This is because the links have to compromise their modulation and coding efficiency to tolerate the increased interference caused by concurrency. This result demonstrates that, unlike the classical ET problem, exploiting RETs improperly may degrade system performance.

Table 3.3: Throughput measured on the topology of benchmark 3.

Throughput (MBps)	802.11 CSMA		CT		CTRO	
	Mean	STD	Mean	STD	Mean	STD
$s_0 \rightarrow r_0$	11.04	1.97	1.69	0.73	5.32	1.14
$s_1 \rightarrow r_1$	10.95	2.5	11.47	1.93	22.86	2.08
Aggregate	21.99	1.76	13.17	1.83	28.19	2.09

- *Benchmark 3.* Tab. 3.3 shows the throughput measured on the topology of benchmark 3. The two links in benchmark 3 are also mutual RETs. In particular, the channel

quality of $s_0 \rightarrow r_0$ is much stronger than that of $s_1 \rightarrow r_1$. We observe that although CTRO outperforms 802.11 CSMA in terms of aggregate throughput, the throughput of $s_1 \rightarrow r_1$ drops significantly. Moreover, in the case of CTRO, the throughput of $s_1 \rightarrow r_1$ is 17 Mbps lower than that of $s_0 \rightarrow r_0$. The result shows that asymmetry in channel quality may lead to unfair exploitation of individual links when concurrent transmissions of RETs are enabled.

In summary, our measurements demonstrate the abundance of RETs in real-life WLAN deployments (Section 3.2.2). Specifically, 64.4% of link pairs in our study are RETs, while the percentage of ET is only 27.8%. By optimizing the bit rate of RETs, the link concurrency of WLANs can be boosted by up to 131%. Moreover, our benchmarking results in Section 3.2.3 show that harnessing RETs could significantly boost the link concurrency (Benchmark I). However, compared with classical ETs, harnessing RETs is more challenging as exploiting RETs improperly may degrade system throughput (Benchmark II) or result in unfair channel usage among concurrent links (Benchmark III).

3.3 The Blind Terminal Problem

Wireless coexistence is becoming a growing issue due to the proliferation of wireless devices, such as 802.11 b/g/n, ZigBee (802.15.4) and Bluetooth (802.15.1), in the crowded unlicensed 2.4GHz band. The issue is particularly critical for low-power wireless devices like ZigBee, as they must compete for the spectrum resources with other high-power devices such as WiFi enabled netbooks and smartphones.

Despite the ubiquitous deployment of WiFi hotspots in 2.4 GHz band, we observe that there exist abundant *white space* in real-life WiFi traffic, providing significant opportunities

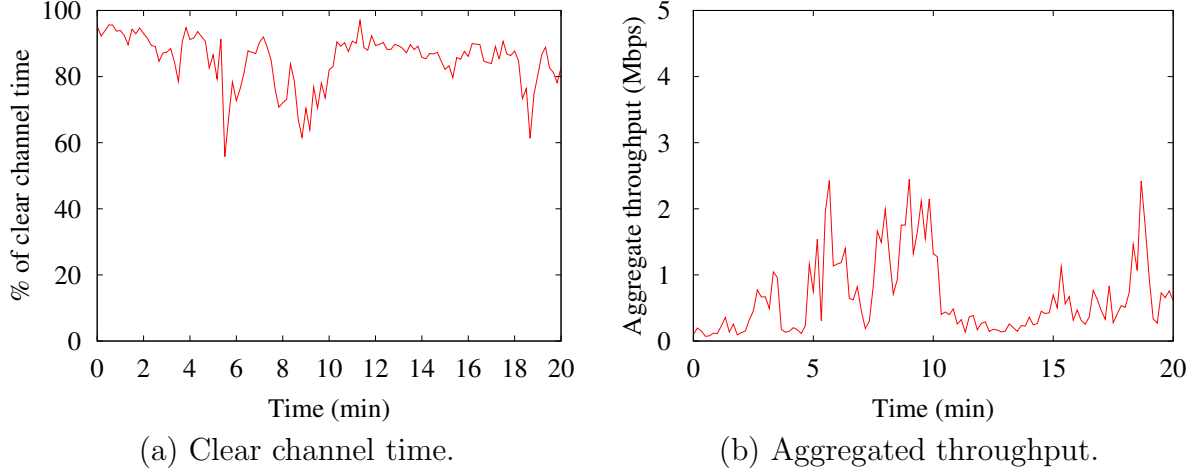


Figure 3.3: Channel utilization trace of a WiFi network consisting of 2 APs and 18 users.

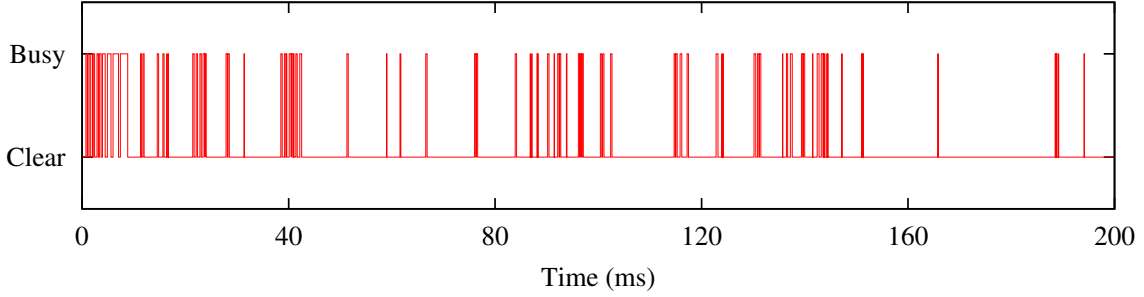


Figure 3.4: WiFi channel state trace.

for ZigBee and WiFi to coexist in the *same* or *overlapping* channels. Fig. 3.3 shows the channel utilization trace captured in a real-life 802.11-based network [12] consisting of 2 APs and 18 active users. It is clear that the channel is free in most of time. Although WiFi traffic surges from 5th to 10th minute, the channel is still free in more than 60% of time. Fig. 3.4 shows a typical trace of channel usage of the same WiFi network. We can see that the network traffic is highly bursty leaving significant amount of *white spaces* between 802.11 frames.

Motivated by the measurements given in Fig. 3.3 and Fig. 3.4, we ask a key question: does the existing WiFi and ZigBee MAC layers allow the white space to be efficiently uti-

lized? The answer to this question is crucial to the feasibility of coexisting ZigBee and WiFi networks within the same frequency domain. In this section, we show experimentally that CSMA, which is the default MAC protocol of 802.15.4, is surprisingly ineffective in utilizing the white space. In particular, ZigBee networks coexisting with WiFi networks often suffer from significant interference due to the *blind terminal* problem. Moreover, using real-life WiFi traffic trace, we statistically characterize WiFi blind terminals, and develop an analytical framework that models the performance of a ZigBee link under heavy interference of WiFi blind terminals. The results will help a network designer analyze and predict the performance of ZigBee links coexisting with WiFi networks.

3.3.1 An Empirical Study of ZigBee and WiFi Coexistence

We now illustrate the blind terminal problem by a case study. We deploy two TelosB motes equipped with 802.15.4 compliant CC2420 radios in an office. Both motes run the CSMA-based B-MAC [49]. ZigBee sender broadcasts at a fixed rate. A Linux netbook equipped with 802.11 compliant Intel Atheros 928x NIC serves as the interferer and is placed at different locations in the same office. We vary the position of WiFi interferer and measure the performance change of ZigBee link. 802.15.4 adopts multiple retransmissions to achieve reliable packet delivery under interference. However, this incurs extra energy consumption. To avoid the complication of retransmissions on the analysis of our results, we intentionally disabled them for ZigBee link. We note that this does not affect the conclusion of this study. The WiFi node runs a traffic generator [3] that generates a combination of UDP and TCP flows at a preset rate. Such a setting allows us to analyze the impact of interference at different traffic rates. We record (a) the WiFi signal power measured at ZigBee sender and

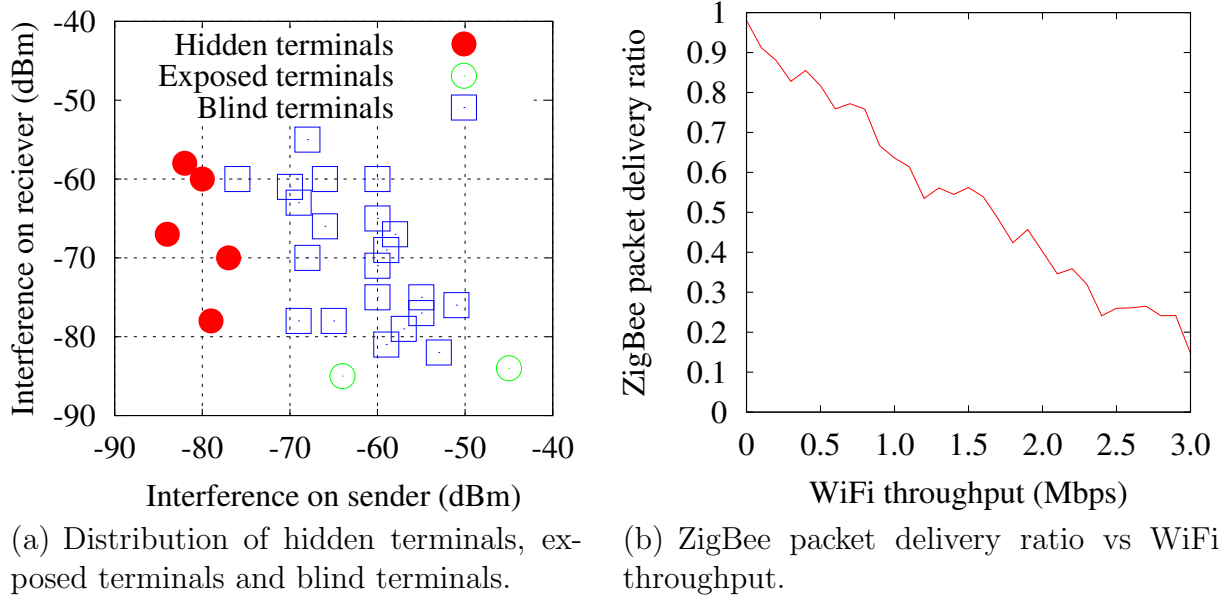


Figure 3.5: The blind terminal problem.

receiver from the received signal strength indicator (RSSI), (b) the sending rate of ZigBee, (c) the receiving rate of ZigBee, and (d) the sending rate of WiFi. In addition, we can calculate the ZigBee packet reception ratio (PRR) using (b) and (c).

Based on the experiment results, we can classify the role of WiFi node as hidden terminal, exposed terminal, or *blind terminal* depending on how it interferes ZigBee sender and receiver. Table 3.4 shows the condition of each role. Fig. 3.5(a) shows the distribution of three terminals in the space of interfering powers to the ZigBee sender (X axis) and receiver (Y axis). Each data point (x, y) in Fig. 3.5(a) corresponds to a different location of WiFi node whose signal strength is measured as x and y dBm by the ZigBee sender and receiver, respectively. For instance, the point $(-81\text{dBm}, -59\text{dBm})$ represents a hidden terminal because the experiment results measured satisfy: a) the sending rate of ZigBee does not change, and b) the PRR of ZigBee link drops. In such a scenario, the ZigBee sender cannot sense the transmissions of WiFi (due to the weak signal power of -81 dBm) while the receiver is strongly interfered (with power of -59 dBm). Similarly, we can identify $(-45\text{dBm}, -84\text{dBm})$

Table 3.4: The role of WiFi interferer.

Hidden terminal	The WiFi node is located within the interference range of ZigBee receiver, but outside the carrier sense range of ZigBee sender.
Exposed terminal	The WiFi node is located within the carrier sense range of ZigBee sender, but outside the interference range of ZigBee receiver.
Blind terminal	The WiFi node is located within both the carrier sense range of ZigBee sender and the interference range of ZigBee receiver.

as an exposed terminal, since the sending rate of ZigBee decreases while the PRR remains high. Hidden and exposed terminals have been well studied before. However, our results also indicate the existence of blind terminals (grey points in Fig. 3.5(a)) where both the sending and receiving rates of ZigBee decrease.

To further study the blind terminal problem, we select one blind terminal position, and vary the traffic rate of WiFi to examine its impact on ZigBee link performance. Fig. 3.5(b) shows that the PRR of ZigBee link drops with the increasing traffic rate of WiFi. In addition, we observe that the sending rate of WiFi strictly follows the rate we set in the traffic generator. This result shows that the WiFi sender fails to sense the transmissions of ZigBee.

3.3.2 Analysis of Results

The performance degradation of ZigBee in the presence of blind WiFi terminals is mainly caused by the following two reasons.

- *The heterogeneous PHY layer.* Commodity WiFi NICs typically conduct clear channel assessment by carrier sensing, i.e., declare channel busy only when valid 802.11-modulated signal is detected [24]. As a result, WiFi transmitters cannot sense ZigBee signals and hence do not defer their transmissions even when there exist ongoing ZigBee

packet transmissions. Therefore, WiFi signals can easily corrupt the ongoing reception of ZigBee packets. Although the WiFi’s blindness to ZigBee transmitters can be alleviated by adopting different carrier sensing mechanisms (e.g., energy-based clear channel assessment (CCA)), unfortunately, off-the-shelf WiFi drivers do not provide such an option. Even such an option is available, adopting it for the existing WiFi deployments poses a major management challenge.

- *Power asymmetry.* The second cause of the blind terminals is that the transmit power of 802.11 devices is much higher than ZigBee. In particular, the maximum transmit powers of WiFi and ZigBee are 14 and 0 dBm, respectively. Therefore, even if WiFi MAC layer adopted energy-based CCA, there is still a large region in which ZigBee transmitters can sense WiFi transmitters but not vice versa. As a result, the traditional approaches to dealing with interference, such as RTS/CTS exchanges, cannot effectively handle the blind terminal problem.

3.3.3 Modeling White Space in Real-life WiFi Networks

We now characterize the interference of blind terminals by studying the *white space* between WiFi frames that interferes with ZigBee transmissions. Specifically, we study how to model the temporal white space of WiFi networks. The white space model will be used to control the frame transmissions of ZigBee in presence of WiFi blind terminals (see Chapter 5).

In the following, we first conduct extensive statistical analysis on data traces captured in real-life WiFi networks. We show that, in a channel shared by a group of WiFi blind terminals, the arrival process of aggregate WiFi frame clusters has the feature of self-similarity. We then study in what time scale the temporal white space of WiFi is modelable. Finally

we present a Pareto model that accurately characterizes the white space.

3.3.3.1 WiFi Frame Clustering

As shown in Fig. 3.4, the arrival of WiFi frames is highly bursty and clustered. We observe that frames are clustered together with short intervals typically less than 1 ms, while the idle periods between clusters are significantly longer. The short frame intervals are attributed to the MAC layer contention mechanism of 802.11, in which senders back off for a short random time before each transmission.

According to 802.15.4 [27], the protocol header of ZigBee frame is 17 Bytes, which are transmitted at a rate of 250 Kbps. Thus the packet-in-air time of ZigBee is at least $544 \mu\text{s}$. After accounting for the software overhead (e.g., the delay introduced by CPU and radio interaction), the minimum packet transmission time of ZigBee approaches the maximum backoff window size of 802.11. Therefore, it is very difficult for ZigBee senders to utilize the short WiFi frame inter-arrival times for packet transmission. In the following, we will only focus on modeling the arrival process of WiFi *frame clusters* where each cluster may include multiple frames spaced by intervals less than 1 ms. We define the interval between frame clusters as *inter-cluster space* while the interval between the frames within the same cluster as *intra-cluster space*. Moreover, white space hereafter refers to inter-cluster space unless otherwise indicated.

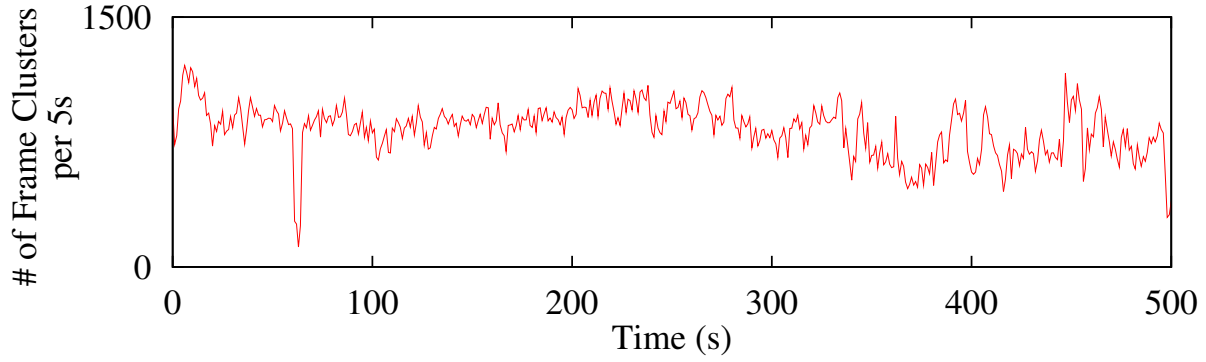
3.3.3.2 Self-Similarity of WiFi Frame Clusters

We plot the scaling behavior of the frame cluster arrival process in Fig.3.6. The data is captured in OSDI 2006 [12], which contains 150 consecutive minutes of monitored WLAN traffic in a channel shared by 2 APs and 18 active users. Fig.3.6 shows the number of arrived

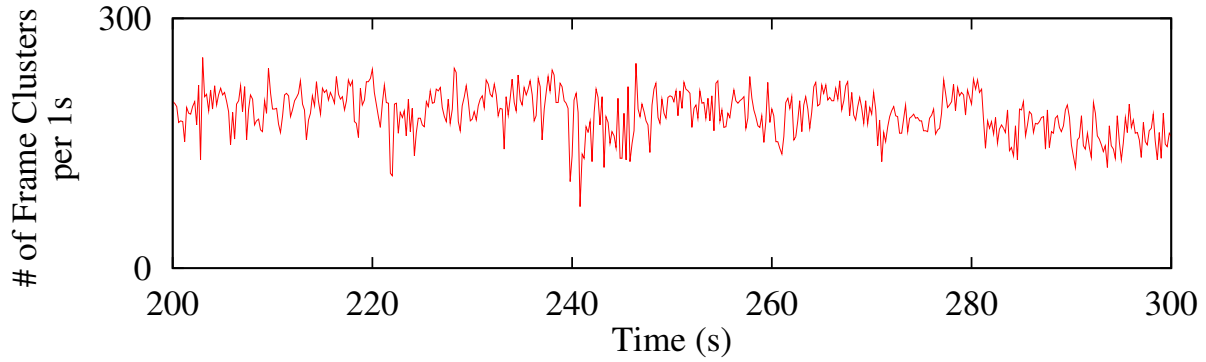
frame clusters for three time units: 5s, 1s and 0.2s. The plots show similar variance at all time scales. This time-scale invariant feature suggests that the arrival process of WiFi frame clusters is self-similar. In the following, we introduce the background on self-similarity, and use statistical and graphical tools to formally test the feature of self-similarity for 802.11 frame clusters. The results will enable us to model the distribution of WiFi white space.

Let $X = (X_t : t = 0, \dots, N)$ be a *covariance stationary stochastic* process with mean μ , and variance σ^2 . Define $X^{(m)} = \{X_k^{(m)}, k \in [0, \frac{N}{m}]\}$ to be the aggregated covariance stationary time series, obtained by averaging the original series over blocks of size m . Then X is *H-self-similar* if it has the same autocorrelation function $r(k) = \mathbb{E}[(X_t - \mu)(X_{t+k} - \mu)]/\sigma^2$ as the series $X^{(m)}$ for all m [14]. This means that the variances of the series are self-similar for all m , except for the change in scale. The degree of self-similarity is expressed by the *Hurst* parameter H , which describes the speed of decay of the series' autocorrelation function. For self-similar series, $1/2 < H < 1$. As $H \rightarrow 1$, the degree of self-similarity increases. We now use statistic and graphical tools to formally test the feature of self-similarity. These tools are described in [69], and widely used in traffic analysis literature [37] [14].

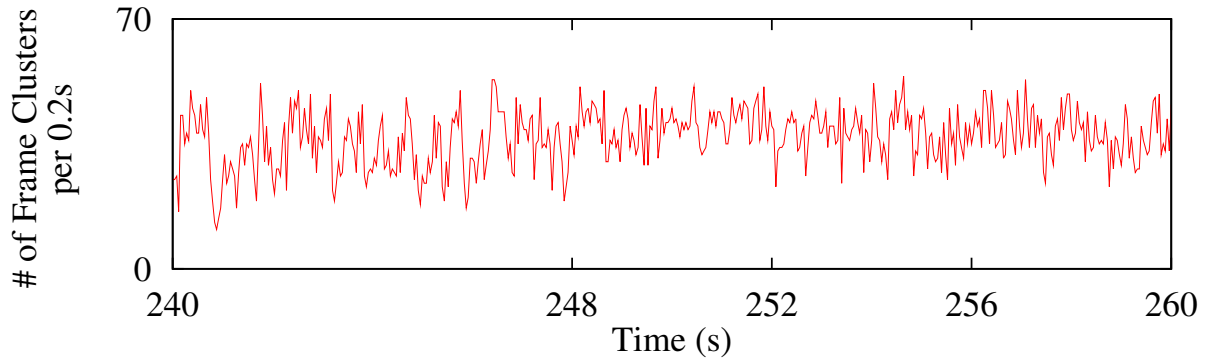
- *Rescaled range statistics (R/S method)* : The R-S method is based on the fact that for a self-similar time series $X = (X_t : t = 0, \dots, N)$, the *rescaled range*, R/S of series $X^{(m)}$ grows according to a power law with exponent H as a function of $\frac{N}{m}$. Thus for a given time series, the log-log plot of R/S against $\frac{N}{m}$ has a slop which is an estimate of the *Hurst* parameter H . Fig.3.7(a) gives the R-S plot for the data trace used in Fig.3.6. The result shows that the asymptotic slope of R/S plot is clearly between 0.5 and 1 (lower and upper dotted lines respectively), which suggests that the WiFi frame



(a) # of frame clusters per 5s. The data in window (200s, 300s) is shown in (b) using the time unit of 1s.



(b) # of frame clusters per second. The data in window (240s, 260s) is shown in (c) using the time unit of 0.2s.



(c) # of frame clusters per 0.2s.

Figure 3.6: Self-similarity of 802.11 frame cluster arrival process.

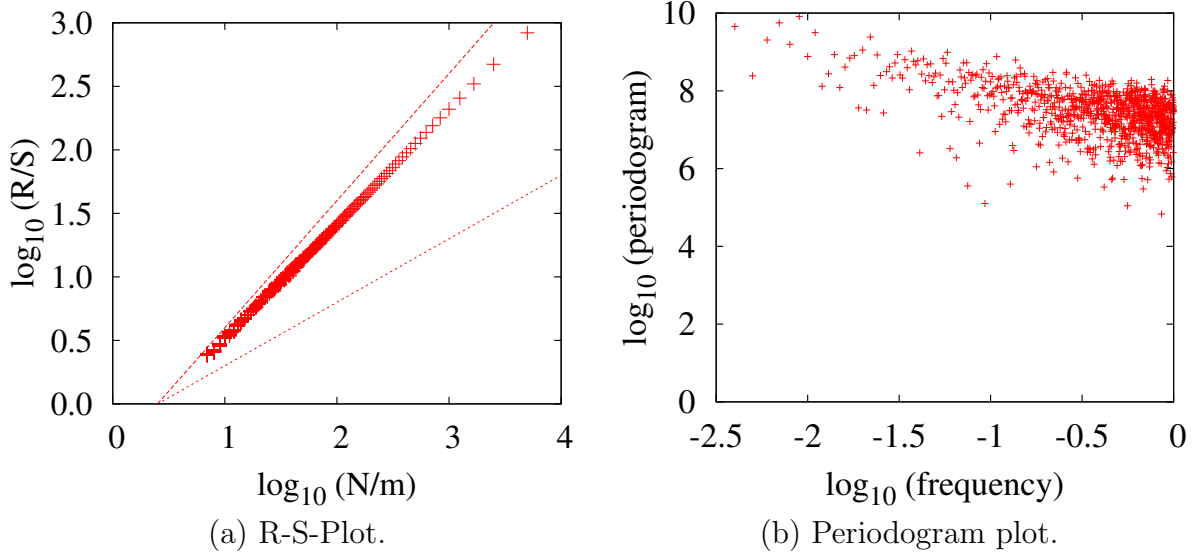


Figure 3.7: Statistical test of self-similar.

cluster arrival is self-similar.

- *Periodogram-based analysis* : The periodogram is an estimate of the spectral density of a given time series. For a self-similar time series, its spectral density obeys a power-law near the origin. Therefore, in a log-log plot of the power spectrum, periodogram should be proportional to the frequency. The *Hurst* parameter H of the time series can be estimated by $\beta = 1 - 2H$, where β is the periodogram slope. Fig.3.7(b) shows periodogram plot of the same trace used in Fig. 3.6. The slope of the regression line is $\beta = -0.75$, yielding an estimate of H as 0.87, which indicates the self-similar nature of the trace.

3.3.3.3 Pareto Model of WiFi White Space

As discussed in Section 3.3.3.2, in a channel shared by a group of 802.11 devices, the arrival process of WiFi frame clusters has the feature of self-similarity. According to [69], the self-similarity is a feature of arrival process with heavy-tailed or power law distributed

inter-arrival time. Since the Pareto process is one of the most widely adopted power law distributions, we chose Pareto model to fit the arrival process of WiFi frame clusters. In the following, we first give the Pareto model and then discuss the goodness-of-fit of it with respect to real WiFi data traces.

We assume the inter-arrival time of frame clusters within time window T fits Pareto model. That is, the distribution of white spaces follows i.i.d Pareto distribution, which satisfies

$$\Pr\{x > t\} = \begin{cases} (\frac{\alpha}{t})^\beta, & t > \alpha \\ 1, & otherwise \end{cases} \quad (3.1)$$

where α and β are the scale and shape of Pareto model respectively. According to the observation in Section 3.3.3.1, we set α to 1 *ms*. In other words, our model only accounts for the inter-cluster space that is longer than 1 *ms*, because shorter white spaces cannot be used by ZigBee links. In Pareto model, β is given by $\frac{\lambda}{\lambda - \alpha}$, where λ is the average inter-arrival time of frame clusters.

We use Kolmogorov-Smirnov Test (K-S test) of 0.95 significance level to evaluate the goodness-of-fit of the Pareto model. K-S test is a widely adopted tool to test the goodness-of-fit. We divide the time into equal sized windows. For each window, a Pareto distribution is fitted by maximum likelihood estimation. K-S test is then applied for each window to test the goodness-of-fit for the estimated Pareto distribution. If a significance level of 0.95 is used, then $0.95k$ out of total k windows should pass the test, if the white space perfectly follows the Pareto distribution. In the Pareto model, we also assume that the inter-arrival times of WiFi frame clusters are independent of each other. To test this assumption, we compute the

one lag autocorrelation for each window. For a time series of n samples generated from an uncorrelated white noise process, the probability that the magnitude of the autocorrelation exceeds $1.96/\sqrt{n}$ is 0.05. Thus we compare the autocorrelation results with $1.96/\sqrt{n}$, and expect that 95% windows will give an autocorrelation value smaller than $1.96/\sqrt{n}$, thus pass the independence test.

Fig. 3.8 gives the results of goodness-of-fit test on the Pareto model. We conduct K-S test on two data traces which are captured in OSDI2006 [12] and SigCOMM2008 [55]. The OSDI and SigCOMM traces includes a group of trace files, which are different in the captured date and time, the monitoring channel, and the position of the traffic sniffer. For each file we check the goodness-of-fit with different window sizes. However, only the results of 100ms and 500ms are shown due to space limitation. Each trace file corresponds to two points in the figure. The x-value of the point is the percentage of windows in the corresponding data trace file that pass the K-S test, when the window size is set to a specific value. The y-value is the percentage of windows that pass the independence test. Thus we expect that, if points are clustered at the top right corner, then the fitness of the Pareto model is good. We observe that the modelability of the frame cluster arrival process varies with time scale. At a small time scale of 100ms, the arrival process can be well characterized by the Pareto model.

3.3.4 Modeling ZigBee Link Performance

In this section, we study the impact of blind WiFi terminals on ZigBee link performance. Based on the Pareto white space model, we will derive the expected frame collision probability. The result will help network designers predict the performance of ZigBee networks in

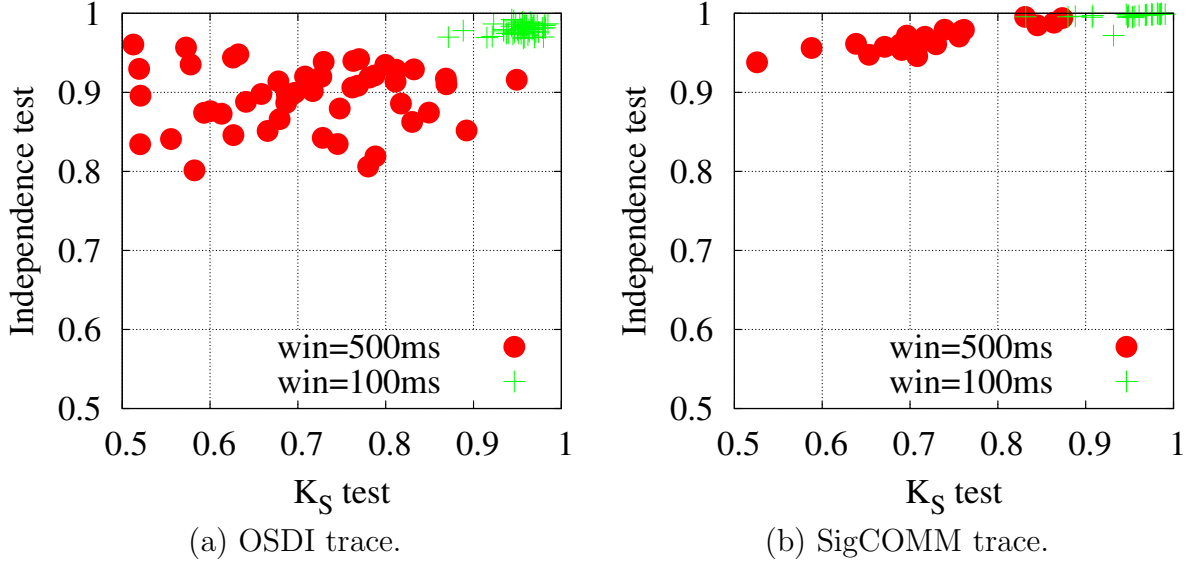


Figure 3.8: Goodness-of-fit tests of Pareto model for real-life WiFi traces.

the presence of WiFi interference. Moreover, it provides foundation for optimizing the link behavior to deal with such interference (Section 5).

According to Section 3.3.1, we define a WiFi blind terminal for a specific ZigBee link as follows: 1) Blind terminal has a carrier-sense based MAC layer, and is located within the CCA range of ZigBee sender, i.e., ZigBee sender will defer if the blind terminal is transmitting; and 2) Blind terminal is located within the interference range of ZigBee receiver, i.e., a ZigBee frame will be colliding with the transmitting frames from the blind terminal. We list the notation used in our analysis in Tab. 3.5. We assume that the channel is shared by a set of k blind terminals $\mathcal{B} = \{\mathcal{B}_i, i = 1 \dots k\}$. The channel condition of a ZigBee sender is modeled by $\langle \alpha, \beta, u, \omega \rangle$, where ω is the percentage of white space, u is the channel utilization ratio of \mathcal{B} . α and β are parameters of the Pareto model given in Eq. (3.1), which characterize the distribution of the white space in the channel.

We now derive the probability of collision between ZigBee and WiFi frames. Our analysis accounts for the ZigBee carrier sensing model and the white space distribution in WiFi traffic.

Table 3.5: Notation used in white space modeling.

τ	packet size of ZigBee.
H	header size of ZigBee.
M	maximum packet size of ZigBee.
D	bit rate of ZigBee.
λ	average white space lifetime.
ω	fraction of channel time that is white space.
u	channel utilization ratio of WiFi.

The main objective of our analysis is to characterize the expected performance of a ZigBee link solely based on the *transmitter's view* of channel condition. The accuracy of our analysis can be easily improved by accounting for the channel condition on the receiver. For instance, a frame may be successfully received by the receiver even when it collides with other frames due to the capture effect [31]. Therefore, we can derive frame delivery ratio based on the probability of frame collision and the signal-to-noise ratio of receiver. However, we argue that such results are not practical because obtaining receiver channel condition requires significant messaging overhead and is not supported by the existing ZigBee MAC layers.

The CSMA of ZigBee will conduct CCA before each transmission and perform exponential backoff if the channel is busy. It will force a frame transmission if the maximum number of CCA tries is reached. A forced transmission will cause the ZigBee frame to collide with the in-air blind terminal frame. In our analysis, we ignore the possibility of forced transmissions because it occurs rarely. Therefore, we will slightly underestimate the overall collision probability. We analyze the following two cases: a) the collision probability when ZigBee transmits a frame in intra-cluster space, denoted by $\mathcal{C}_a(\tau)$, and b) the collision probability when ZigBee transmits a frame in inter-cluster space, denoted by $\mathcal{C}_b(\tau)$, where τ is the ZigBee frame in-air time.

For a randomly arrived ZigBee frame, the probability that it starts to transmit between

two WiFi frames within the same frame cluster is the fraction of frame interval in the total clear channel time. It is given by

$$p_a = \frac{1 - u - \omega}{1 - u} \quad (3.2)$$

As discussed earlier, when the inter-arrival time of frames is shorter than 1 ms, the in-air time of a ZigBee frame of reasonable size will always be longer than the WiFi frame intervals, which will cause a collision between ZigBee and WiFi frames. So we have $\mathcal{C}_a(\tau) = 1$.

We now focus on the case where ZigBee sender transmits a frame in inter-cluster space. For a randomly arrived ZigBee frame, the probability that it starts to transmit in white space is the fraction of white space in the total clear channel time, which is given by

$$p_b = \frac{\omega}{1 - u} \quad (3.3)$$

In this case, the collision probability depends on the *lifetime* and *age* of the white space upon the start of ZigBee transmission, where the *lifetime* is the time interval between two WiFi frame clusters, and the *age* is the time interval between the start of the white space to the start of the ZigBee transmission. Since the distribution of white space age is affected by the backoff process of ZigBee, we now consider two cases: a) ZigBee transmits frame after backoff, and b) ZigBee transmits frame without backoff.

The CSMA of ZigBee will perform backoff if the channel is busy upon the arrival of frame. We assume that the backoff will always align the start of ZigBee transmission with the start of white space. In this case collision occurs only when the white space lifetime is shorter than ZigBee's frame in-air time. Note that here we will underestimate the collision probability since we ignore the actual age of white space when the ZigBee senses a clear

channel. However, the inaccuracy caused by this assumption is not significant due to the short backoff interval of ZigBee MACs. Denote the expected collision probability by $\mathcal{C}_b^0(\tau)$, it satisfies

$$\mathcal{C}_b^0(\tau) > \mathcal{F}\left(\frac{\tau}{D}\right) = 1 - \left(\frac{\alpha}{\tau}\right)^{-\beta} \quad (3.4)$$

where \mathcal{F} is the CDF of Pareto distribution.

When a ZigBee frame arrives within white space, ZigBee will send the frame without backoff. In this case the white space age is uniformly distributed over the entire white space lifetime. Given that the white space lifetime is ℓ , collision occurs if the frame arrives $\ell - \tau$ later than the start of the white space. Its probability is given by $\min\{\frac{\tau}{\ell}, 1\}$. We consider an arrival process of k frame clusters $X = \{X_1, \dots, X_k\}$, and denote the set of white spaces by $L = (\ell_1, \dots, \ell_k)$, where ℓ_i is the time interval between X_{i-1} and X_i . The arrival time of ZigBee frame is uniformly distributed over L . The probability that the ZigBee frame falls into the ℓ_i is given by $\ell_i / \sum_{j=1}^k \ell_j$. Denote the expected collision probability by $\mathcal{C}_b^1(\tau)$, it is given by

$$\mathcal{C}_b^1(\tau, k) = \sum_{i=1}^k \left(\frac{\min(\frac{\tau}{D}, \ell_i)}{\ell_i} \times \frac{\ell_i}{\sum_{j=1}^k \ell_j} \right) \quad (3.5)$$

As the number of frame clusters ranges from 1 to ∞ , we have

$$\mathcal{C}_b^1(\tau) = \frac{\int_0^\infty \min(\frac{\tau}{D}, \ell) f_P(\ell) d\ell}{\int_0^\infty \ell f(\ell) d\ell} = 1 - \frac{1}{\beta} \left(\frac{\alpha D}{\tau} \right)^{\beta-1} \quad (3.6)$$

where $f(\cdot)$ is the PDF of Pareto distribution. The probability that the channel is busy upon the arrival of ZigBee frame is the fraction of WiFi transmission time, which is given by WiFi

channel utilization u . Therefore the expected collision probability of frame transmission in white space is given by $\mathcal{C}_b(\tau) = u\mathcal{C}_b^0(\tau) + (1-u)\mathcal{C}_b^1(\tau)$. According to Eq. (3.4) and Eq. (3.6), we have

$$\mathcal{C}_b(\tau) > 1 - \left(\frac{\alpha D u}{\tau} + \frac{1-u}{\beta} \right) \left(\frac{\alpha D}{\tau} \right)^{\beta-1} \quad (3.7)$$

Note that for Pareto model, $\beta > 1$. Putting all together, the overall expected collision probability $\mathcal{C}(\tau)$ is given by

$$\mathcal{C}(\tau) = p_a \mathcal{C}_a(\tau) + p_b \mathcal{C}_b(\tau) \quad (3.8)$$

Since β is given by $\beta = \frac{\lambda}{\lambda-\alpha}$, and $\frac{\omega}{1-u} < 1$. Taking it into Eq. (3.8), the overall expected collision probability satisfies:

$$\mathcal{C}(\tau) > 1 - \left(1 + \left(\frac{\alpha D}{\tau} - 1 \right) u \right) \left(\frac{\alpha D}{\tau} \right)^{\frac{\alpha}{\lambda-\alpha}} \quad (3.9)$$

3.4 Summary of Chapter

Based on extensive experiment studies, this chapter identified the RET problem and the blind terminal problem in the design of existing link layer protocols. The main contribution of this chapter can be summarized as follows.

- *The rate-adaptive exposed terminal problem.*

Based on the topology of a production enterprise WLAN composed of 23 802.11 a/g APs, we demonstrate the abundance of RETs in real-life WLANs. Specifically, our measurements show that 64.4% of link pairs in our study are RETs of each other. By

optimizing the bit rate of RETs, the link concurrency of WLANs can be boosted by up to 131%. Moreover, our micro-benchmarking results reveal that exploiting RETs introduces new challenges that have not been addressed before. Unlike the classical exposed terminal problem, harnessing RETs improperly may lead to lower system throughput or unfair channel usage among concurrent links.

- *The blind terminal problem.*

We identify the blind terminal problem based on an empirical study of WiFi and ZigBee coexistence. Our measurements show that, although WiFi traffic contains abundant white space, the existing MAC protocols such as CSMA are surprisingly inadequate for exploiting it, resulting in poor performance on ZigBee links in the presence of WiFi blind terminals. Moreover, using real-life WiFi traffic trace, we statistically characterize the interference caused by WiFi blind terminals. Finally, an analytical framework is developed to model the performance of a ZigBee link in the presence of WiFi blind terminals.

Chapter 4

TRACK: Unleashing Link

Concurrency in Enterprise WLANs

4.1 Chapter Introduction

Recent years have witnessed the phenomenal surge of bit rate in wireless communication. For example, the state-of-the-art 802.11n offers 31 bit rates in physical layer (PHY), ranging from 6.5 Mbps to 600 Mbps. Induced by different amount of redundancy in modulation and coding, different bit rates provide different trade-offs between transmission efficiency and reliability. Generally, higher bit rate is more efficient as less redundancy is used, while lower bit rate is more robust against noise and interference.

The increasing diversity of bit rates has enabled great opportunities to further unleash link concurrency in wireless networks. Specifically, links that are considered to be conflicting using fixed bit rates may successfully share the channel without interfering each other, if their bit rates are optimized. However, our extensive experiment studies presented in Chapter 3

have shown that, existing link layer design cannot fully exploit the link concurrency due to their oblivious to the bit rate diversity of underlying PHY, which we referred to as *the rate-adaptive exposed terminal problem*.

Motivated by this observation, this chapter presents TRACK, a novel protocol of *Transmission Rate Adaptation for Colliding linKs*, to harness RETs in enterprise WLANs. Leveraging rate diversity of 802.11, TRACK jointly optimizes the bit rate selection and channel access of concurrent links to improving the aggregate system throughput. TRACK detects the existence of RETs by profiling interfering downlinks based on online channel measurements that account for the effect of frequency selective fading. The bit rate adaptation algorithm of TRACK can optimize different metrics of system performance such as fairness and aggregate throughput. We implement TRACK on commodity 802.11 nodes and evaluate its performance through extensive experiments on a WLAN testbed of 17 nodes. Our results show that, by effectively exploiting link concurrency leveraging bit rate diversity, TRACK improves system throughput by 67% and 35% over DCF and a state-of-the-art protocol, without compromising link fairness

The rest of this chapter is organized as follows. Section 4.2 reviews related work. Section 4.3 and 4.4 discuss the design and implementation of TRACK, respectively. Section 4.5 presents experimental results and Section 4.6 summarizes this chapter.

4.2 Related Work

An effective way of boosting link concurrency in wireless networks is to exploit exposed terminals. For example, CMAP [67] infers conflicting links with packet loss rate passively learned during concurrent transmissions, and opportunistically disables carrier sense when

non-conflicting links are transmitting. CENTAUR [59] periodically measures conflict graph [7], and leverages centralized scheduling to mitigate downlink exposed terminals in enterprise WLANs. However, these systems either assume a uniform bit rate across the network [67], or rely on existing rate control algorithms designed for interference free scenarios [59]. In contrast, we demonstrate the significant impact of bit rate on exposed terminals, and present a practical protocol to improve link concurrency through rate adaptation for interfering links.

Several cross-layer designs have been proposed to improve link concurrency of wireless networks. In [22] [25], successive interference cancelation (SIC) is employed to recover collision packets. However, SIC is available only on software-defined radios and would require substantial modifications to commodity 802.11 receivers, making it difficult for practical deployment. Several recent works [57] [61] study the approach of tuning transmission power to allow concurrent channel access in wireless networks. FLUID [54] exploits flexible channelization to improve system throughput of enterprise WLANs. Power control and channelization are orthogonal to our rate adaptation approach. The TRACK protocol can be integrated with power control and flexible channelization to further improve the spatial reuse of WLANs.

4.3 TRACK Design

We present TRACK, a novel protocol of *Transmission Rate Adaptation for Colliding linKs*. The goal of TRACK is to harness concurrent transmissions of RETs to improve the aggregate system throughput while maintaining satisfactory link fairness. In this section, we first give an overview on TRACK, and then introduce its design in detail.

4.3.1 Overview

TRACK targets enterprise 802.11 a/g WLANs, where APs are densely deployed and connected with a high speed wired LAN. The architecture of TRACK is illustrated in Fig. 4. Exploiting the wired LAN as a messaging channel to APs, TRACK implements a centralized controller to perform admission control for downlinks. Downlinks are admitted to transmit concurrently, if this will improve the aggregate system throughput without compromising link fairness. Recent empirical studies showed that most network traffic in enterprise WLANs is downlink (e.g., 85% shown in [18]). Therefore boosting downlink concurrency will significantly improve overall system performance. In practice, the controller can be deployed on any server that connected with APs through the wired LAN. To reduce the communication overhead, the controller is implemented in the firmware of ethernet NIC to avoid the processing delay caused in network stack. The front end of TRACK is deployed on each AP, which collects and reports channel measurements to the centralized controller. When an AP has packets to send, it submits a transmission request to the controller. Based on the collected channel measurements and the set of currently active links, the controller makes admission decisions. If the link is admitted, the controller notifies the AP to start transmission, and configures the rates of active APs to tolerate the interference caused by concurrent transmissions. If it is rejected, the controller logs the requesting link in a queue, and recomputes admission decisions when one of the active links finishes transmission. We introduce the detail of admission control in Section 4.3.2.

Although the basic design of TRACK is simple, deploying TRACK in practice is challenging due to the following reasons. First, in TRACK, APs and the centralized controller communicate through the wired LAN, which induces delay that affects the timing efficiency

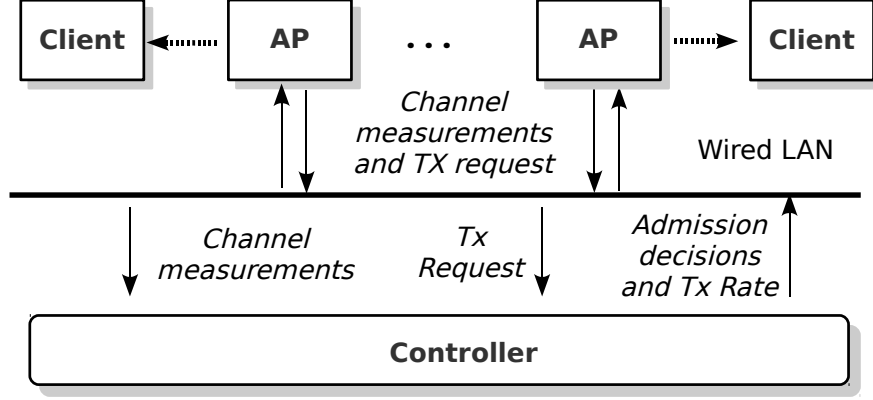


Figure 4.1: The architecture of TRACK.

of link admission decisions. To amortize this overhead, TRACK uses a *packet batching* mode which groups downlink packets into blocks for transmission (see Section 4.3.3). Second, conventional SINR usually performs poorly in predicting link performance due to effects of frequency selective fading [26]. To address this issue, TRACK adopts a novel metric called *effective SINR* to improve the accuracy of interference estimation (see Section 4.3.4). Third, although TRACK exploits the wired LAN as a messaging channel to coordinate the transmissions of APs, it is difficult to control clients and other non-enterprise WLAN devices. To address the conflicts between downlinks and non-scheduled uplinks, TRACK employs a novel approach called *selective CCA*, which allows a TRACK AP to detect and ignore the signal of scheduled downlinks in clear channel assessment, while preserving the CSMA-based contention between scheduled and other non-scheduled transmissions (see Section 4.3.5). In the following, we will introduce the design of TRACK in detail.

4.3.2 Link Admission Control

To improve link concurrency, TRACK opportunistically harnesses RETs by admitting downlinks if they will improve the aggregate system throughput without compromising link fair-

ness. In the following, we first formulate the problem of RET admission control, and then introduce the protocol design.

We assume that a *packet batching* mode [4] [2] is employed by the WLAN, where packets are combined into block for transmission. We discuss the motivation and design of packet batching in Section 4.3.3. Link admission decision is made before the transmission of each block. Rate is controlled on a per-packet basis. Let \mathbf{L} be the set of active links in the network, and l the new link that has a batch of packets to send. The RET admission control problem can be formulated as follows. For a set of active links \mathbf{L} , a new link is allowed to transmit concurrently with the links in \mathbf{L} , if and only if (1) $\mathcal{T}(\{l\} \cup \mathbf{L}) > \mathcal{T}(\mathbf{L})$, and (2) $\mathcal{J}(\{l\} \cup \mathbf{L}) \geq \alpha$, where $\mathcal{T}(\cdot)$ and $\mathcal{J}(\cdot)$ are the functions of aggregate effective throughput and link fairness under the *optimal* rate assignment. α is a pre-defined threshold on link fairness. The two constraints assure that concurrent transmissions of RETs will not lead to system performance degradation or link unfairness, as demonstrated in benchmark 2 and 3 in Section 3.2.3.

We define the optimal bit rate for link i , denoted by γ_i^{opt} , as the maximum rate that assures reliable packet delivery. In our implementation, we consider a link to be reliable if its packet reception ratio (PRR) is higher than 95%. Formally, given a set of active links \mathbf{L} , γ_i^{opt} can be derived as,

$$\gamma_i^{\text{opt}} = \max_{\gamma \in \mathbf{R}} \{PRR(\text{sinr}_i, \gamma) \geq 95\%\} \quad (4.1)$$

where \mathbf{R} is the set of legitimate bit rates defined in 802.11 standard; $g(\cdot)$ is the model of PRR, which depends on the employed bit rate γ and the SINR measured at the receiver of link. The SINR can be computed as follows,

$$\text{SINR} = \frac{RSS_{l_0}}{n_r + \sum_{i>0} RSS_{l_i}} \quad (4.2)$$

where n_i is the noise floor measured at the receiver of link i ; rss_i^j is the average signal strength of packets transmitted from the sender of link j to the receiver of link i . In practice, $g(\cdot)$ can be profiled offline, while rss_i^j and n_i should be measured at runtime to estimate SINR. We discuss accurate SINR estimation in Section 4.3.4.

Assuming that all links in \mathbf{L} use the optimal rates computed by Eq. (4.1) for concurrent transmissions, the aggregate effective throughput can be computed as follows,

$$\mathcal{T}(\mathbf{L}) = \sum_i \frac{d}{h_{PHY}/\gamma_0 + (h_{MAC} + d)/\gamma_i^{\text{opt}}} \quad (4.3)$$

where d is the size of payload; h_{PHY} and h_{MAC} are the sizes of PHY and MAC headers, respectively. In 802.11, the PHY header is always transmitted with the lowest rate, denoted by γ_0 . Since the optimal rate γ_i^{opt} is selected such that the resulted PRR is higher than 95%, we neglect the impact of packet loss on effective throughput.

We model the link fairness as follows. For each active link, we compute its channel utilization as $u_i = \gamma_i^{\text{opt}}/c_i$, where c_i is the channel capacity when there is no interference. In practice, c_i can be estimated by the optimal bit rate used in an interference-free channel. In this work, we use Jain's fairness index [28] to quantify the fairness of channel usages, although other fair measures can also be adopted. Formally, the fairness is computed by,

$$\mathcal{J}(\mathbf{L}) = \frac{(\sum_{i=1}^k u_i)^2}{k \times \sum_{i=1}^k u_i^2} \quad (4.4)$$

where k is the number of active links in \mathbf{L} .

```

1 repeat
2    $update\_needed \leftarrow False$ ;
3   if transmission request received from downlink  $q$  then
4     if  $Q = \phi$  then
5        $update\_needed \leftarrow True$ ;
6     end
7      $Q \leftarrow Q + \{q\}$ ;
8   end
9   if downlink  $q'$  finishes transmission then
10    if  $Q \neq \phi$  then
11       $update\_needed \leftarrow True$ ;
12    end
13     $L \leftarrow L - \{q'\}$ ;
14  end
15  for  $i \leftarrow 1$  to  $|Q|$  do
16    Compute optimal rates for  $\{q_i\} + L$ ;
17    if  $\mathcal{T}(\{q_i\} + L) > \mathcal{T}(L)$  and  $\mathcal{J}(\{q_i\} + L) \geq \alpha$  then
18       $L \leftarrow L + \{q_i\}$ ;
19      Remove  $q_i$  from  $Q$ ;
20    end
21    else
22      break;
23    end
24  end
25  Send updated bit rates to links in  $L$ ;
26 until stop;

```

Figure 4.2: Downlink admission control of TRACK.

When receiving a request of packet batch transmission from AP, the controller logs the information of requested downlink in a *waiting queue*, denoted by $\mathbf{Q} = \{q_1, \dots, q_n\}$, where q_i is sorted in descending order based on their waiting time. Let \mathbf{L} be the set of active downlinks. The pseudocode of downlink admission control algorithm is described in Fig. 4.2. Specifically, the controller makes admission decisions when (1) a request is received atop the waiting queue, or (2) a downlink finishes its transmission of packet batch. The controller attempts to accept transmission requests in a FIFO manner. It checks the throughput and fairness constraints from the top of the waiting queue. The process stops when the check fails at one of waiting downlinks. Then the controller notifies the APs of active downlinks to update their bit rates. To avoid corrupting ongoing packets of other concurrent downlinks before their rates are updated, the AP of newly admitted downlink should stagger its transmission by a packet air time. The time complexity of Fig. 4.2 is $O(n)$, where n is the number of requesting downlinks.

As the link admission algorithm accepts links in a FIFO manner, it may not yield the maximum set of concurrent downlinks. However, the FIFO-based admission policy assures that no downlink will be starved when there is consistent contention from other APs. Moreover, compared with computing the optimal transmission schedule of all active links, Fig. 4.2 has significantly lower computation cost. This is especially important for the enterprise WLANs with heavy traffic load. Lastly, although Fig. 4.2 is adopted in our implementation, TRACK can integrate other scheduling algorithms.

4.3.3 Packet Batching

TRACK uses *packet batching* to amortize the communication overhead between APs and the centralized controller. Specifically, it aggregates multiple packets into a block for transmission. Channel access decision is made before the transmission of each packet block. However, bit rate is controlled on a per-packet basis. TRACK employs a block ACK scheme and packet ACK is disabled during packet batch. When the transmission of a block finishes, the client replies the AP with a vector, where each bit indicates the reception of one packet. Similar with the per-packet ACK defined in 802.11, block ACK is transmitted using lower rate to mitigate the inefficiency caused by ACK loss. In our implementation, block ACK is transmitted with a rate of 6 Mbps.

We note that packet batching mode adopted by TRACK is similar with the TXOP operation defined in 802.11e [4], which groups packets for transmission to amortize the overhead of per-packet channel contention. Although the packet batching operation can be extended to be compliant with 802.11e, we leave this for future work. In our implementation, we set the duration of packet batch to 4 *ms*, which is equal to the maximum TXOP period defined in 802.11e.

4.3.4 Interference Estimation

TRACK employs SINR as the metric for quantifying interference caused by concurrent transmissions, and then maps SINR to PRR for rate selection (see Eq. (4.1) and Eq. (4.2)). However, conventional SINR often performs poorly in predicting PRR due to effects of frequency selective fading [26], where different sub-carriers of the channel suffer different degrees of fading due to multipath propagation, causing variation of delivery performance across sub-

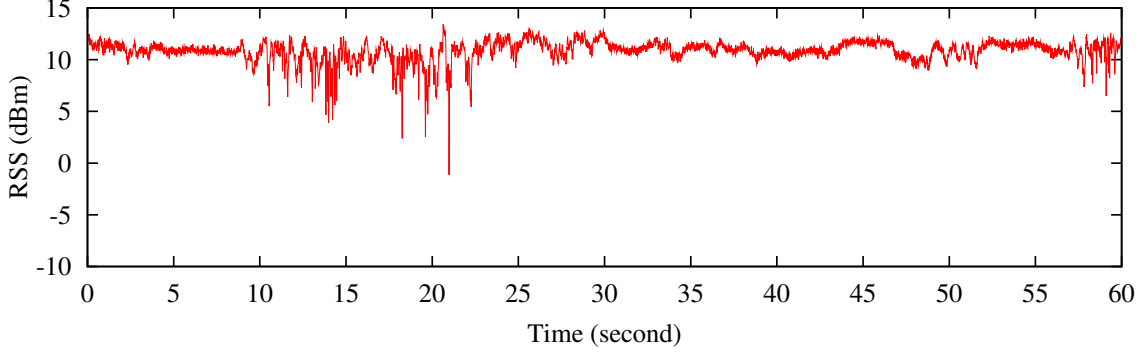


Figure 4.3: The subcarrier with least variation in our testbed.

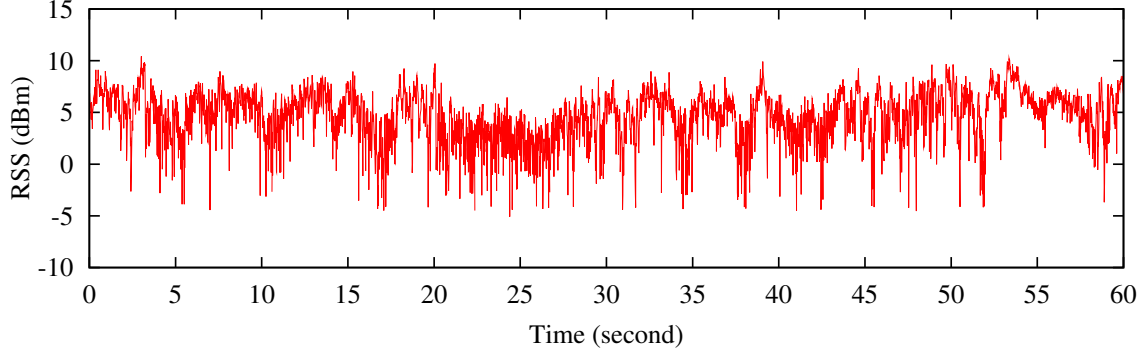


Figure 4.4: The subcarrier of most variation in our testbed.

carriers. Ignoring this effect, conventional SINR is not predictive of real link performance, as the same SINR value may correspond to different PRRs on the links that experience different fading conditions.

TRACK mitigates the effect of frequency selective fading by extending the effective SNR model proposed in [26] to account for the effect of interference. The controller collects channel state information (CSI) measured on downlinks, and estimates sub-carrier SINRs for predicting uncoded bit error rate (BER) using the theoretical model. The *uncoded* BER is then averaged over sub-carriers and mapped back to obtain *effective SINR*. Different from the conventional SINR that simply averages SINR over sub-carriers, the effective SINR is calculated by averaging delivery performance across subcarriers to account for frequency selective fading. We refer to [26] for the detail of model derivation.

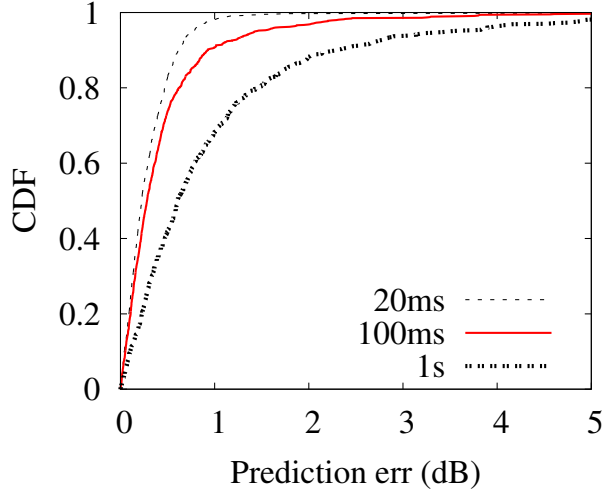


Figure 4.5: Prediction error of different measurement period.

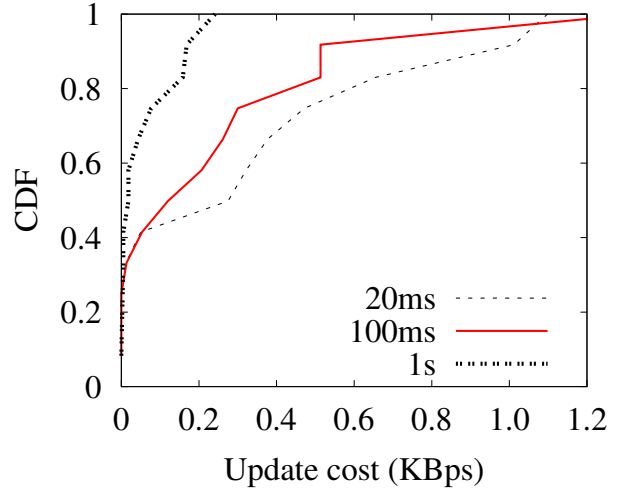


Figure 4.6: Update overhead of reporting CSI to the controller.

The downlink CSI is measured as follows. First, AP periodically pings client, and uses the response packet to extract CSI. Then the channel reciprocity theory is applied to derive the downlink channel model. The measured CSI is reported to the controller if the signal variation on any subcarrier exceeds a threshold since last update. We observe that a threshold of 2 dB on subcarrier signal is enough to assure the accuracy of effective SINR estimation. We set the measurement period based on an empirical approach. We deploy 12 links to mimic the topology of a production WLAN, and then transmit back-to-back packets at a frequency of 500Hz to probe the CSI. Each experiment lasts for 10 minutes. The subcarriers with least and most variations in our testbed are shown in Fig. 4.3 and Fig. 4.4, respectively. Fig 4.5 illustrate the prediction error when different measurement periods are used. We observe that smaller period results in higher prediction accuracy. Specifically, a measurement period of 100ms is enough to limit the prediction error below 1 dB with a probability around 90%. Fig 4.6 shows the per-link update overhead incurred by reporting CSI to the controller. The result shows that the overhead is lower than 0.6 KBps per link in more than 90% cases. In our implementation, we will use a measurement period of 100ms. We study the performance

of effective SINR driven rate selection in Section 4.5.3.

4.3.5 Coexistence with Non-Scheduled Traffic

Leveraging the architecture of enterprise WLAN to harness downlink RETs, TRACK uses a centralized controller to adapt the rates of concurrent downlinks. However, it is difficult for TRACK to control the uplink traffic as clients do not run TRACK. Moreover, TRACK may be deployed in the vicinity of other non-TRACK WLANs. As a result, TRACK must be able to coexist with unscheduled traffic. A case of coexistence is given in Fig. 4.7. To avoid interfering unscheduled packet transmissions, we devise a novel clear channel assessment algorithm called *selective CCA* to handle non-TRACK traffic. The selective CCA allows a TRACK AP to detect and ignore the signal of scheduled concurrent downlinks in channel assessment, while preserving the CSMA-based contention between TRACK downlinks and unscheduled links. In the following we discuss the design and implementation of selective CCA in detail.

Suppose there is a scheduled concurrent downlink set of $\mathbf{L} = \{l_0, \dots, l_m\}$. At the sender of downlink i , the measured signal strength of sender j is rss_i^j dBm. To allow concurrent transmission of \mathbf{L} , the selective CCA sets the CCA threshold as follows

$$t_i = 10 \log_{10} \left(\sum_{k \neq i}^n 10^{rss_i^k / 10} \right) \quad (4.5)$$

The rationale behind Eq. (4.5) is to set the CCA threshold by the total signal power of scheduled downlinks. When there is unscheduled traffic, the measured signal power at TRACK APs will exceed the power threshold given in Eq. (4.5), causing selective CCA to detect a busy channel. To update the CCA threshold, TRACK APs periodically record the

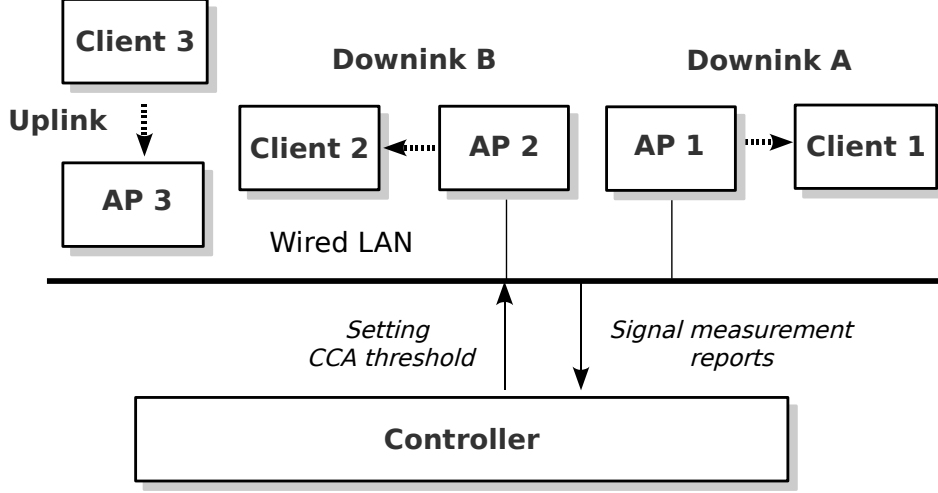
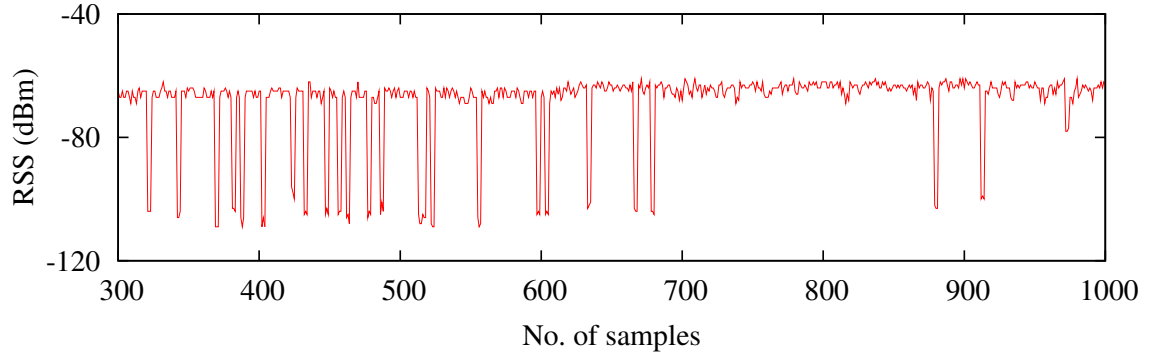


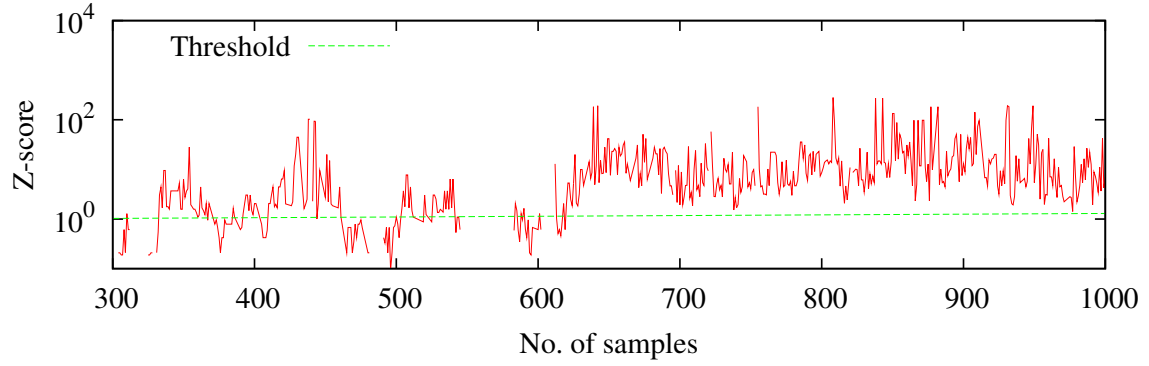
Figure 4.7: A case study of coexistence of scheduled concurrent downlinks (A and B) with unscheduled uplink transmission. In this case, downlink A and the uplink are out of carrier sense range of each other, while B and the uplink can hear each other. The objective of selective CCA is to disable carrier sense among scheduled downlinks, while preserving the channel contention between downlinks and the uplink. In this case, the transmitter of B (AP 2) is required to detect the signal of unscheduled transmitter (client 3) before accessing the channel, under the interference of scheduled on-going transmissions (from AP 1 to client 1). In our deployment, the measured signal strength from AP 1 to AP 2 is 15 dB stronger than the link from client 3 to AP 2.

signal strength of overheard AP beacons. When TRACK configures bit rates through wired LAN, it also distributes the updated threshold, calculated by Eq. (4.5).

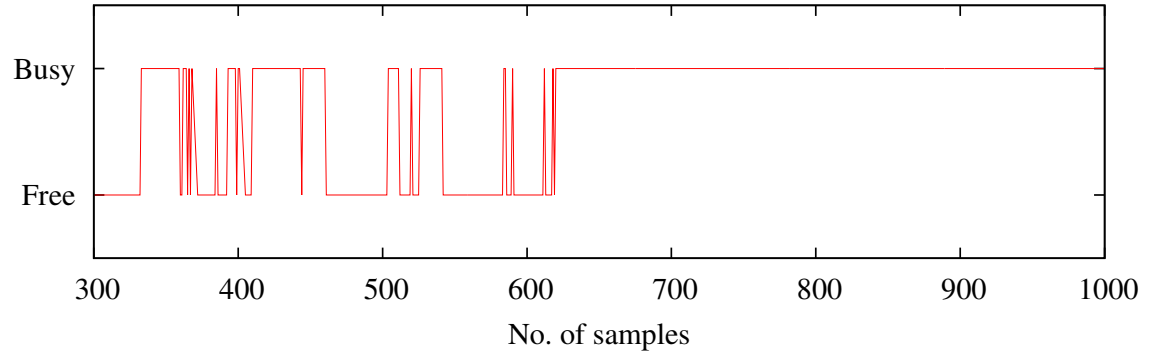
We implemented the selective CCA algorithm in ath9k [1] on Atheros AR928x NICs. We disable the default CCA module before performing selective CCA, and sample the RSSI register for signal strength measurements. The RSSI register on AR928x reports instant SNR measurement. We subtract this value by the reading of noise floor register to get the energy level of channel. We note that the signal measurements are not reliable when RSSI register is sampled in packet gaps (as shown in Fig.4.8(a)). Therefore we apply a filter to remove these samples. A potential problem of selective CCA is that it may suffer lower carrier sensing sensitivity when the signal of unscheduled transmission is much weaker than scheduled links.



(a) RSS trace measured at AP 2.



(b) Z-score trace computed at AP 2.



(c) Selective CCA output at AP 2.

Figure 4.8: Trace of selective CCA at AP 2 of the link deployment in Fig. 4.7. Selective CCA samples the received signal strength at 250KHz. The size of CCA window is set to 5. A z-score threshold of 90% confidence is used. Uplink starts transmission around 600 samples. AP 2 senses a free channel with high probability when AP 1 is transmitting. After the uplink becomes active, AP 2 detects busy channel reliably.

Moreover, the signal power measurement often experiences temporal variations. The above two factors make it difficult to detect incoming weak uplink signals among strong in-air downlink signals. We draw on statistical testing techniques to address this problem. When performing channel assessment, the AP first assumes a clear channel, and then performs the *z-test* on a window of collected signal samples to test the hypothesis. Suppose that the AP has a group of n measured signal samples which has mean μ and standard deviation σ . The AP computes the *z-score* by $z = \frac{\mu - thresh}{\sigma/n}$. The channel is deemed busy if the *z-score* rejects the hypothesis at a given confidence level. Fig. 4.8 shows the working trace of selective CCA on AP 2 of the link deployment given in Fig. 4.7. We observe that AP 2 reliably detects the existence of uplink signal in presence of scheduled concurrent transmissions, as *z-test* is sensitive to the statistical variation of received signal strength.

4.4 TRACK Implementation

We have implemented TRACK in mac80211 [5] with ath9k [1]. In this section, we present the implementation of TRACK. To assure the practical deployability, we emphasize the principle of minimizing the modification of WLAN clients.

To adapt rate for concurrent transmissions, TRACK plugs effective SINR into an offline profiled SINR-PRR model to select the maximum rate that achieves reliable packet delivery. An important issue in profiling the SINR-PRR model is to limit the effect of multipath propagation in model profiling. We carefully place sender, receiver and jammer such that both of the data and interference links have line-of-sight connections. We tune the transmission powers on sender and jammer to vary the SINR on receiver, and record PRR for each of 100 probes. The model profiled for Atheros-based cards of 12 Mbps is shown in Fig. 4.9.

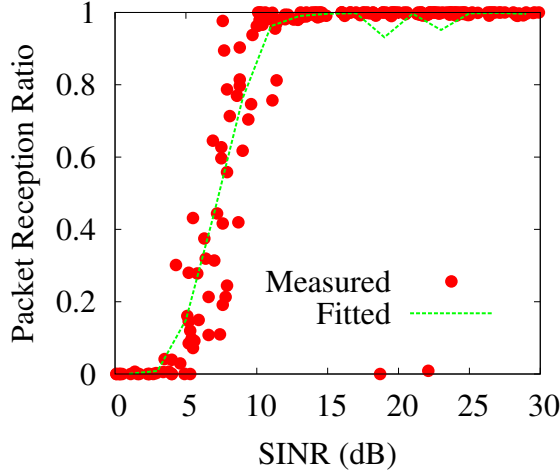


Figure 4.9: The SINR-PRR model profiled using 12 Mbps rate.

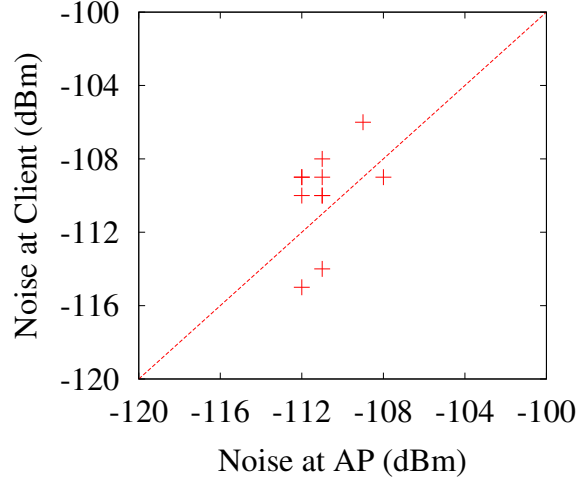


Figure 4.10: Noise floor measured at clients and their associated APs.

For computing the effective SINR, TRACK requires channel measurements of downlinks on sub-carrier level to account for frequency selective fading. Ideally, this can be done by using channel sounding to extract channel state information (CSI) from received uplink packets, and then applying the channel reciprocal theory to derive downlink channel model [26]. Unfortunately, the Atheros cards in our implementation do not provide open source firmware for supporting channel sounding function. To address this issue, we leverage the spectrum sampling utilizes provided by Atheros NICs [1] to measure link qualities on subcarriers. Specifically, the AP first pings client to get response packet, and then switches to spectrum analyzer mode to quickly scan the received signal strength on subcarriers. We note that the measurements obtained by this method is only an approximation of the real CSI. However, our results show that it works efficiently in practice. We evaluate the performance of model-driven rate selection in Section 4.5.3.

Estimating effective SINR requires measuring noise floor at receiver. However, it is difficult to get clients' noise floor without deploying any code on them. We use the noise floor

measured at the associated AP to approximate the noise floor of client. This approximation is reasonable as APs are usually densely deployed to ensure the coverage of enterprise WLAN networks, and hence the clients and their associated AP are often in proximity. We validate this assumption by measuring the noise floor in a 17-node testbed, including 6 APs and 11 clients (Section 4.5). The noise floor measured for all 11 AP-client pairs are shown in Fig. 4.10. We observe that clients and their associated APs share similar noise floor.

4.5 EVALUATION

In this section, we study the performance of TRACK through extensive experiments. We evaluate TRACK against two baseline protocols to show the advantages of harnessing RETs. In the following, we will first introduce the experiment setting and then discuss evaluation results in detail.

4.5.1 Experiment Setting

We deploy a testbed consisting of 6 APs and 11 clients within an office building spread over 1,800 square feet. Each node is a laptop equipped with Atheros AR928x radio, running mac80211 driver [5] with ath9k [1]. To ensure the realism of our testbed deployment, we place our APs close to the APs of a production WLAN deployed in the same area, and then randomly place clients around them. Our measurements show that a total of 178 links in the testbed can achieve a packet delivery ratio higher than 95% under at least one bit rate. Among all the link pairs, 23.8% are exposed terminals, and 8.4% are hidden terminals. To further validate our deployment, we measure and compare the downlink quality of our testbed with the production WLAN in Fig. 4.11. We observe that signal strength of our

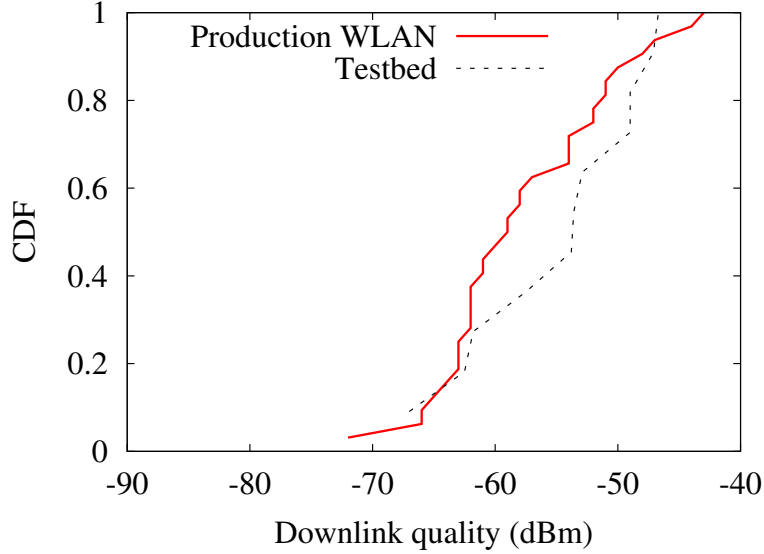


Figure 4.11: Testbed link quality vs production WLAN.

testbed downlinks is slightly stronger, indicating that the part of deployment we emulated is denser than the production WLAN.

To demonstrate the efficiency of TRACK, we compare its performance with two baseline protocols, including 802.11 DCF (the default CSMA algorithm of 802.11) and a centralized scheduling algorithm for harnessing exposed terminals, which we refer to as *HET*. Different with TRACK, HET does not perform rate adaptation for concurrent transmissions. Instead, it uses the measured downlink channel model to estimate effective SINR and predicts resulted PRR for concurrent downlinks under default bit rate, i.e., the rate learned in interference-free channels. HET admits downlinks for concurrent transmission if all downlinks can achieve reliable packet delivery. Therefore HET only exploits ET link pairs. We note that HET is similar to existing solutions designed to address the classical ET problem [59] [67]. For DCF and HET, we employ the algorithm proposed in [26] for rate control. For TRACK, the centralized controller estimates effective SINR, and plugs it into an offline profiled SINR-PRR model for rate selection.

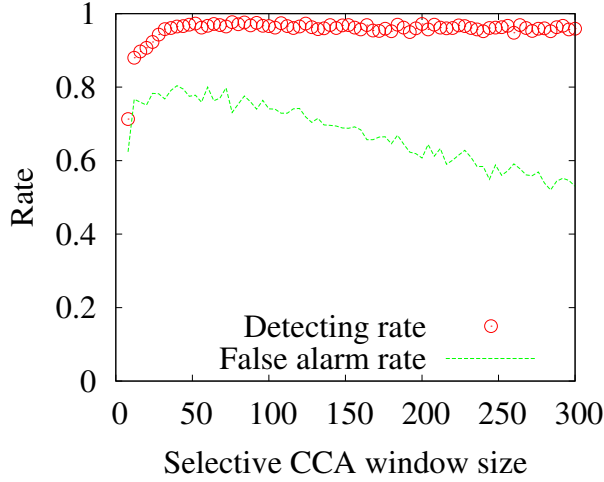


Figure 4.12: CCA sensitivity.

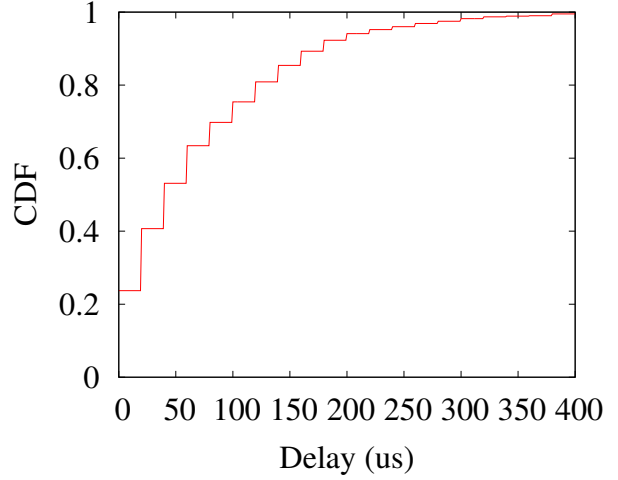


Figure 4.13: CCA Delay.

4.5.2 Selective CCA

Before large-scale evaluation of system performance, we study the efficiency for several key components of TRACK. TRACK performs selective CCA to avoid interference between downlinks and unscheduled transmissions. We evaluate the sensitivity of selective CCA in Fig. 4.12. We run the algorithm proposed in Section 4.3.5 on a monitor to detect the existence of unscheduled traffic under the interference of a scheduled link. The monitor samples the RSSI register at a frequency of 250KHz. A z-score threshold of 90% confidence is used. We first measure the signal strength of scheduled transmitter for setting the CCA power threshold at the monitor, and then start scheduled and unscheduled transmission subsequently. Our experiment is conducted on a deployment where the signal of scheduled transmission is 14dB stronger than the unscheduled transmitter. We observe that when the size of CCA window is larger than 5, selective CCA reliably detects unscheduled transmission with a probability higher than 90%. We also observe that the false alarm rate is around 60%. We note that false alarm will cause additional transmission delay in a channel without unscheduled links. However, as shown in Fig. 4.13, the cost is acceptable. Specifically, when

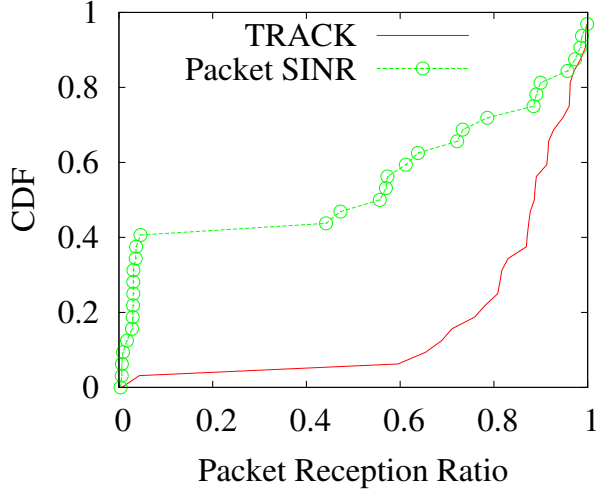


Figure 4.14: CDF of PRR for TRACK and packet SINR based rate selection.

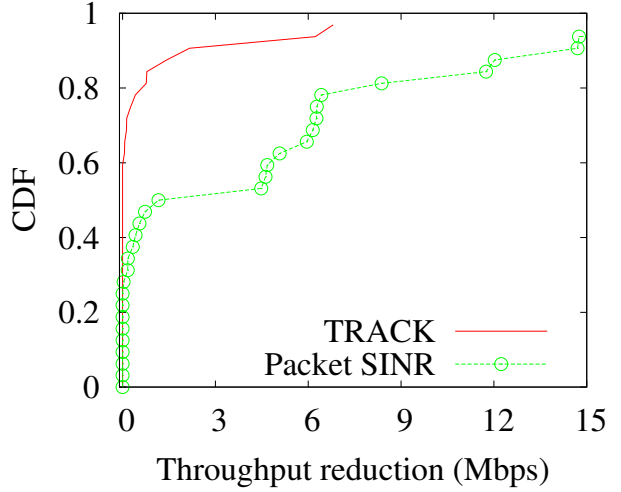


Figure 4.15: CDF of throughput reduction caused by sub-optimal rate selection.

the size of CCA window is 5, which is around $20\mu s$ under 250KHz sampling frequency, the average transmission delay caused by false alarm is only $92\mu s$. In the following experiments, we use a z-score threshold with 90% confidence, along with a CCA window of size 5.

4.5.3 Performance of Rate Selection

In this section, we study the performance of the model driven rate adaptation algorithm, which takes effective SINR as input for handling frequency selective fading. We evaluate the proposed method against packet SINR driven rate selector, which is based on a conventional approach that directly maps on-line measured SINR to a bit rate by looking up the profiled SINR-PRR models. However, the SINR-PRR curves often suffer from spatial and temporal variations caused by the effect of frequency selective fading [26]. Therefore packet SINR method usually performs poor in reality.

We conduct experiments by randomly selecting sender and receiver in our testbed to form the data link, and then place a node around the link for serving as interferer. The sender first

collects channel measurements with a probing process of 10ms. For the conventional rate selector, these measurements are used to compute packet SINR, while the proposed algorithm derives effective SINR to feed rate selector. Then we load 1000 broadcast probes on data link to measure packet delivery for each bit rate. The rate which results in best observed throughput is selected as optimal rate, which serves as the reference for evaluating packet SINR and the proposed approach. The experiment is repeated for 32 randomly selected settings.

Fig. 4.14 shows the PRR measured for the two algorithms. We observe that both of the methods experience packet loss due to sub-optimal rate selection. However, the proposed algorithm achieves satisfiable PRR under interference, performing much better than packet SINR based approach. Specifically, the proposed algorithm delivers more 80% of packets in more than 70% of cases, while the packet SINR based approach loses half of the packets with probability higher than 50%. We further evaluate the impact of sub-optimal rate selection in Fig. 4.15 by measuring throughput reduction compared with the optimal rate. We observe that the proposed method does not decrease throughput in 60% cases, implying the accurate selection of optimal rate. Even when a sub-optimal rate is selected, the caused throughput reduction is always less than 6 Mbps, which is much better than the results of packet SINR based algorithm.

4.5.4 Performance on Two-AP topologies

In this section, we evaluate the performance of TRACK on two-AP topologies. We conduct the experiments by randomly picking two downlinks that are within the carrier sense range of each other, and then measure throughput for each link using broadcast traffic in a period

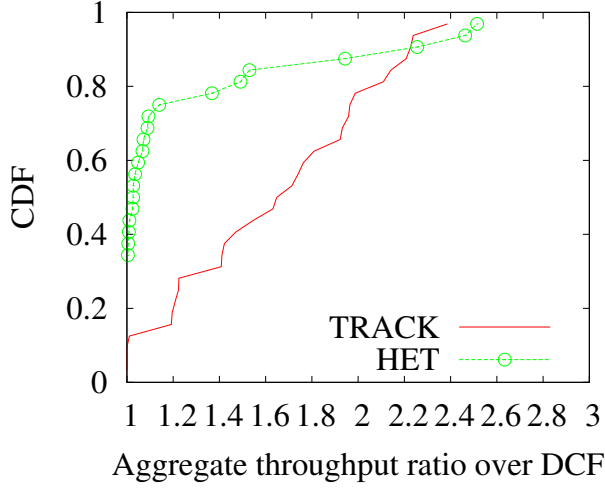


Figure 4.16: Aggregate throughput on 2-AP topology.

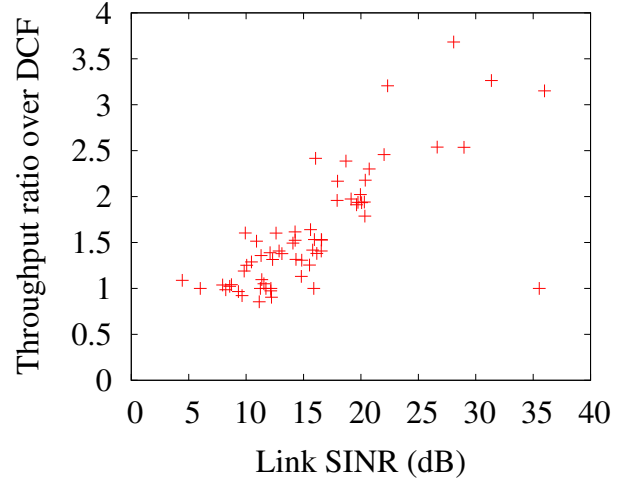


Figure 4.17: Throughput improvement vs interference.

of 1 minutes. The experiment are repeated for 32 randomly chosen settings.

Fig.4.16 compares TRACK with HET in terms of the aggregate throughput improvement ratio of DCF. We observe that TRACK performs better than HET as it harnesses RETs by rate adaptation, while HET only exploits ETs. Specifically, compared with DCF, TRACK doubles the throughput with a probability higher than 65%, while HET only results in better performance in less than 60% cases. We further study the impact of interference on TRACK performance in Fig. 4.17. We find that TRACK performs better on links of strong signal. In particular, the throughput improvement ratio over DCF can be higher than 2.5x for links of SINR higher than 25dB. The reasons are two-fold. First, strong links that support higher bit rates usually suffer more from the overhead of channel contentions and backoff procedure. Therefore they are more sensitive to the impact of enabling concurrent transmissions, which mitigates the MAC layer overhead. Second, links of high quality usually perform poor when coexist with links of lower rates. As each time when the slow link wins in channel contention, it takes more time to transmit a packet than faster links, causing the rate anomaly problem [65]. Allowing concurrent transmission by rate adaptation, TRACK

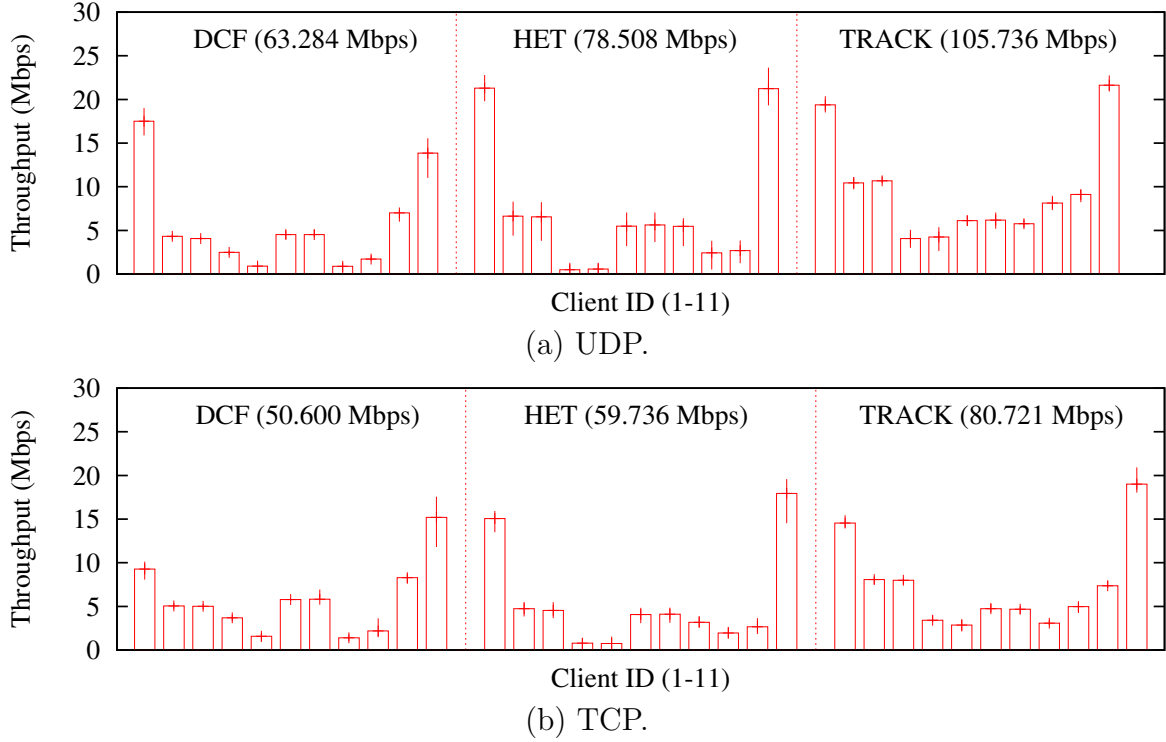


Figure 4.18: Throughput analysis of 11 clients. The aggregate system throughput is shown after the name of each protocol.

avoids this problem, thereby enhancing the performance for faster links.

We also note that the performance reported in this section is better than the measurements results in Section 3.2. The reason is that our testbed has a slightly higher density than the production WLAN, as compared in Fig. 4.11. Therefore there are more fast links in network, which are favored by TRACK. The results also suggest that TRACK may perform better on dense 802.11n WLAN, which supports higher rates than 802.11a/g. However, we left the extension as future work.

4.5.5 Throughput, Fairness and Overhead

We conduct large-scale evaluation on our testbed to study the performance of TRACK with both of UDP and TCP traffics. We first evaluate the downlink throughput of TRACK

against DCF, and HET. We load saturated traffic on all downlinks. Since TRACK only schedules downlink transmissions, the results measured on downlink-only scenario gives the maximum performance gain that could be achieved by harnessing RETs. To mitigate the impact of short-term channel variation, we perform the experiment by switching between the four protocols in a round-robin fashion. Each protocol runs for one minute in each round. The experiment lasts for 2 hours. For the TRACK and HET, we set the period of transmission burst to 4ms, as discussed in Section 4.3.3.

Fig. 4.18(a) plots the UDP throughput of the 11 clients under the three protocols. The result shows that DCF suffers serious link unfairness. This is due to the unfairness nature of CSMA, where the links having more contenders are penalized due to the high channel utilization around the senders. Compared with DCF, HET achieves a 1.24x aggregate throughput improvement by exploiting exposed terminals. However, HET exacerbates the throughput unfairness among links, since some links get more transmission opportunities as exposed terminals are allowed to transmit concurrently, which hinders the channel access of their contenders. TRACK performs best among the four protocols. Specifically, by exploiting RETs, TRACK achieves a 1.67x throughput over DCF, and improves the aggregate throughput by 35% over HET. Fig.4.18(b) gives the evaluation results in the scenario of TCP traffic. We found that the result is similar. Specifically, TRACK improves the system throughput by 59% over DCF, and 35% over HET.

We further compare the throughput improvement ratio over DCF achieved by TRACK and HET for individual links. The result is given in Fig. 4.19 and Fig. 4.20. We observe that TRACK achieves better throughput on most of the links. In particular, when compared with HET, TRACK boosts throughput by up to 6x, and achieves an improvement ratio of more

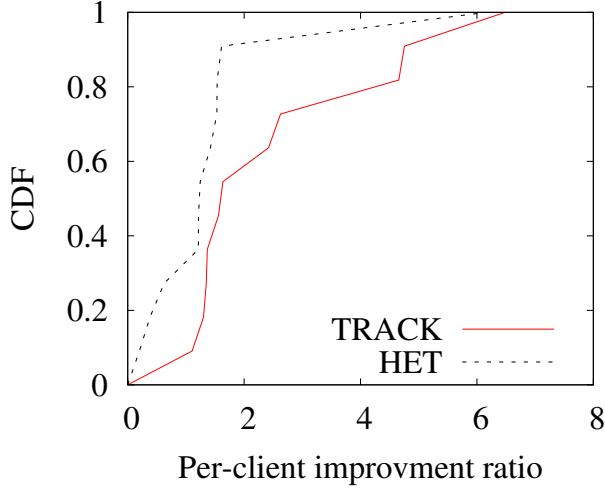


Figure 4.19: Per-link UDP throughput

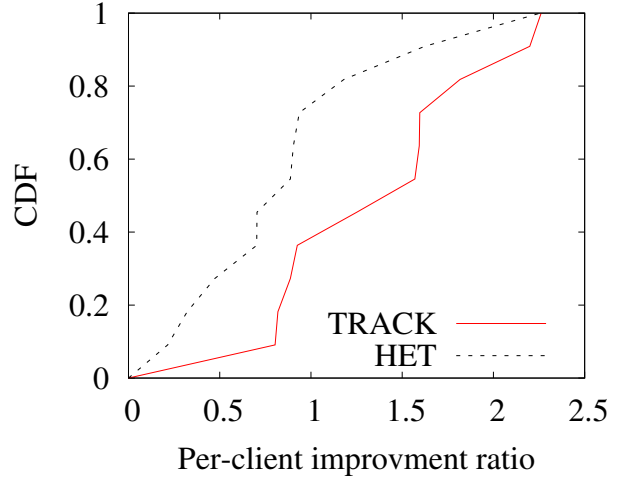


Figure 4.20: Per-link TCP throughput.

Table 4.1: Per-client traffic load vs throughput fairness.

Traffic load	UDP		TCP	
	DCF	TRACK	DCF	TRACK
2 Mbps	0.970	0.987	0.972	0.987
6 Mbps	0.873	0.996	0.861	0.994
10 Mbps	0.781	0.937	0.768	0.916
14 Mbps	0.727	0.869	0.706	0.840
16 Mbps	0.698	0.792	0.664	0.729

than 50% on 54.5% of the links; while only 9.1% links experience a throughput decrease, with highest drop less than 11.9%. When compared with DCF, 45.4% of the links enjoy an improvement higher than 100%; 4.5% of the links have a drop, with maximum decreasing less than 15.5%.

As discussed in Section 3.2.3, exploiting RET may result in link unfairness. To show that TRACK improves the overall throughput without unfairly exploiting individual links, we measure the Jain's fairness for the throughputs of individual links, and compare it with DCF in Table 4.1. The result shows that for both TRACK and DCF, the throughput fairness drops as traffic load increases, since heavy traffic load will intensify channel contentions, thus exacerbates the unfairness problem of CSMA. However, we observe that TRACK con-

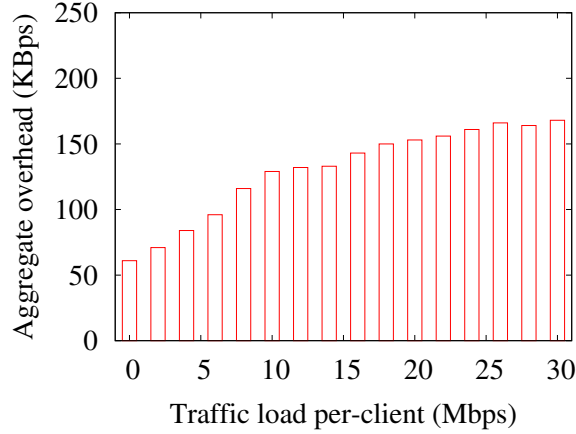


Figure 4.21: Overhead.

sistently performs better than DCF, as it exploits rate adaptation to allow more transmission opportunities on starved links.

We evaluate the overhead of TRACK by measuring the load of control messages over the wired LAN, including AP packets of reporting channel measurements, transmission requests, and TRACK controller commands for configuring bit rates and selective CCA threshold. We show the result of UDP traffic in Fig. 4.21. The TCP result is similar. We observe that overhead increases with per-link traffic load as more traffic incurs frequent interactions between controller and APs. However, the aggregate traffic load is still lower than 200KBps even when the network becomes saturated.

4.5.6 Impact of Non-scheduled Traffic

Table 4.2: Impact of non-scheduled traffic on throughput ratio.

Traffic load ratio (uplink:dnlink)	HET/DCF		TRACK/DCF	
	Dnlink	Uplink	Dnlink	Uplink
0.100	1.236x	1.047x	1.571x	1.168x
0.250	1.189x	1.036x	1.544x	1.153x
0.600	1.199x	1.039x	1.550x	1.009x
0.800	1.198x	1.030x	1.509x	1.048x
1.000	1.171x	1.003x	1.276x	1.121x

We conduct experiments to study the impact of unscheduled traffic. We repeat the experiment introduced in Section 4.5.5 with different uplink traffic loads, and compare TRACK with HET in terms of the UDP throughput improvement ratio over DCF. Table.4.2 shows that TRACK performs consistently better than HET as uplink traffic load grows. For instance, when the ratio between uplink and downlink traffic load is 25%, which is the general case in enterprise WLANs [18], HET improves the downlink throughput by 18.9%, while TRACK achieves a ratio of 1.54x, while maintaining similar uplink throughput. The results demonstrate the advantage of exploiting RETs, as well as the efficiency of TRACK to avoid interference on unscheduled traffics.

4.5.7 Impact of Topology

To evaluate the impact of topology, we change the placement of testbed clients and adjust the antenna orientation of APs to create hidden terminal intensive and exposed terminal intensive topologies. Then we repeat the experiment introduced in Section 4.5.5 to measure the aggregate UDP throughput, and compare TRACK with HET in Tab. 4.4 and Tab. 4.3, respectively. The results show that although HET improves the median throughput, it is outperformed by DCF at the 10th percentile in all topologies. Meanwhile, the median throughput gain decreases as the ratio of ET drops. This is because HET intensifies the link starvation, where some of the links obtain significantly more transmission opportunities due to the exploitation of exposed terminal, hindering the transmissions of contending links. Moreover, the results show that the performance gain of HET depends on the ratio of ET in the topology, as it is blind to the underlying concurrency opportunities provided by 802.11 rate diversity. Specifically, HET boosts the median throughput by 46.4% in ET intensive

Table 4.3: The UDP throughput improvement ratio of HET nad TRACK measured on three topologies. Each topology is characterized by the percentages of ET and HT among all link pairs.

Ratio of ET	Ratio of HT	10th percentile	Median	90th percentile
0.052	0.184	0.842x	1.101x	1.485x
0.238	0.084	0.533x	1.209x	1.484x
0.420	0.050	0.091x	1.464x	1.922x

Table 4.4: The UDP throughput improvement ratio of TRACK measured on three topologies.

Ratio of ET	Ratio of HT	10th percentile	Median	90th percentile
0.052	0.184	26.789x	1.537x	1.227x
0.238	0.084	16.367x	1.670x	1.215x
0.420	0.050	2.605x	2.011x	1.647x

topologies, while only improves 10.1% in HT intensive topologies. In contrast, TRACK improves the 10th percentile over DCF in all topologies, as it allows starved links to transmit with coordinated rate adaptation among concurrent links. Moreover, it is interesting to notice that, even in a topology with only 5.2% exposed link pairs, TRACK still improves the aggregate throughput by 53.7% over DCF. The results demonstrate the ability of TRACK to exploit RETs. In particular, TRACK allows non-ET link pairs to transmit concurrently with coordinated rate adaptation, therefore unleashing the concurrency opportunities in enterprise WLANs.

4.6 Summary of Chapter

In this chapter, we present *TRACK*, a novel protocol that harness rate-adaptive exposed terminals exploiting the bit rate diversity of enterprise WLANs. We implement TRACK on commodity 802.11 nodes and evaluate its performance through extensive experiments on a WLAN testbed of 17 nodes. Our results show that TRACK improves system throughput

by up to 67% and 35% over DCF and a conventional approach of harnessing ETs, without compromising link fairness.

Chapter 5

WISE: Exploiting WiFi White Space for ZigBee Performance Assurance

5.1 Chapter Introduction

Wireless coexistence is becoming a growing issue due to the proliferation of wireless devices in the crowded unlicensed 2.4GHz band. This issue is particular critical to low-power wireless devices, such as ZigBee-based ECG sensors, as they must compete for the limited spectrum resources with WiFi devices that have much higher transmission power. Although this issue can be partially addressed by assigning orthogonal channels to ZigBee and WiFi, such a solution is often infeasible as the 2.4 GHz band is populated by ever increasing number of WiFi devices.

Our experiment studies presented in Section 3.3 have identified *the blind terminal* problem as the major cause of poor ZigBee performance in co-existing environment. Specifically, we showed that WiFi devices are often blind terminals to ZigBee nodes because existing link

layer protocols fail to work due to the heterogeneous PHY and power asymmetry in co-existing environment.

Motivated by this observation, this chapter tackles the blind terminal problem with a practical solution. Specifically, we present the design, implementation and evaluation of WISE, a WhItE Space-aware framE adaptation protocol for ZigBee devices to achieve assured performance under the heavy interference of WiFi blind terminals. WISE predicts the length of white space in WiFi traffic based on the Pareto model proposed in Section 3.3, and intelligently adapts frame size to maximize the throughput efficiency while achieving assured packet delivery ratio. We implement of WISE in TinyOS 2.0 on TelosB mote equipped with 802.15.4 compliant CC2420 radio. We evaluate the performance of WISE under the interference of controlled WiFi traffic. Our results demonstrate significant advantages of WISE over B-MAC - the default MAC protocol of TinyOS that is compliant with the standard MAC layer of 802.15.4, and OppTx - a state-of-the-art protocol designed to utilize opportune conditions of bursty links. We show that under heavy interference of WiFi blind terminals, WISE achieves performance gains of 4x and 2x over B-MAC and OppTx, while only incurs 19.5% and 42.5% of their overhead.

The rest of this chapter is organized as follows. Section 5.2 reviews related work. Section 5.3 presents the design of WISE. We offer experimental results in Section 5.5 and conclude the chapter in Section 5.6.

5.2 Related Work

The coexistence of heterogeneous devices is a critical issue in unlicensed ISM bands. In [27], Adaptive Frequency Hopping (AFH) is proposed for Bluetooth and WiFi coexistence. AFH

is further improved in [21] by sensing and predicting the WiFi behavior using the model proposed in [20]. However, these approaches are designed for frequency hopping systems, and require the support of cognitive radios for spectrum sensing.

Several recent studies have been conducted to mitigate the bursty interference on low power 802.15.4 links. Srinivasan et al. [63] proposed an opportunistic transmission (OppTx) protocol to improve the performance of bursty 802.15.4 links. OppTx measures and quantifies the correlations in packet delivery and loss, and use them to set transmission backoff delay. However, OppTx is oblivious to the probabilistic feature of white space, and hence cannot explicitly utilize the white space in WiFi channel.

Several protocols have been proposed to address the coexistence between ZigBee and WiFi in the open radio spectrum. Zhang et al. [73] proposed Cooperative Busy Tone (CBT), which allows a ZigBee node to schedule a busy tone concurrently with the desired transmission, thereby improving the visibility of ZigBee devices to WiFi. However, CBT requires modifications of ZigBee's PHY layer and cannot be implemented on commercial ZigBee devices. Wang et al. [68] developed WiCop which broadcasts fake WiFi preamble headers to mute WiFi interferers when ZigBee devices are transmitting. However, WiCop requires the cooperation of WiFi devices, which poses a major management challenge for the existing WiFi deployments. Liang et al. [38] proposed BuzzBuzz to mitigate WiFi interference through multi-header design and forward error correction. However, performing error corrections on low-power ZigBee devices usually yields large processing delays of tens of milliseconds. Moreover, BuzzBuzz incurs additional overhead due to the uses of header and payload redundancy.

Several error detection and recovery methods [71] [19] [29] [17] are proposed to utilize

partial packets to improve the link reliability. These approaches typically work at the MAC layer. The frame control protocol proposed in this paper operates transparently to the MAC layer, hence can be integrated with these approaches. In our earlier work [74], system called ZiFi was developed to utilize ZigBee radio to detect the existence of WiFi hotspots based on the unique interference signatures of WiFi. However, mitigating the interference of WiFi for ZigBee devices is not addressed.

5.3 WISE: White Space-Aware Frame Adaptation

In this section, we propose a novel frame control protocol called *White Space-aware frameE adaptation* (WISE) to tackle the blind terminal problem for ZigBee networks. WISE predicts the length of white space in WiFi traffic based on the Pareto model, and intelligently adapts frame size to maximize the throughput efficiency.

5.3.1 Overview of WISE

The design objective of WISE is to maximize the throughput efficiency of ZigBee while bounding the packet collision probability under user requirement. WISE consists of the following two components that reside between PHY and MAC layers. The *white space modeling* component builds the Pareto model based on maximum likelihood estimation. The *frame adaptation* component computes the size of frame that maximizes the throughput efficiency while limiting the collision probability within the user given bound.

When WISE gets a frame from the MAC layer, it may split the frame into *sub-frames* and the size of each sub-frame is determined by predicting the remaining lifetime of the white space using the Pareto model. WISE maintains a *session* for transmitting all sub-frames of

a MAC frame. Each sub-frame carries a session ID and delimiters. This is necessary because a) the white space may be too short to accommodate the transmission of entire frame when the channel is heavily loaded with WiFi traffic; and b) the integrity of the MAC frame needs to be protected, so that the receiver can process the frame correctly. Such a design allows WISE to operate transparently to the MAC layer and the modification to the MAC layer is kept the minimum. The receiver will assemble all sub-frames within the same session into a integral MAC frame and pass to the MAC layer. In the following, we discuss the optimization of sub-frame size and the design of WISE.

5.3.2 Optimizing Sub-Frame Size

As blind WiFi terminals cannot sense the signal of ZigBee, collisions will occur if a ZigBee frame cannot finish its transmission before the arrival of the next WiFi frame cluster. Therefore, to reduce collision probability, the transmission time of ZigBee should be shorter than the remaining lifetime of the current WiFi white space. Let ρ be the white space age when a frame is ready for transmission. As the lifetime of white space follows the Pareto model defined in Eq. (3.1), we have the following conditional collision probability $\mathcal{C}(\tau, \rho)$ for a given frame size τ

$$\mathcal{C}(\tau, \rho) = \Pr\{t < \rho + \frac{\tau}{D} \mid \rho\} = 1 - (\frac{\rho}{\frac{\tau}{D} + \rho})^\beta \quad (5.1)$$

where D is the channel rate of ZigBee, α and β are the scale and shape of the Pareto model of white space.

The goal of frame adaptation of WISE is to maximize the efficiency of transmission while limiting the collision probability under user requirement. Since the size of protocol header

is fixed, the transmission efficiency is a monotonic increasing function of the sub-frame size. Given a specific collision probability threshold T , the optimization problem of WISE frame adaptation can be formulated as follows:

$$\text{Maximize} \quad \tau \quad (5.2)$$

$$\text{Subject to} \quad \mathcal{C}(\tau, \rho) < T \quad (5.3)$$

$$\tau \leq M \quad (5.4)$$

where M is the maximum frame size of ZigBee. Solving the problem, we obtain the optimal sub-frame size:

$$\tau = \text{Min} \{ \rho \times \gamma, M \} \quad (5.5)$$

where γ is given by

$$\gamma = D \times \left((1 - T)^{-\frac{\lambda - \alpha}{\lambda}} - 1 \right) \quad (5.6)$$

where λ is the mean in the Pareto model of the white space.

5.3.3 Frame Session Management

When a frame is passed to WISE by the MAC protocol, WISE computes the sizes of sub-frames and then starts a session to transmit them. The receiver maintains the states of sub-frames in a session in order to keep the integrity of the original MAC frame. Each sub-frame is composed of a 1-byte WISE header and payload. The WISE header includes 1-bit

start session delimiter, 1-bit *end session delimiter* and 6-bit *session ID*. We now discuss how a session is managed by WISE in details.

1. *Session ID assignment.*

Each sub-frame in a session carries the same session ID. The session ID is assigned by the sender by randomly generating a number between 0 and 63, as WISE header uses 6-bit ID to identify each session. Note that the receiver identifies whether it is the destination of a sub-frame solely by the session ID, because a sub-frame is only part of MAC frame and hence may not include the MAC address. Two sessions on different nodes within the communication range of each other may accidentally choose the same session ID and start at the same time. However, such a possibility is low and its impact on the performance of WISE is negligible.

2. *Session initialization and sub-frame transmission*

WISE sender initiates the session by transmitting a *session registration frame* (SRF), which is identified by setting 1 in the *start session delimiter* bit. The SRF must protect the integrity of the MAC layer header of the frame, so that the receiver can conduct frame pre-processing correctly. Due to the criticality of SRF, we carefully control the collision probability of SRF as follows. When the frame size derived from Eq. (5.5) is smaller than the sum of the PHY header, WISE header and MAC header, the transmission will be deferred by a random backoff. The sender will repeat this process until it can transmit the entire MAC layer header within one sub-frame.

When a SRF is received, WISE receiver will conduct MAC layer pre-processing on the MAC header, such as address recognition etc. If the receiver is the destination, it

records the session ID in a *session table* and allocates buffer for the session. At the same time, a session lifetime timer is initiated. The session will be forcibly terminated upon the timeout, so that the receiver will not wait too long when the last sub-frame of the session is lost. In this case, the received partial packet will be assembled and submitted to the MAC layer.

3. *Sub-frames transmission and assembling*

Given a collision probability bound, WISE sender uses Eq. 5.5 to estimate the maximum frame size which can be transmitted in the white space. As discussed earlier, if the model predicts that the remaining lifetime of white space cannot afford the transmission of the protocol header within the collision probability bound, then the sender will defer the transmission for a random backoff. For the last sub-frame in the buffer, WISE sender will set the session end delimiter for identification. WISE receivers uses the session table to identify all sub-frames that destined to itself. Received sub-frames will be buffered until the last sub-frame is received, or the session lifetime timer fired. Then all sub-frames within the same session will be assembled into one integral frame, and submitted to the MAC layer.

5.3.4 Sub-frame Retransmission

WISE employs a sub-frame retransmission algorithm to further improve the reliability and throughput of ZigBee. For this purpose, each sub-frame is appended with a one-byte checksum. Whenever a sub-frame is received, WISE receiver checks whether it contains errors or not. After receiving the last sub-frame of a frame session, the WISE receiver sends an ACK to the WISE sender, indicating which parts of the frame should be retransmitted.

To reduce the protocol cost of encoding the size and sequence number fields of sub-frame, WISE divides the frame into 8-byte blocks¹. Each sub-frame consists of several blocks. WISE appends a *block map* for each sub-frame, where the number of ones in the block map denotes the number of contained blocks, and the positions of ones indicates the positions of blocks in the frame. Note that the maximum frame size of ZigBee is 128 bytes, which requires only two bytes to encode the block map. When the last sub-frame is received, the WISE receiver embeds a block map in the ACK packet, where the positions of ones indicates the successfully received blocks. The WISE sender will retransmit the entire frame if the last sub-frame or the ACK packet is lost.

Due to the use of sub-frame checksum and block map, sub-frame retransmission of WISE may increase the protocol overhead. We expect that this additional cost will be paid off by the improved packet delivery performance. By only retransmitting the corrupted blocks, sub-frame retransmission reduces the overhead of recovering a erroneous frame, increasing the probability of successful recovery. We will evaluate the impact of sub-frame retransmission in Section 5.5.5.

5.4 Implementation

WISE is implemented in TinyOS on TelosB motes equipped with 802.15.4 compliant CC2420 radios. In the following, we discuss the details of the implementation.

¹The time it takes a ZigBee radio to transmit a 8-byte block is around 256 us.

5.4.1 White space sampling

We implement the channel modeling algorithm in the driver of CC2420. CC2420 radio exposes CCA and start of frame delimiter (SFD) pins to the micro-controller. When a signal above the CCA threshold is detected, the CCA pin goes low to indicate the busy channel, otherwise it goes high. Whenever there is a change of the pin state, a signal is triggered to interrupt the micro-controller. The SFD pin indicates the start of a decodable packet. It interrupts the micro-controller when a SFD (0xA7 in 802.15.4) is detected. WISE treats all undecodable signals as blind terminal interference. We note that an undecodable signal may be attributed to a 802.15.4 interferer. We will discuss how to deal with this issue in Section 5.4.4. To sample the white space, WISE captures all the interruptions on CCA and SFD pins. Whenever the CCA pin goes low but the SFD pin remains unchanged, an arrival of undecodable signal is detected.

5.4.2 White space modeling

WISE periodically samples the channel and measures the interval between two undecodable signals in order to build the white space model. According to Section 3.3.3, if the length of the interval is longer than 1 ms, it is considered as a sample of white space. WISE keeps a moving window for collected samples, and uses the maximum likelihood estimation to derive a Pareto model. The size of the moving window is set to 100 ms as it is shown in Section 3.3.3 that the distribution of inter-frame spaces fits the Pareto model only if the time scale is shorter than 100 ms. The channel sampling frequency is a tunable parameter. In our experiments, we observe that a maximum sampling frequency of 200 Hz is high enough to ensure the accuracy of Pareto model. Since the sampling window is 100 ms, at most

20 samples need to be stored. We will evaluate the impact of the sampling frequency on modeling accuracy in Section 5.5.

5.4.3 Sub-frame Adaptation

To reduce the computational overhead, we adopt a discrete approach to optimizing the size of WISE sub-frames. According to Eq. (5.5), for a given collision probability bound, the optimal size of sub-frame depends on ρ and γ , where ρ is the white space age upon the start of channel assessment, and γ is a function of the scale of Pareto model α and the average lifetime of the white space λ . Since we set α to 1 ms, γ only depends on λ . For the purpose of computational efficiency, we discretize the time into slots of 1 ms. For a given collision bound, γ is calculated offline for each integer value of average white space lifetime λ . The results are stored in γ -table, which can be looked up online by λ . In our experiments, we observed that the impact of blind terminal on ZigBee performance is neglectable when the average lifetime of the white space is longer than 20 ms. Therefore, the storage cost of γ -table is small.

5.4.4 Discussion

As discussed earlier, WISE derives the white space Pareto model by sampling intervals between undecodable signals. However, the undecodable signal may be attributed to a 802.15.4 source that is located within the CCA range, but outside the communication range. Therefore, the 802.15.4 signal may introduce errors in the estimation of Pareto model that is originally derived for 802.11 traffic. We now discuss two solutions to this issue. However, evaluating these solutions is left for future work.

- *WiFi detection using PHY features.*

A ZigBee node may detect WiFi signals by capturing specific PHY signatures of the signal. As a result, WISE will only use samples of detected WiFi signals to build the white space model. This approach is feasible if the PHY specification of WiFi is known to the ZigBee detector. In [20], a feature based approach to 802.11 signal detection is proposed to search the preamble and SFD field of 802.11 packets. Feature-based signal detection has high accuracy, but may require nontrivial support from the radio hardware.

- *WiFi detection using MAC features.*

The MAC features of 802.11 and 802.15.4 are significantly different. The difference can be utilized by ZigBee radio to distinguish 802.11 and undecodable 802.15.4 signals. For example, the frame transmission times of 802.11 and 802.15.4 follow distinct distributions, due to the significant differences in channel rate and frame size. In addition, the interval between two back-to-back 802.11g frames is much shorter than that of 802.15.4. This is because the minimum contention window of 802.11g is 32, with a time unit of $9\ \mu\text{s}$ while it is 8 for 802.15.4 with a time unit of $320\ \mu\text{s}$. Therefore, captured signals are attributed to 802.11 source with higher probability, if the intervals between them are shorter than $32 \times 9 = 288\ \mu\text{s}$.

5.5 Evaluation

In this section, we present the evaluation results of WISE. We implemented WISE in TinyOS 2.0 on TelosB motes equipped with 802.15.4 radios. We employ B-MAC [49] (without low

power listening), which is the default MAC in TinyOS, for medium access control. B-MAC is designed based on CSMA and is compliant with the standard MAC layer of 802.15.4. Our implementation of WISE did not require any change in B-MAC's implementation. Unless otherwise indicated, the transmit power is set to -7 dBm, which assures good delivery performance for ZigBee links in our setup. We use netbooks equipped with 802.11 compliant Intel Atheros 928x NICs as WiFi interferers.

We first evaluate the performance of WISE using controlled WiFi traffic. For this purpose, D-ITG [3] is used to generate WiFi traffic at different rates. As a high-fidelity Internet traffic generator, D-ITG is capable of generating simultaneous flows from different protocols. Empirical results showed that D-ITG can reproduce realistic traffic patterns under a wide range of network settings [3]. Then we evaluate the impact of real-life WiFi traffic on ZigBee performance. To generate interference, a WiFi-enabled laptop is employed to replay the packet traces collected from real-life WiFi deployment. During trace replaying, the in-air time of each packet and the intervals between packets are kept to be consistent with the original trace. In this way the trace replayer is able to accurately reproduce the packet arrival process of real-life WiFi traffic. Finally, we turned on the sub-frame retransmission protocol introduced in Section 5.3.4, and evaluate its impact on WISE performance.

Our evaluation focuses on three performance metrics: frame delivery ratio, throughput, and throughput overhead. Throughput is measured as the total number of bytes of *impactive* payloads delivered in one second. Throughput overhead is defined as $\frac{N_t - N_d}{N_d}$, where N_t is the total number of bytes transmitted per second, and N_d is the throughput. Thus the throughput overhead quantifies the additional bytes transmitted by the sender to deliver one byte of impactive payload. We compare WISE to two baseline protocols: 1) B-MAC

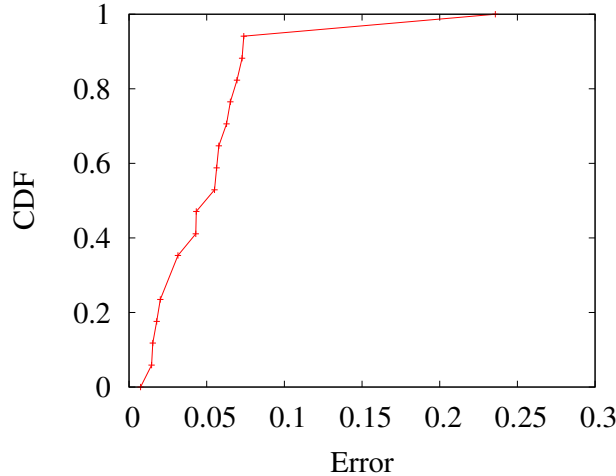


Figure 5.1: Evaluation of model.

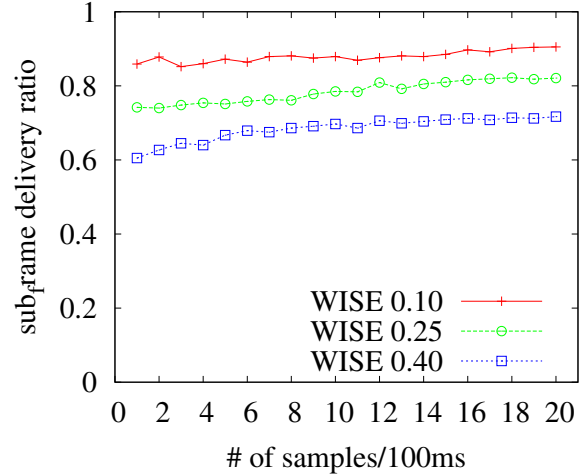


Figure 5.2: Sampling frequency.

without WISE and 2) the opportune transmission (OppTx) protocol proposed in [63]. OppTx is a state-of-the-art low-power sensor network protocol designed to mitigate the impact of interference. It significantly improves the throughput of bursty links by transmitting back-to-back packets and controlling the backoff delay when a failure occurs [63]. For a fair comparison, the backoff delay of OppTx is always tuned to the optimal value. In contrast to WISE, OppTx is oblivious to WiFi traffic and hence cannot explicitly utilize white space in WiFi channels.

5.5.1 Accuracy of the Performance Model

In this section, we study the accuracy of the performance model proposed in Section 3.3.4. We deploy two TelosB motes in an indoor environment and ensure that the PRR of the ZigBee link is above 95% without WiFi interference. To introduce blind terminal interference, we intentionally place a WiFi interferer close to the ZigBee sender and receiver. The average interference powers on ZigBee sender and receiver are -60 dBm and -66 dBm, respectively. The WiFi node generates traffic of UDP and TCP flows. We vary the traffic rate of WiFi

to evaluate the impact on frame delivery ratio of ZigBee link. To measure the channel utilization of WiFi, the ZigBee sender samples the CCA pin at a frequency of 100Hz. The experiment is conducted for 10 mins. The channel utilization of WiFi is given by the portion of number of ‘0’ samples, which indicates a busy channel. The results in Fig. 5.1 show that our model matches the experiment results closely. In particular, 90% of the errors are smaller than 0.1.

5.5.2 Impact of Sampling Frequency

WISE needs to periodically sample the channel for deriving the Pareto model of white space, which may pose considerable overhead for low-power 802.15.4 devices. We now study the impact of sampling frequency. The experimental setting is the same as in Section 5.5.1. The WiFi node generates traffic of UDP and TCP flows at 2.3 Mbps. The sampling frequency of WISE is varied from 1 to 20 samples/100ms while collision probability bound is varied from 0.1 to 0.4.

Fig. 5.2 shows the impact of sampling frequency on the frame delivery ratio of WISE. Since WISE uses the white space model to control the collision probability, a higher frame delivery ratio implies a better modeling accuracy. We observe that the link performance meets the given collision bounds, and the frame delivery ratio grows with the sampling frequency. However, sampling frequency only shows small impact on frame delivery ratio. For instance, when sampling frequency is increased from 1 to 20 samples/100 ms, the frame delivery ratio of WISE (under 0.4 collision bound) only changes from 0.605 to 0.717 with a difference of 0.112. Under collision bound of 0.1, the difference is only 0.046. This result implies that WISE can achieve high modeling accuracy with extremely low sampling overhead.

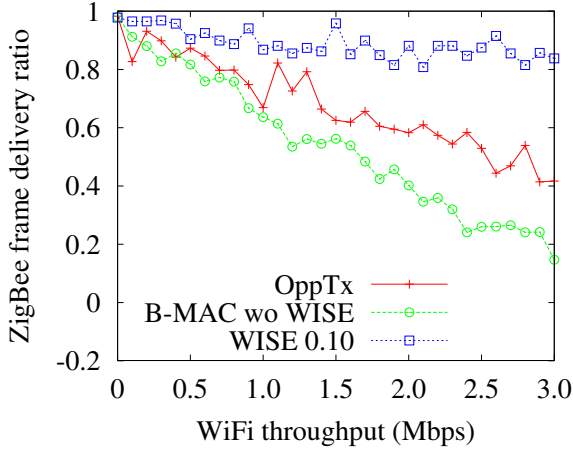


Figure 5.3: Broadcast frame delivery ratio.

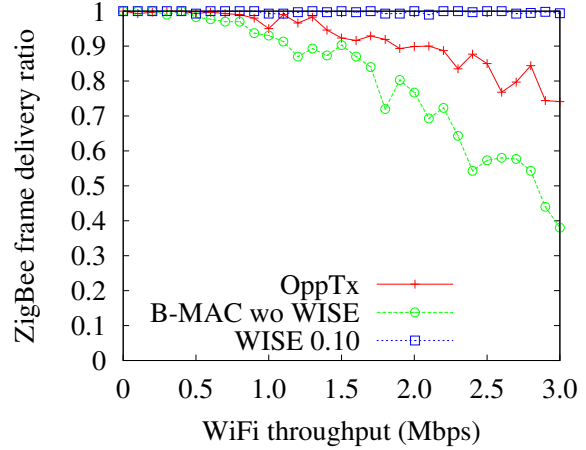


Figure 5.4: Unicast Frame delivery ratio.

This feature is particularly desirable for ZigBee devices due to their resource limitation.

5.5.3 Performance Comparison

We now compare the performance of WISE to that of B-MAC and OppTx under different levels of WiFi interference. The traffic rate of ZigBee is 4Kbps. Fig. 5.3 shows the frame delivery ratio for broadcast traffic. We observe that the frame delivery ratio of B-MAC and OppTX drops linearly with the increase of WiFi data rate, while WISE significantly outperforms these two protocols. Since WISE uses the white space model to control the collision probability of each transmission, the delivery performance remains relatively stable despite the increasing data rate of WiFi. We observe that when WiFi data rate is 3Mbps, the performance gains of WISE over B-MAC and OppTx are 4x and 2x, respectively.

We now evaluate the performance of WISE for unicast traffic. The maximum retransmission tries is set to 3, which is the default setting in 802.15.4. Fig. 5.4 shows the impact of WiFi data rate on the frame delivery ratio of ZigBee link. The result is similar as the case of broadcast traffic. Despite the high traffic load of WiFi blind terminal, WISE with collision

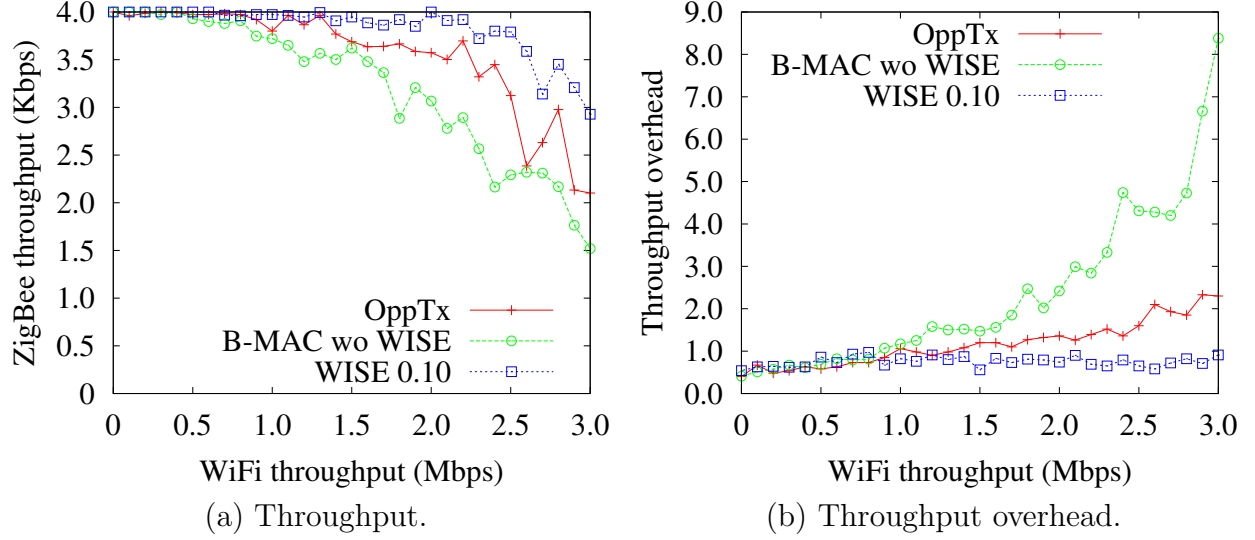
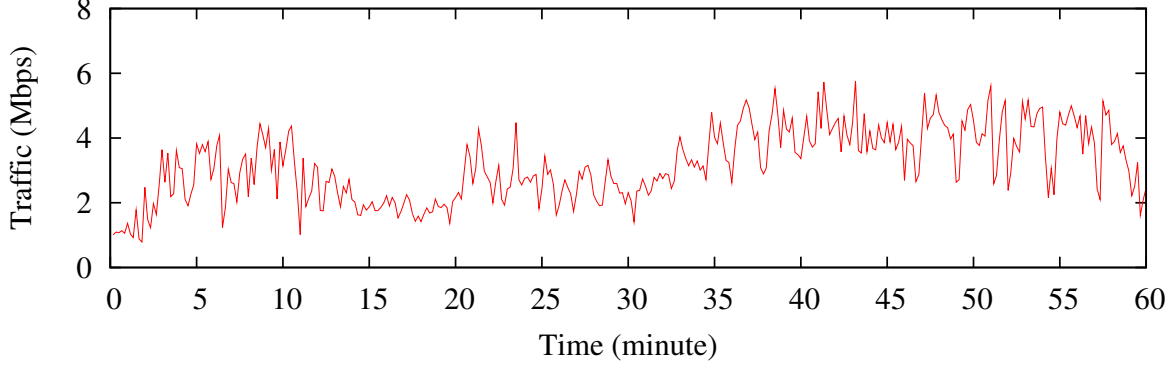


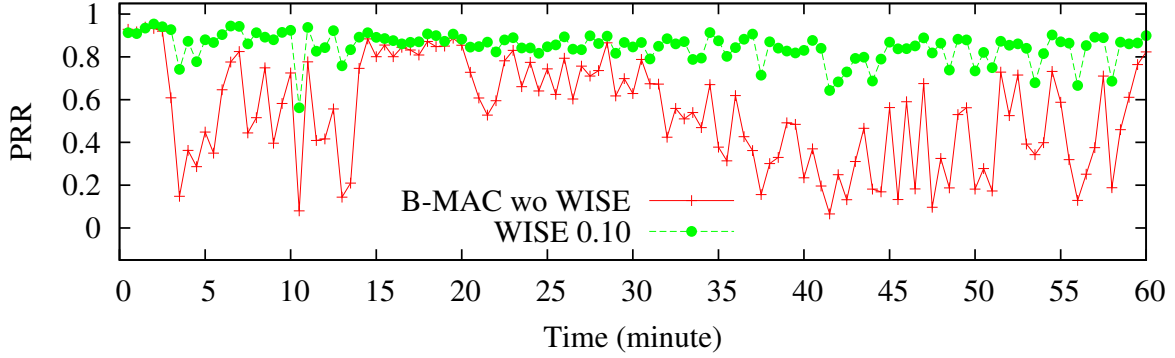
Figure 5.5: ZigBee throughput vs WiFi throughput.

bound of 0.1 constantly achieves a frame delivery ratio above 98%. Fig. 5.5(a) shows the throughput of OppTx, B-MAC, and WISE with collision bound of 0.1. It can be seen that WISE performs consistently better than the other two protocols.

We observe from Fig. 5.4 that, although the frame delivery ratio is always higher than 95%, the throughput of WISE begins to drop when the traffic load of WiFi exceeds 2Mbps. We note that the reason of the throughput drop for WISE is different from the other two protocols. For B-MAC and OppTx, the throughput drop is mainly caused by frame loss because they failed to predict the interference of future WiFi transmissions. In contrast, WISE maintains constant loss rate (as required by the collision probability bound) while decreases the sending rate. This result suggests that, when the channel is heavily loaded by WiFi traffic, WISE will gracefully lower the sending rate to avoid frequent retransmissions, which leads to significantly lower overhead. This phenomenon is illustrated more clearly in Fig. 5.5(b) that shows the impact of WiFi data rate on the throughput overhead of ZigBee. Due to channel sampling, the overhead of WISE is slightly higher than that of



(a) Trace of WiFi traffic volume in OSDI'06 data set.



(b) Trace of ZigBee PRR.

Figure 5.6: Trace of the performance of WISE and B-MAC under the interference of real-life WiFi traffic.

other protocols when WiFi traffic load is low. However, when WiFi traffic load is high, WISE achieves significantly lower overhead than other protocols. Specifically, when the throughput of WiFi is 3.0 Mbps, the throughput overhead of WISE is only 0.65, which is 10.9% and 39.5% of that of B-MAC and OppTx, respectively.

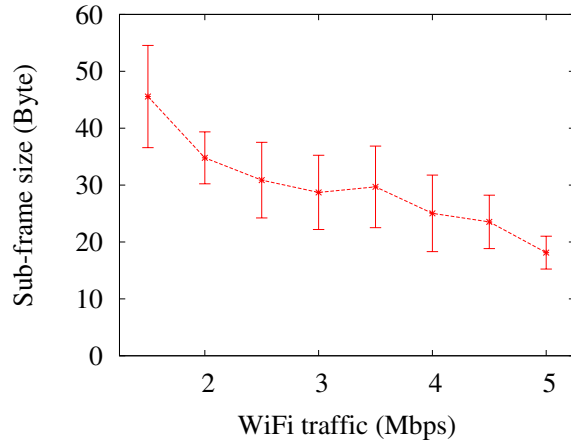
5.5.4 Impact of Real-life WiFi Traffic

We now evaluate WISE under the interference of real-life WiFi traffic. For this purpose, we replace the D-ITG traffic generator used in Section 5.5.3 with a WiFi laptop which replays the packet traces collected in real-life WiFi deployment. In our experiments, 30 packet traces

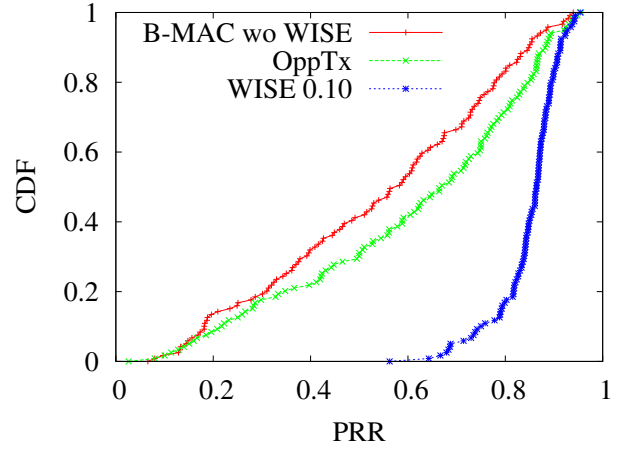
from OSDI'06 data set are randomly selected for replaying, total lasting for around 7 hours. During the experiment, the ZigBee sender unicasts a frame of 80 bytes every 10 ms. Sub-frame retransmission is disabled in WISE. Fig 5.6(a) illustrates the WiFi traffic rate of a one-hour packet trace. We observe that the network is heavy loaded. The traffic rate varies significantly over time, ranging from 1 Mbps to 5 Mbps.

Fig. 5.6(b) evaluates the PRR of B-MAC and WISE under the traffic trace shown in Fig. 5.6(a). We observe that the PRR of B-MAC varies significantly due to the interference of WiFi traffic; while WISE consistently maintains PRR around 90%. This is because WISE tunes its sub-frame size to adapt to the WiFi interference. This is further illustrated in Fig. 5.7(a), which studies the relation between WiFi traffic rate and WISE sub-frame size during a 7-hour experiment. The average of WiFi traffic volume and the WISE sub-frame size are logged every minute. We observe that, under heavy WiFi traffic, WISE adopts a small sub-frame size to assure low probability of collision with WiFi packets; while under light WiFi traffic, WISE increases sub-frame size to improve throughput efficiency.

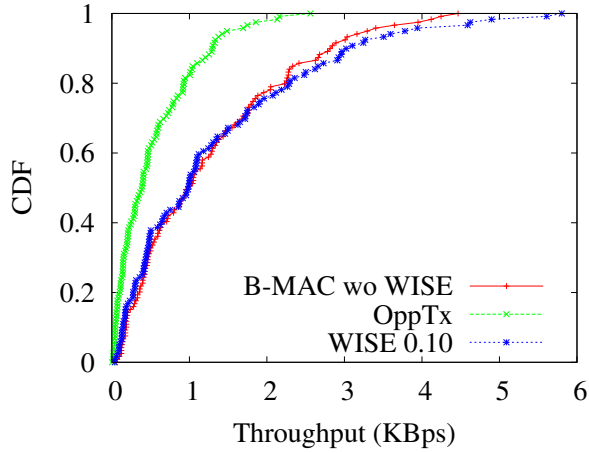
We study the impact of real-life WiFi traffic on ZigBee performance in Fig. 5.7(b). Similar with the results reported in Section 5.5.3, we find that WISE achieves significantly higher PRR than B-MAC and OppTx. In particular, the average PRRs of B-MAC, OppTx and WISE are 0.49, 0.59 and 0.85, respectively. The performance gains of WISE over B-MAC and OppTx are 73% and 44%, respectively. We further evaluate the throughput and throughput overhead of B-MAC, OppTx and WISE in Fig. 5.7(c) and Fig. 5.7(d). We observe that OppTx yields the lowest throughput among all protocols. This is because it defers ZigBee transmissions when the channel is heavily loaded with WiFi traffic. In comparison, WISE splits large frame into sub-frames to utilize the WiFi white space, resulting in better



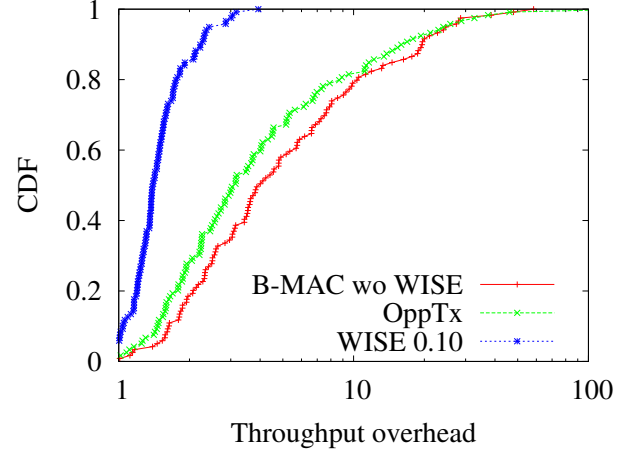
(a) Sub-frame size vs WiFi traffic.



(b) CDF of PRR.



(c) CDF of Throughput.



(d) CDF of throughput overhead.

Figure 5.7: Performance under the interference of real-life WiFi traffic.

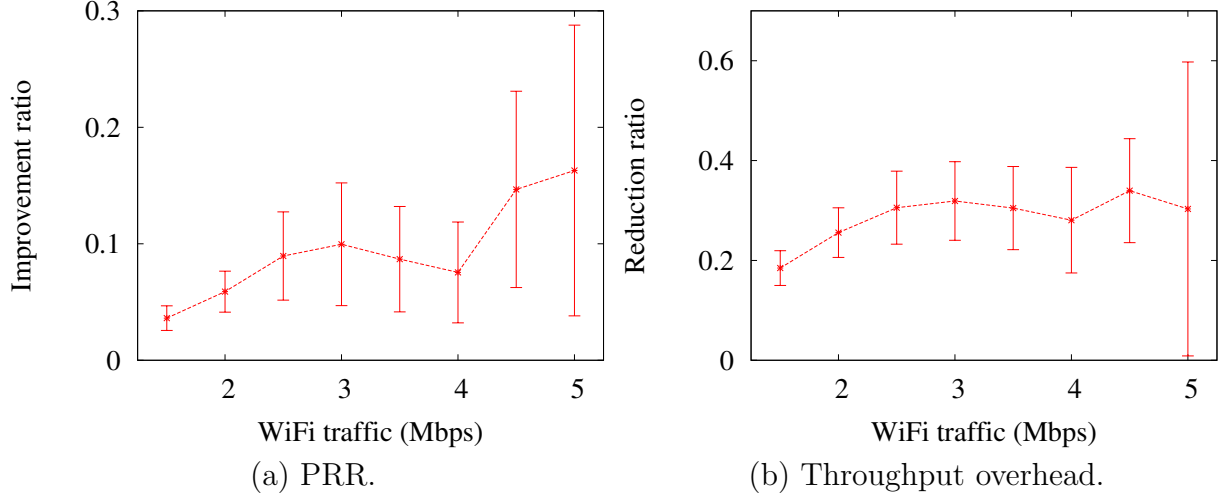


Figure 5.8: Impact of sub-frame retransmission on WISE performance.

throughput. We also observe that WISE incurs the lowest throughput overhead among the three protocols. In particular, the average throughput overhead for B-MAC, OppTx and WISE are 4.0, 3.1 and 1.4. WISE reduces the overhead by 35% and 45%, respectively when compared with B-MAC and OppTx.

5.5.5 Impact of Sub-Frame Retransmission

We also evaluate the impact of sub-frame retransmission on WISE performance. We repeat the experiment introduced in Section 5.5.4, and study the performance of WISE when sub-frame retransmission is enabled. Fig. 5.8(a) shows the PRR improvement ratio under different WiFi traffic rates. We observe that sub-frame retransmission improves PRR for WISE, and the performance gain is higher under a heavy WiFi traffic load. Specifically, when the WiFi traffic rate is below 1.5 Mbps, sub-frame retransmission improves the average PRR by 4.9%; while when the WiFi traffic rate is higher than 3 Mbps, the average improvement ratio is 13.5%. The results demonstrate that partial retransmissions of corrupted sub-frames is more robust than retransmitting the entire frame, especially under interference of heavy

WiFi traffic.

Fig. 5.8(b) evaluates the impact of sub-frame retransmission on ZigBee throughput overhead under different WiFi traffic rates. We observe that sub-frame retransmission consistently reduces throughput overhead of ZigBee despite the variation of WiFi traffic rate and the additional protocol cost, such as sub-frame checksum and block map. The reduction ratio is higher under heavy WiFi traffic load. In particular, when the WiFi traffic load is below 3.5 Mbps, WISE with sub-frame retransmission incurs 22% less throughput overhead. When the WiFi traffic load is higher than 3 Mbps, sub-frame retransmission reduces the average throughput overhead by 31%. The results suggest that sub-frame retransmission is more efficient under heavy WiFi traffic load, where the protocol cost of sub-frame checksum and block map is paid off by the performance gain of partial retransmissions.

5.6 Summary of Chapter

In this chapter, we present WISE, a WhItE Space aware framE adaptation protocol, to tackle the blind terminal problem. WISE allows ZigBee links to achieve assured performance under the heavy interference of WiFi blind terminals. Exploiting the white space of WiFi traffic, WISE allows ZigBee networks co-existing with WiFi to achieve desired link throughput and delivery ratio. Our extensive experiments on a testbed of 802.11 netbooks and 802.15.4 TelosB motes show that WISE achieves 4x and 2x performance gains over the default MAC protocol of 802.15.4 and a state-of-the-art opportunistic transmission scheme, respectively, while only incurs 10.9% and 39.5% of their overhead.

Chapter 6

Conclusion and Future Work

This thesis tackles the challenges of wireless interference and coexistence in the link layer design of wireless networks. In particular, we present the experimental studies and practical solutions of two problems that are not addressed in existing literature, including (1) *the rate-adaptive exposed terminal* (RET) problem and (2) *the blind terminal* problem. In this chapter, we summarize the contributions of this thesis, and then present the open problems.

6.0.1 Contributions

The main contributions of this thesis are summarized as follows.

- *The rate-adaptive exposed terminal problem.*

This thesis investigates the RET problem, where link concurrency cannot be fully exploited by conventional link layers because of their oblivious to the bit rate diversity of underlying PHY. We characterize the properties of RETs based on the topology of a production enterprise WLAN composed of 23 802.11 a/g APs. Our measurements demonstrate the abundance of RETs. Our experiment shows that link concurrency

can be boosted by up to 131% through optimizing the bit rate and medium access of concurrent links. Motivated by the measurements, we develop TRACK, a novel protocol that harnesses RETs leveraging the bit rate diversity of 802.11. TRACK is implemented on commodity 802.11 devices and is evaluated through extensive experiments on a testbed of 17 nodes. Our results show that TRACK improves system throughput by up to 67% and 35% over 802.11 CSMA and a state-of-the-art protocol, without compromising link fairness.

- *The blind terminal problem.*

Based on an empirical study of WiFi and ZigBee coexistence, we identify the blind terminal problem where existing link layer protocols fail to work in co-existing environments due to the heterogeneous PHY and power asymmetry of co-existing devices. We characterize the interference of WiFi blind terminals by studying the *white space* in WiFi traffic. Based on statistical analysis on traces of real-life WiFi traffic, we propose a Pareto model that accurately characterizes the white space in WiFi traffic. Moreover, an analytical framework is developed to model the performance of a ZigBee link co-existing with heavy-loaded WiFi transmitters. We then develop WISE - an adaptive frame control protocol, which allows ZigBee networks co-existing with WiFi to achieve desired link throughput and packet delivery ratio. Our extensive experiments on a testbed of 802.11 laptops and 802.15.4 TelosB motes show that WISE achieves 4x and 2x performance gains over the default MAC protocol of 802.15.4 and a state-of-the-art opportunistic transmission scheme, respectively, while only incurs 10.9% and 39.5% of their overhead.

6.0.2 Open Problems

In this section, we present some open problems in the design of TRACK and WISE.

1. *Extending the design of TRACK.*

TRACK is designed to improve link concurrency using a rate adaptation based approach for 802.11a/g WLANs. We will extend TRACK for 802.11n, which adopts multiple-input multiple-output (MIMO) and 40 MHz channel in the PHY layer. We also plan to incorporate TRACK with flexible channelization [54] in future work. The major extension needed is to model the link conflicts when MIMO and flexible channelization are enabled. The framework of TRACK can be integrated seamlessly with different conflicts maps.

The current design of TRACK only considers clients of limited mobility. Supporting mobile user in TRACK raises two challenges. First, mobile wireless clients usually work in the power saving mode [41] [50], in which the 802.11 radio is scheduled to sleep periodically. Packets are received only when the client is awake. To address this issue, we plan to investigate a joint design of link admission and sleep scheduling. Second, when the client is moving, downlink quality will experience more variations, complicating channel measurement. We will study efficient algorithms for measuring mobile channels.

2. *Extending the design of WISE.*

As we discussed in Section 3.3.3, white space can be very limited when there is heavy WiFi traffic. Currently, WISE cannot utilize those short white space, if the lifetime of white space is only a little longer than the time required to transmit a ZigBee pream-

ble. We plan to exploit forward error correction to harness short WiFi white space. Specifically, we will extend WISE by using coding to protect the payload of a ZigBee frame, while protecting the preamble using the model-driven prediction presented in Section 5.3. We will extend the frame adaptation model to optimize the trade-off between coding gain and coding overhead.

BIBLIOGRAPHY

Bibliography

- [1] ath9k-linux wireless. <http://linuxwireless.org/en/users/Drivers/ath9k>.
- [2] Atheros super g. <http://www.super-g.com/>,.
- [3] D-itg distributed internet traffic generator. <http://www.grid.unina.it/software/ITG/>.
- [4] Ieee 802.11e-2005. <http://standards.ieee.org/>.
- [5] Linux wireless. <http://linuxwireless.org/>.
- [6] Minstrel specification. <http://madwifi-project.org/browser/madwifi/>.
- [7] Nabeel Ahmed, Usman Ismail, Srinivasan Keshav, and Konstantina Papagiannaki. On-line estimation of rf interference. In *ACM CoNext*, 2008.
- [8] P. Bahl, R. Chancre, and J. Dungeon. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *ACM MobiCom*, 2004.
- [9] Bandspeed. Understanding the effects of radio frequency (rf) interference on wlan performance and security, 2010.
- [10] John Bicket. Bit-rate selection in wireless networks. Master’s thesis, MIT, 2005.
- [11] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. A case for adapting channel width in wireless networks. In *ACM SIGCOMM*, 2008.

- [12] Ranveer Chandra, Ratul Mahajan, Venkat Padmanabhan, and Ming Zhang. Crawdad data set microsoft/osdi2006 (v. 2007-05-23), 2007.
- [13] Yu-Chung Cheng, John Bellardo, Péter Benkő, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *ACM SIGCOMM*, 2006.
- [14] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 1997.
- [15] R.L. Cruz and A.V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 702 – 711 vol.1, march-3 april 2003.
- [16] Shuguang Cui, R. Madan, A.J. Goldsmith, and S. Lall. Cross-layer energy and delay optimization in small-scale sensor networks. *Wireless Communications, IEEE Transactions on*, 6(10):3688 –3699, october 2007.
- [17] Henri Dubois-Ferrière, Deborah Estrin, and Martin Vetterli. Packet combining in sensor networks. In *ACM SenSys*, 2005.
- [18] Jakob Eriksson, Sharad Agarwal, Paramvir Bahl, and Jitendra Padhy. Feasibility study of mesh networks for all-wireless offices. In *ACM MobiSys*, 2006.
- [19] Raghu K. Ganti, Praveen Jayachandran, Haiyun Luo, and Tarek F. Abdelzaher. Datalink streaming in wireless sensor networks. In *SenSys*, 2006.
- [20] Stefan Geirhofer, Lang Tong, and Brian Sadler. Dynamic spectrum access in wlan channels: Empirical model and its stochastic analysis. In *ACM TAPAS*, 2006.
- [21] Stefan Geirhofer, Lang Tong, and Brian Sadler. Cognitive medium access: Constraining interference based on experimental models. *IEEE Journal on Selected Areas in Communications*, 2009.
- [22] Shyamnath Gollakota, Samuel David Perli, and Dina Katabi. Interference alignment and cancellation. In *ACM SIGCOMM*, 2009.
- [23] Farpoint Group. Evaluating interference in wireless lans: Recommended practice, 2010. White paper FPG 2010-135.1.

- [24] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *ACM SigCOMM*, 2007.
- [25] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: Interference cancellation for wireless lans. In *ACM MobiCom*, 2008.
- [26] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetheral. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM*, 2010.
- [27] IEEE. Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Standard 802.15.4*, 2003.
- [28] Raj Jain, Dah-Ming Chiu, and William Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Technical report, Eastern Research Lab, 1984.
- [29] Kyle Jamieson and Hari Balakrishnan. Ppr: Partial packet recovery for wireless networks. In *ACM SigCOMM*, 2007.
- [30] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. Efficient channel-aware rate adaptation. In *ACM MobiSys*, 2008.
- [31] J. H. Kim and J. K. Lee. Capture Effects of Wireless CSMA/CA Protocols in Rayleigh and Shadow Fading Channels. *IEEE Transactions on Vehicular Technology*, 1999.
- [32] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. Cara: Collision-aware rate adaptation for ieee 802.11 wlans. In *IEEE INFOCOM*, 2006.
- [33] U.C. Kozat, I. Koutsopoulos, and L. Tassiulas. A framework for cross-layer design of energy-efficient communication with qos provisioning in multi-hop wireless networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1446 – 1456 vol.2, march 2004.
- [34] Hojoong Kwon, Tae Hyun Kim, Sunghyun Choi, and Byeong Gi Lee. A cross-layer strategy for energy-efficient reliable delivery in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 5(12):3689 –3699, december 2006.
- [35] Hieu Khac Le, Dan Henriksson, and Tarek Abdelzaher. A control theory approach to throughput optimization in multi-channel collection sensor networks. In *IPSN*, 2007.

- [36] Hieu Khac Le, Dan Henriksson, and Tarek Abdelzaher. A practical multi-channel media access control protocol for wireless sensor networks. In *IPSN*, 2008.
- [37] Will E. Leland, Walter Willinger, Murad S. Taqqu, and Daniel V. Wilson. On the self-similar nature of ethernet traffic. *ACM SIGCOMM Comput. Commun. Rev.*, 1995.
- [38] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *ACM SenSys*, 2010.
- [39] R. Madan, S. Cui, S. Lal, and A. Goldsmith. Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 5(11):3142–3152, november 2006.
- [40] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *ACM SenSys*, 2008.
- [41] Justin Manweiler and Romit Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. In *ACM MobiSys*, 2011.
- [42] Miercom. Cisco cleanair competitive testing, 2010. Lab test report DR100409D.
- [43] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for wlans. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3):19–31, July 2005.
- [44] Arunesh Mishra, Vivek Shrivastava, Suman Banerjee, and William Arbaugh. Partially overlapped channels not considered harmful. In *SIGMetrics*, 2006.
- [45] Thomas Moscibroda, Ranveer Chandra, Yunnan Wu, Sudipta Sengupta, Paramvir Bahl, and Yuan Yuan. Load-aware spectrum distribution in wireless lans. In *ICNP*, 2008.
- [46] Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *in 4th USENIX Symposium on Networked Systems Design and Implementation*, pages 87–100, 2007.
- [47] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 1995.
- [48] Guangyu Pei and C. Chien. Low power tdma in large wireless sensor networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-*

Centric Operations: Creating the Information Force. IEEE, volume 1, pages 347 – 351 vol.1, 2001.

- [49] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *ACM SenSys*, 2007.
- [50] Andrew J. Pyles, Zhen Ren, Gang Zhou, and Xue Liu. Sifi: exploiting voip silence for wifi energy savings insmart phones. In *ACM UbiComp*, 2011.
- [51] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 181–192, New York, NY, USA, 2003. ACM.
- [52] Bhaskaran Raman and Kameswari Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, pages 156–169, New York, NY, USA, 2005. ACM.
- [53] Ananth Rao and Ion Stoica. An overlay mac layer for 802.11 networks. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, MobiSys '05, pages 135–148, New York, NY, USA, 2005. ACM.
- [54] Shravan K. Rayanchu, Vivek Shrivastava, Suman Banerjee, and Ranveer Chandra. Fluid: improving throughputs in enterprise wireless lans through flexible channelization. In *ACM MobiCom*, 2011.
- [55] Aaron Schulman, Dave Levin, and Neil Spring. Crawdad data set umd/sigcomm2008 (v. 2009-03-02), 2009.
- [56] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Accurate: Constellation based rate estimation in wireless networks. In *NSDI*, 2010.
- [57] Mo Sha, Guoliang Xing, Gang Zhou, Shucheng Liu, and Xiaorui Wang. C-mac: Model-driven concurrent medium access control for wireless sensor networks. In *IEEE INFOCOM*, 2009.
- [58] Liqi Shi and Abraham O. Fapojuwo. Tdma scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(7):927–940, July 2010.

- [59] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. Centaur: Realizing the full potential of centralized wlans through a hybrid data path. In *ACM MobiCom*, 2009.
- [60] Jungmin So and Nitin Vaidya. Multi-channel mac for ad hoc networks:handling multi-channel hidden terminals using a single transceiver. In *MobiHoc*, 2004.
- [61] Dongjin Son, Bhaskar Krishnamachari, and John Heideman. Experimental study of concurrent transmission in wireless sensor networks. In *ACM Sensys*, 2006.
- [62] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *ACM Sensys*, 2006.
- [63] Kannan Srinivasan, Maria A. Kazandjieva, Saatvik Agarwal, and Philip Levis. The beta-factor: measuring wireless link burstiness. In *ACM Sensys*, 2008.
- [64] Wong Starsky, Yang Hao, Lu Songwu, and Bharghavan Vaduvur. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, 2006.
- [65] Godfrey Tan and John Guttag. Time-based fairness improves performance in multi-rate wlans. In *USENIX Annual Technical Conference*, 2004.
- [66] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM*, 2009.
- [67] Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan. Harnessing exposed terminals in wireless networks. In *NSDI*, 2008.
- [68] 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 RTSS*, 2011.
- [69] William Wei. Time series analysis. *Addison-Wesley*, 1990.
- [70] Yafeng Wu, John A. Stankovic, Tian He, and Shan Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Infocom*, 2008.
- [71] Yafeng Wu, Gang Zhou, and John A. Stankovic. Acr: Active collision recovery in dense wireless sensor networks. In *INFOCOM*, 2010.

- [72] Jiansong Zhang, Kun Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. A practical snr-guided rate adaptation. In *IEEE INFOCOM*, 2008.
- [73] Xinyu Zhang and Kang G. Shin. Enabling coexistence of heterogeneous wireless systems: case for zigbee and wifi. In *ACM MobiHoc*, 2011.
- [74] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. Zifi: Wireless lan discovery via zigbee interference signatures. In *ACM MobiCom*, 2010.