IoT Networking: From Coexistence to Collaboration

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Song Min Kim

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Tian He, Adviser

May, 2016

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school. First of all, I am deeply indebted to my advisor Prof. Tian He, for his guidance, support, inspiration, as well as the great confidence and the tremendous patience he has shown me. Without him this dissertation would not been possible. I am very grateful for Prof. Gerald Sobelman, Prof. Zhi-Li Zhang, and Prof. Abhishek Chandra, for serving as my dissertation committee and for their insightful suggestions and feedback. Also, I am extremely thankful to Prof. Kang G. Shin at the University of Michigan and Dr. Yu Gu at IBM for their support on my work and future career.

I was fortunate to be a part of MiNDs group, where I have spent a large part of my time. I sincerely appreciate the help and contributions of the former and current members including Dr. Jaehoon Jeong, Dr. Ziguo Zhong, Dr. Ting Zhu, Dr. Shuo Guo, Dr. Desheng Zhang, Yongle Cao, Shi Bai, Shuai Wang, Wenchao Jiang, Zhimeng Yin, and Song Liu. It has always been enjoyable and fruitful to work with them. I would also like to thank my collaborator Dr. Shigemi Ishida for the valuable input with in-depth systems perspective. Furthermore, I would like to show my greatest thanks and love to my family – my wife, baby son, parents, parents-in-law, my only brother, and sisters-in-law – for their unwavering support. Without their love, care and encouragement, I would not have come this far. Last but not the least, I gratefully acknowledge financial support from the Department of Computer Science and Engineering at the University of Minnesota, the National Science Foundation, ACM, and IEEE.

Dedication

To Nahyun and Tobin.

Abstract

The new computing paradigm of IoT (Internet of Things) is carried out by globally and massively interconnected devices. This is founded on the unprecedented proliferation of heterogeneous wireless technologies in the last decade, each offering convenience in different aspects of our daily lives – WiFi enables hassle-free Internet access and Bluetooth allows prevalent healthcare with its wireless heart monitors. However, wireless technologies are victims of their own success; dense deployment of wireless devices intensify the interference between them, which recently has become a major cause of performance degradation. My dissertation work is an effort to address the issue with practical and cost-effective solutions.

The dissertation consists of two main parts. The first part proposes approaches to achieve high-performance networking under arbitrary interference from unknown sources. Namely, cETX [1] and CorrModel [2] are proposed, where they enable efficient unicast and broadcast under severely interfered channel. This is achieved by observing the different aspects of interference including volume as well as temporal/spatial pattern and dynamics. To maximize the adaptability, both cETX and CorrModel are designed as generic techniques to support networks running different protocols, under a wide range of settings. Indoor and outdoor testbed evaluations performed on twelve unicast and nine broadcast protocols demonstrate that they bring 20-30% performance gain.

The second part of the dissertation takes a step further, to explore the opportunity behind the coexistence of heterogeneous wireless technologies. The dissertation introduces FreeBee [3], which extends direct connectivity beyond wireless technologies to enable collaboration and mutual supplementation. By doing so FreeBee not only enables cross-technology interference mitigation but also brings advanced services such as context-aware smart operation. Evaluations reveals that FreeBee achieves reliable communication in under a second and supports mobility of up to 30mph.

Contents

List of Figures

Chapter 1

Introduction

IoT (Internet of Things) envisions improving the quality of everyday lives through pervasively interconnected objects, covering every corner of our living space. To this end, my dissertation is an effort to bring wireless networking into the picture of a realistic IoT environment; massive number of devices operating under diverse, specialized networks. According to ABI Research, 3B Bluetooth, 1.5B WiFi, and 70M ZigBee devices were deployed in 2014, with the exponential growth expected with emerging technologies ranging from wearable gadgets to smart buildings. I have investigated resolving the issues as well as exploring opportunities behind these crowded devices; from harmonious coexistence to co-prosperity via collaborative networking.

First, I introduce networking techniques to enable heterogeneous technologies harmoniously utilize the spectrum, the limited resource in nature. The key issue of coexistence is interference between devices that cause spectrum under-utilization, where the degree and pattern are dynamic with the number of interferers and their activities (e.g., video streaming/web surfing). My work within this category, cETX [1] and CorrModel [2], proposes to autonomously accommodate the temporal and spatial aspects of interference. To take it a step further, collaborative networking is designed to exploit the full potential of the dense, heterogeneous devices in IoT. Technologies are specialized, possessing strengths in distinct areas and carrying particular data. For example, WiFi has a high data rate and Internet accessibility, while Bluetooth is energy-efficient and often holds health information. The best of both worlds is achievable via collaboration, where the cross-connectivity is required and critically beneficial. My work of

FreeBee [1], for the first time, establishes direct communication among heterogeneous wireless entities without additional spectrum usage.

Cost is often the biggest hurdle for a new technology to attract users and vendors. All my work is commonly built upon the mindset of practicality, pursuing immediate impact while minimizing cost, achieved by (i) implementation/verification on multiple, easily attainable off-the-shelf devices, (ii) retaining compatibility to industry standards such as IEEE 802.15.4 and 802.11, and (iii) evaluations in diverse realistic settings spanning across both indoors and outdoors. These principles enable utilizing readily deployed platforms and minimize conflicts with legacy systems to significantly cut down the adaptation cost. Specifically, the contributions of the dissertation is as follows:

- I introduce cETX, a network metric that enable efficient unicasting under arbitrary interference from heterogeneous wireless technologies. To the best of my knowledge, this is the first work to introduce a unified metric embracing full aspects of interference: volume, patter, dynamics. The highlight of the cETX metric is its broad applicability and effectiveness. Evaluations on ZigBee (802.15.4) and Wi-Fi (802.11b/g/n) testbeds deployed in a lab, corridor, and on a bridge reveal that: Simply replacing ETX with cETX (i) cuts down the error by 70.2% and 62.1%, respectively, and (ii) saves averages of 22% and 37% communication cost in three unicast [4, 5, 6] and nine broadcast protocols [7, 8, 9, 10, 11, 12, 13] at the price of only 0.7% additional overhead.

- I porpose CorrModel, an online scheme that effectively captures spatial patterns of interferences to enable energy-efficient broadcasting under coexisting environment. CorrModel is a general framework that uses SINR (Signal to Interference plus Noise Ratio) to detect interference patterns, followed by modeling for in-network use. It is shown that my design is light-weight, both computation and storage-wise. CorrModel is applied to opportunistic routing and network coding on a physical 802.15.4 test-bed for energy savings of 25% and 15%.

- A novel cross-technology communication of FreeBee is presented. It enables direct unicast as well as cross-technology/channel broadcast among three popular wireless technologies: WiFi, ZigBee, and Bluetooth. The design aims to shed the light on the opportunities that cross-technology communication has to offer including,

but not limited to, cross-technology interference mitigation and context-aware smart operation. The key concept of FreeBee is to modulate symbol messages by shifting the timing of periodic beacon frames already mandatory for wireless standards without incurring extra traffic. Such a generic cross-technology design consumes zero additional bandwidth, allowing continuous broadcast to safely reach mobile and/or duty-cycled devices. A new *interval multiplexing* technique is proposed to enable concurrent broadcasts from multiple senders or boost the transmission rate of a single sender. Theoretical and experimental exploration reveals that FreeBee offers a reliable symbol delivery under a second and supports mobility of 30mph and low duty-cycle operations of under 5%.

The dissertation is organized as follows. Chapter 2 offers background knowledge on IoT networking and the issue of wireless coexistence. Chapters 3 and 4 introduces unicast and broadcast techniques under coexistence. Chapter 5 presents collaboration framework between heterogeneities, followed by future research plan presented in Chapter 6. Finally, Chapter 7 concludes the dissertation.

Chapter 2

Background

In the aim to provide sufficient context, this chapter discusses essential background concepts related to this dissertation, namely, IoT networking and wireless coexistence.

2.1 IoT Network

Figure 2.1: Heterogeneous networks in IoT

The emerging computing paradigm of IoT envisions massively and seamlessly connected devices covering every corners of our living spaces as well as our bodies via wearables, which is anticipated to be as many as 50 billion devices by 2020, according

to Cisco [14]. Through global connectivity these devices offer ubiquitous services ranging from data collection to actuation to improve our daily lives by supporting diverse applications in the domains including, but not limited to, smart healthcare, intelligent transportation systems, and manufacturing. It should be noted that all these applications are built on top of ad-hoc and Internet connectivity of the devices. Therefore networking naturally plays a fundamental and essential role in IoT, where majority of them are expected to be connected wirelessly for non-intrusiveness.

Figure 2.1 demonstrates IoT networking scenario within a smart hospital environment, where heterogeneous networks operating at different scales are utilized; Among them, WiFi, and Bluetooth form Local Area Network (LAN), and Personal Area Network (PAN), respectively. Aforementioned technologies are commonly used today, where IoT also consists of a new layer of network referred to as *Network of Things*. This embeds diverse everyday objects including thermostat and humidity detector, where via Machine-to-Machine (M2M) communication they collaboratively and autonomously controls the hospital environment without human intervention.

2.2 Wireless Heterogeneity and Coexistence

Table 2.1: Wireless technology specifications

	ZigBee (IEEE 802.15.4)	**Bluetooth (IEEE 802.15.1)**	**WiFi (IEEE 802.11n)**
Modulation	OQPSK, DSSS	GFSK, 8DPSK	64-QAM, OFDM
Range	30m	100m	250m
Data rate	250kbps	3Mbps	600Mbps
Operating freq.	2.4GHz ISM	2.4GHz ISM	2.4, 5GHz ISM
E consumption	Very low	Low	High

As demonstrated in the previous subchapter, IoT network consists of diverse wireless technologies, so as to take advantage of unique strengths of each technology. Table 2.1 enlists differences between the technologies, where the clear differences indicate their specialized usage. For example, WiFi offers high communication speed of 600Mbps suitable for delivering large-size data including multimedia files. While the ZigBee exhibits a low rate of 250kbps, it is extremely energy efficient – making it a great option for devices with small form-factor that operate on a limited energy source, such as a

coin cell battery.

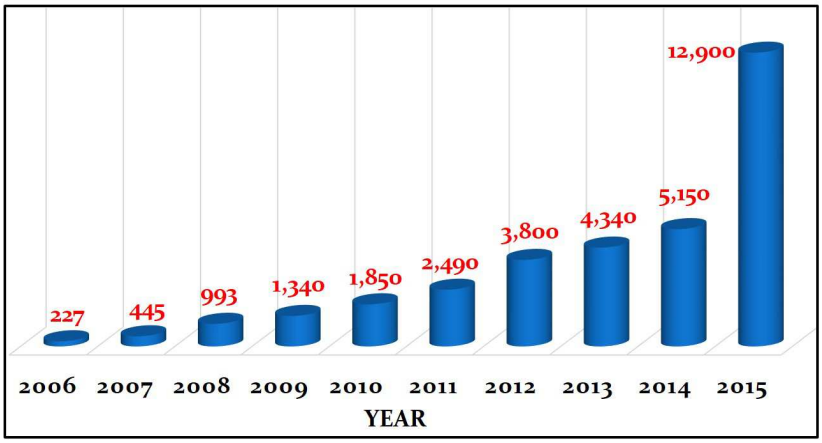


Figure 2.2: Number of news articles over the last decade

As in Table 2.1, the technologies all share the common wireless medium of the unlicensed 2.4GHz ISM (industrial, scientific and medical) band. However, as much as they are specialized, they are incompatible with each other. In other words, the technologies lack the functionality of harmonious and cooperative sharing of the resource; instead, they compete with each other and cause destructive wireless interference, which is known as the *coexistence problem* [15, 16, 17, 18, 19, 20]. This has become a significant issue, reported in various domains including healthcare and mobile communication. Figure 2.2 depicts the number of news articles on wireless interference over the last decade (according to Google), which has increased by 57× to reach more than 12K in 2015. Moreover, the issue is expected to be intensified in the coming years, with ever-increasing wireless devices penetrating into every corner of our daily lives.

Chapter 3

Unicast under Coexistence

3.1 Introduction

Wireless links are unreliable and prone to losses due to noise and interference [21, 22, 23, 24, 25]. To effectively utilize these unreliable wireless links, research calls for metrics that can accurately estimate the qualities of individual links as well as end-to-end paths. Among the extensive volume of admirable studies dealing with routing under lossy links [26, 27, 28], expected transmission count (ETX) [27] is the most popular for its generality and effectiveness, and has been cited over 3,000 times according to Google Scholar.

My further studies on link characteristics, however, reveal that ETX suffers from a few limitations. First, recent work on the temporal property of wireless links [29, 30] shows that transmissions over a link within a short time interval are highly dependent, indicating the existence of temporal correlation among transmissions. This property is not taken into account in the design of the ETX metric, imposing a limitation on its ability to reflect the true link quality. This is because ETX is defined as the inverse of packet reception ratio (PRR). This value is in fact the mean of the geometric random variable with independent trials, indicating that ETX is implicitly built on the independence assumption among transmissions.

Another link characteristic enlightened by current work ([31, 32]) is the existence of a reception correlation among the receivers of broadcast packets due to cross-network interference and correlated shadowing. Simply put, the receptions of broadcast packets at adjacent nodes are correlated at the same time instant, indicating spatial correlation. In this work, it is observed that spatial correlation interplays with temporal correlation to exhibit reception correlation at consecutive links. In other words, there exists spatiotemporal correlation, i.e., the packet reception of a link is dependent on the recent reception at preceding links along a path. Since existing metrics including ETX are hop-by-hop designs, they fail to take spatiotemporal correlation into account.

This chapter reveals why ETX cannot represent the true quality of links in the face of temporal and spatiotemporal correlation. To address this issue, I design an enhanced version of ETX, called correlated ETX (cETX) that explores the phenomena to reflect the true performance of links and paths more precisely compared to ETX. In detail, I provide a unified metric embracing both temporal and spatiotemporal correlation

factors. While keeping the intuitive idea of ETX intact (i.e., explicitly quantifying expected number of transmissions), my metric selects better links and paths that leads to less transmissions, thus achieving energy savings. In summary, my contributions are as follows:

- I experimentally reveal the phenomenon of spatiotemporal correlation and its potential impact on routing designs.
- I propose a new metric – cETX, capable of capturing temporal and spatiotemporal properties of wireless links. To the best of my knowledge, this is the first work to embed the two effects into a unified metric.
- I demonstrate how to incorporate cETX into (i) classical shortest path unicast algorithms and (ii) classical/collaborative broadcast algorithms, while maintaining the unicast path optimality and minimizing network-wide broadcast overhead distributively.
- I compare the accuracy of cETX and ETX through ZigBee and Wi-Fi testbed experiments, and via analysis on the Roofnet [33] trace to demonstrate error reduction of 70.2%, 62.1%, and 33.2%, respectively.
- I experimentally validate the applicability and effectiveness of cETX by replacing ETX with cETX in three unicast and nine broadcast protocols utilizing diverse network structures. cETX cuts down averages of 22% and 37% communication cost at the price of only 0.7% additional overhead via parameter piggybacking.

This chapter is organized as follows. Subchapter 3.2 motivates the work. Subchapter 3.3 follows with the underlying model and reasoning. Subchapter 3.4 and Subchapter 3.5 present the design and implementation of cETX, where its applications are introduced in Subchapter 3.6. Subchapter 3.7 evaluates cETX, followed by in-situ deployment demonstration in Subchapter 3.8. I summarize related work in Subchapter 3.9 and conclude in Subchapter 3.10.

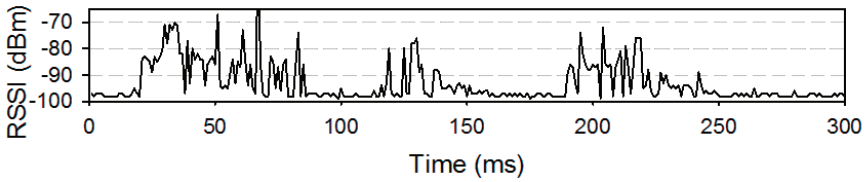


Figure 3.1: Interference Burst

3.2 Motivation

Spatiotemporal correlation occurs with the interference and shadowing spanning through multiple links and last for a duration of multiple packets. Figure 3.1 shows the interference captured under an office environment where many of the spike bursts last over tens of milliseconds. The captured signal strength ranges from -70 to -90dBm, a level of typical Wi-Fi signals. In sum, the observed interference from Wi-Fi is not only explosive, but also strong enough to simultaneously affect multiple low-power 802.15.4 links, suggesting that it is one of the causes of spatiotemporal correlation.

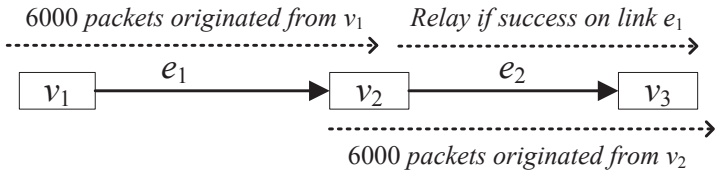


Figure 3.2: Experiment Topology

3.2.1 Spatiotemporal Correlation Revealed

I experimentally validate the phenomenon of spatiotemporal correlation in practice, among consecutive wireless links along a path. A simple two-hop network is formed as shown in Figure 3.2, operating on 802.15.4 channel 12 with a transmission power of -25dBm. Ten such networks are deployed for statistical significance. In each network, two cases are tested. **Case #1:** Node v_1 transmits 6000 packets to v_2. Upon successfully receiving a packet, v_2 *relays* the packet to v_3 five times, each after 2, 8, 14, 20, and 26ms delay. **Case #2:** Node v_2 transmits 6000 packets generated by itself to v_3. Please note that, in case #1, transmission over link e_2 only occurs upon success on e_1. Therefore

the reception success probability on e_2 in this case is conditional, given the success on preceding link e_1. On the other hand, the success probability over e_2 in case #2 is considered marginal, as it is irrelevant to e_1.

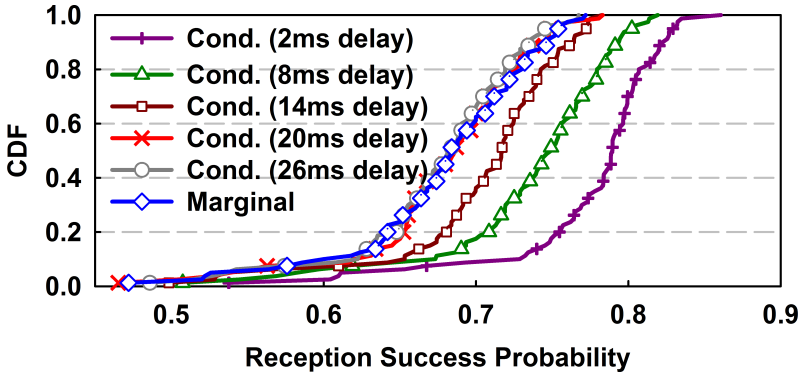


Figure 3.3: Spatiotemporal correlation validation

Figure 3.3 shows the experiment results of e_2's reception success probabilities, where conditional curves are obtained from case #1, and the marginal is from case #2. There exists a significant difference between the conditional probability with $2ms$ delay and the marginal probability. This supports the existence of spatiotemporal (positive) correlation between e_1 and e_2. That is, success on the preceding link (e_1) strongly implies success on the following link (e_2). Moreover, 20 and $26ms$ curves nearly overlap with the marginal curve, indicating spatiotemporal correlation fades away after $20ms$. To conclude, the experiment result gives us a simple but effective design guidance: I can increase the chance of a successful reception of a packet by simply relaying it within a short time interval, e.g., within $20ms$ in my experiment.

Link1:	*F*	*S*	*S*	*F*	*S*	*F*	*F*	*S*	*F*	*S*
Link2:	*F*	*F*	*F*	*S*	*S*	*S*	*F*	*F*	*S*	*S*
	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9

Figure 3.4: Temporal correlation effect

3.2.2 Impact of Temporal Correlation

I illustrate the effect of temporal correlation via a simple example in Figure 3.4. In the history of ten transmissions of two links, S and F represent transmission success and failure. Since both links experience five S and five F, ETX indicates that the two links have equal performance, i.e., 10/5=2 expected transmissions. However, the true average transmissions are not equal. Let's consider the case where I start transmitting at t_0. On Link1, the transmission succeeds at t_1 which causes two transmissions (i.e., fail at t_0, success at t_1). It costs four on Link2 as it succeeds at t_3 after three failures. Similarly, when I start the transmission at t_1, Link1 succeeds in a single transmission, while Link2 requires three.

When the start of a transmission occurs with equal probabilities across $t_0 - t_9$, I obtain the true average number of transmissions by repeating the above process up to t_9 and taking the average. This yields $(2 + 1 + \cdots + 1)/10 = 1.6$ and $(4 + 3 + \cdots + 1)/10 = 1.9$ for Link1 and Link2, indicating that Link1 is indeed a better link, despite their equal ETX. The performance difference of the two links can be described via conditional probability of failure followed by success, i.e, $Pr(S^n|F^{n-1})$ for $1 \leq n \leq 9$. This represents the link's ability to recover from failures. Thus, larger value indicate better performance. The values are 0.8 and 0.4 for Link1 and Link2, which confirms my argument.

Figure 3.5: Temporal Correlation in empirical traces

The observation is further verified on the real traces in Figure 3.5, collected from two pairs of MICAz nodes. Each trace from two links holds a record of 10^3 packet transmissions. Black bars indicate failures while whites indicate successes. The packet reception ratios of Link3 and Link4 are 0.49 and 0.5, leading to approximately the same ETX of 2 for both links. However, the difference in the true average transmissions is quite large;

2.05 and 2.98 for Link3 and Link4, respectively. This is due to the consecutive failure chunks demonstrated in Link4; it is less able to recover from the failure than Link3. This difference again can be represented by the conditional probability. The values of $Pr(S^n|F^{n-1})$ are 0.49 and 0.29 for Link3 and Link4.

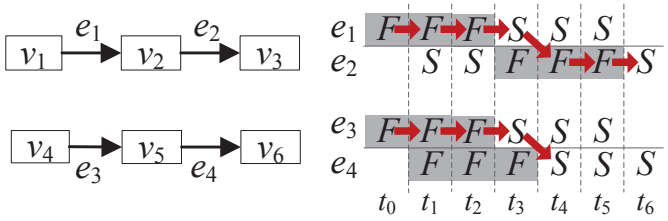


Figure 3.6: Impact of spatiotemporal correlation

3.2.3 Impact of Spatiotemporal Correlation

In Figure 3.6, I use two two-hop paths to show how spatiotemporal correlation affects the network performance that ETX fails to capture. The packet reception ratios of every link in the figure, e_1-e_4, are equally 1/2. This yields ETX value of $2 + 2 = 4$ for both upper and lower paths. However, the links in the lower path, e_3 and e_4, are more positively correlated to each other than those (e_1 and e_2) in the upper path. Starting with the upper path, suppose that node v_1 has a packet to deliver to node v_3 and starts sending at time t_0. The packet successfully reaches v_2 after four transmissions (i.e. F→F→F→S). The transmission on e_2 starts at t_4 but succeeds at t_6, consuming three transmissions (F→F→S). Therefore, end-to-end delivery in the upper path consumes a total of seven transmissions. This forwarding flow is demonstrated in Figure 3.6 as arrows. Similarly, I can easily see that the lower path requires five transmissions. This indicates that the lower path performs better than the upper path, despite their equivalence in the ETX values.

This example suggests that spatiotemporal correlation can improve the network performance. That is, when consecutive links in a path are strongly (positively) correlated, forwarding a packet shortly after receiving it from the preceding link has a high chance of successful receptions in the following link. Therefore, spatiotemporal correlation should be considered for an accurate performance measurement in multi-hop scenarios. To do

so, I represent the degree of spatiotemporal correlation with conditional probability, i.e., $Pr(S_i^n|S_{i-1}^{n-1})$ at arbitrary time t_n when e_{i-1} precedes e_i. To verify, the conditional probability in this example yields $0.33 = Pr(S_2^n|S_1^{n-1}) < Pr(S_4^n|S_3^{n-1}) = 1$, correctly reflecting the relative performance of the two paths under spatiotemporal correlation.

3.3 Design Background

In this subchapter, I first present the model that lays the foundation of my design. Then, I provide empirical evidences and analysis on the suitability of such a model for my purpose of capturing correlation.

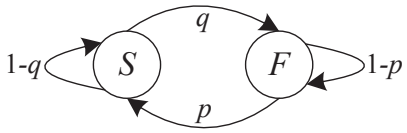


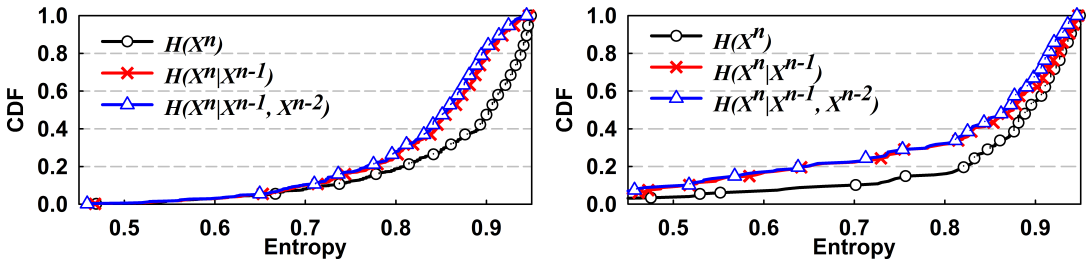
Figure 3.7: Gilbert Model

3.3.1 Underlying Model

This subchapter presents the underlying model to help understand my design presented in the next subchapter. Recall in Subchapter 4.2 I showed that both temporal and spatiotemporal correlations can be quantified by conditional probabilities. A simple way of modeling with conditional probabilities is the Markov representation. My design starts off from the *Gilbert Model* [34] in Figure 3.7, a famous channel model based on two state Markov chain. In the model, S indicates error-free state, while F is when the channel is erroneous. The state transition probabilities are denoted as p and q. That is, by letting $Pr(F^n)$ and $Pr(S^{n-1})$ denote the probabilities of being at states F and S at times t_n and t_{n-1} respectively, then $p = Pr(S^n|F^{n-1})$. Similarly, $q = Pr(F^n|S^{n-1})$. Moreover, steady state probabilities of the two states are computed as

$$\pi_S = \frac{p}{p+q}, \pi_F = \frac{q}{p+q} \tag{3.1}$$

where they indicate the chance of being at the corresponding state at an arbitrary time instant regardless of the initial state, after a sufficient number of state transitions. I



(a) Under Temporal Correlation (b) Under Spatiotemporal Correlation

Figure 3.8: Residual entropies under correlations

follow this model to represent status of a single link.That is, a transmission over the link succeeds if and only if the link is in state S, while fails in state F. This lets transition probabilities, p and q, to indicate temporal correlation within the link. Extending the model to capture spatiotemporal correlation will be discussed later in Subchapter 3.4.

3.3.2 Validity of 1st Order Markov Design

Mydesign is based on the Gilbert Model which is a first-order Markov model. In other words, the design benefits from the success/failure information obtained from the previous transmission to infer success of the following transmission. This raises a natural follow-up question: Can I even better estimate the success/failure of the next transmission, given the information on the last two, or even three transmissions? In other words, will a higher order Markov model help to further improve the quality of estimation? To this end, I analyze experimental data to show that the first order Markov model is the right choice; adopting higher order models will not make noticeable differences, while imposing additional complexity to the design and computation.

I achieve this by applying the concept of *entropy* from the information theory, which quantifies the uncertainty within a random variable as number of bits. Higher bit indicates higher uncertainty, which leads to lower estimation quality. Let X^n be a bernoulli random variable indicating reception success or failure at an arbitrary time instant, t_n. Then, the entropy within X^n is denoted as $H(X^n)$, indicating the marginal entropy. I further let X^{n-1} and X^{n-2} indicate reception success or failure at t_{n-1} and t_{n-2}, respectively (i.e., one and two time instants preceding X^n). Then, the conditional entropy expresses the amount of uncertainty in X^n when the value of X^{n-1} or both

X^{n-1} and X^{n-2} are known, denoted as $H(X^n|X^{n-1})$ and $H(X^n|X^{n-1}, X^{n-2})$. For brevity entropy computations are given in the appendix at the very end of the chapter.

Since success or failure can be expressed in a single bit, the maximum uncertainty is 1, when I do not have any clue on success or failure. Conversely, the uncertainty of 0 is when it can be perfectly estimated. Figure 3.8 shows marginal and conditional entropies in the face of two correlations. Figure 3.8(a) was obtained from experimental data of 5×10^5 consecutive packets over 50 links with packet interval of $6ms$, where experiment in Figure 3.8(b) was done similarly to that in Subchapter 3.2.1. While achieving the entropy of 0 is impossible due to natural randomness in links caused by shadowing and multipath fading, two clear observations can be made from the figures: (i) The gaps between conditional and marginal entropies indicate that the knowledge on previous transmission is helpful in estimating the success or failure of the following transmission, again verifying the existence of spatiotemporal and temporal correlations. (ii) $H(X^n|X^{n-1})$ and $H(X^n|X^{n-1}, X^{n-2})$ are very close in both figures, suggesting 1st order is as good as higher order design in the estimation quality, despite its lower complexity. This supports the effectiveness and sufficiency of my 1st order Markov-based design.

3.4 Main Design

Based on the observations in the previous subchapter I design cETX on top of the Gilbert Model. However, this model is only capable of representing the status of a single link. This means that the model is not enough to express spatiotemporal correlation, which is a relationship across multiple links. In this subchapter, I address this issue by proposing an extended model, from which my metric cETX is derived.

3.4.1 Design Overview

As an extension of ETX, cETX inherits the main idea of ETX to represent link quality by the expected number of transmissions. It is shown later in this subchapter that cETX also retains the additive property of ETX, such that the path cost is simply the sum of cETX. Nevertheless, unlike ETX which assumes the packet receptions are independent, I take both temporal and spatiotemporal correlations into account to more

accurately estimate the link/path cost. In fact, cETX is a generalized version of ETX, extended to incorporate knowledge on any degree of correlation into the link/path cost. For example, cETX simply reduces to ETX when there is no correlation among links.

3.4.2 Extended Gilbert Model

I propose an extended representation of the Gilbert Model for a few distinctive features that the original model does not possess. Specifically, it enables (i) the straightforward model extension to multi-hop scenarios, (ii) tracking the packet forwarding status to allow the computation of cETX directly from the model, and (iii) capturing spatiotemporal correlation among consecutive links in a path.

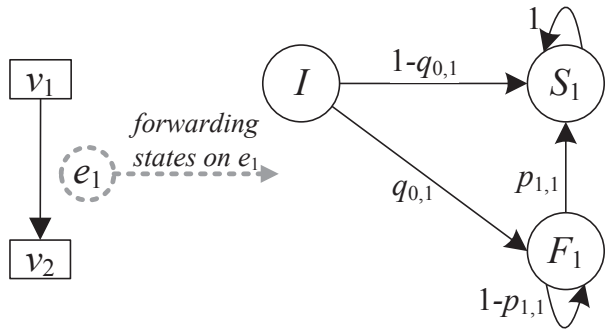


Figure 3.9: Single-hop network Gilbert Model representation

The Extended Gilbert Model describes the forwarding status of each packet in the form of 1^{st} order Markov chain. For ease of understanding I start from a single-hop network and its corresponding model depicted in Figure 3.9. The three states, S_1, F_1, and I, express packet forwarding on e_1. States S_1 and F_1 indicate the success and failure of the transmission over e_1, while I stands for the initial state indicating that the transmission has not yet taken place. Then, the packet forwarding process on the model follows intuitive steps: starting from state I, a transition to either S_1 or F_1 occurs depending on the transmission success or failure. Upon failures, retransmission attempts are made to transit from F_1 to S_1, until S_1 is reached.

From the definition of state I, it is clear that its initial probability is 1. Now let us describe the transition probability introduced in the model. Keeping notations consistent to the previous subchapters, let S_1^n and F_1^{n-1} denote the transmission success and failure over e_1 at time t_n and t_{n-1}, respectively. Then, $p_{1,1}$ denotes the probability of a

successful transmission after a failure in the previous time instant, $p_{1,1} = Pr(S_1^n | F_1^{n-1})$. Note that the subscript $_{1,1}$ used in $p_{1,1}$ denotes these two transmissions occur on the same link, e_1. Essentially $p_{1,1}$ represents the temporal correlation between consecutive transmissions within e_1. On the contrary, $q_{0,1}$ simply indicates the probability of failure in the initial transmission over e_1. Zero in the subscript implies that no previous transmission exists. Because I do not retransmit a packet that has already been received, the transition probability from S_1 to any other state is 0. In other words, the forwarding process over link e_1 terminates once S_1 has been reached.

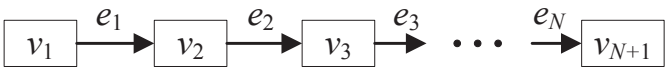


Figure 3.10: N-hop Linear Network

3.4.3 Multi-hop Gilbert Model

In the previous subchapter I proposed the extended Gilbert Model for a single-hop network. I now further expand the model to a multi-hop scenario depicted in Figure 3.10.

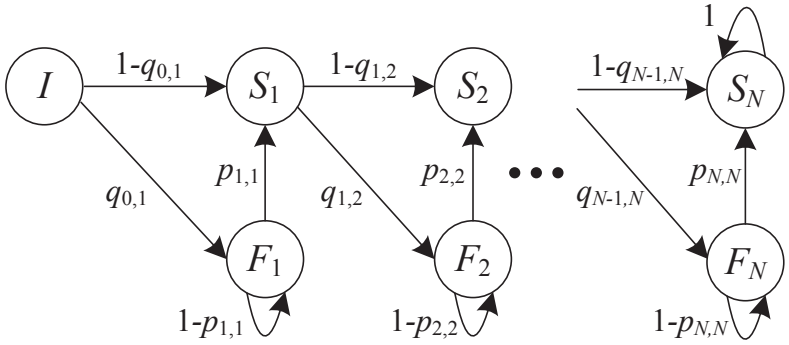


Figure 3.11: N-hop Gilbert Model

The multi-hop Gilbert Model for the corresponding network is given in Figure 3.11, where it is an extension from the model in Figure 3.9 to express multiple links. As shown in the figure, the extension is as simple as adding a pair of S and F states per link. Then, the spatiotemporal parameter is introduced between consecutive links. In the multi-hop path once a successful transmission takes place in the preceding link, an immediate transmission follows in the next link. For instance, v_2 starts its transmission over e_2

shortly after the transmission success from v_1 to v_2 over e_1. The dependency between these two consecutive events is spatiotemporal correlation, which can be represented as $q_{1,2} = Pr(F_2^n|S_1^{n-1})$ where the subscript $_{1,2}$ on $q_{1,2}$ denotes that success at previous time t_{n-1} occurred on e_1, where the following failure was on e_2. Generally, spatiotemporal dependency is considered between any two consecutive links, indicated as $q_{i-1,i}$ for $2 \leq i \leq N$ in the multi-hop Gilbert Model. Clearly the multi-hop model captures both temporal and spatiotemporal correlation via p and q parameters.

3.4.4 cETX Computation

I now show how mymetric cETX can be computed from the multi-hop model in Figure 3.11. Note that cETX over a link is defined as the estimated number of transmissions to deliver a packet over the link. Suppose I want to compute cETX for an arbitrary link, $cETX_{e_i}$. Transmission on e_i starts on state S_{i-1}, and ends when S_i has been reached (I and S_1 for e_1). Since the state transition occurs for every transmission, the cETX$_{e_i}$ value equals the expected number of transitions to reach state S_i from S_{i-1}. This is often referred to as the expected first passage time in Markov chain theory, which can be computed as follows:

$$cETX_{e_i} = 1 + q_{i-1,i} \times \frac{1}{p_{i,i}} \tag{3.2}$$

The equation can be intuitively understood by referring to my model. The first term in Eq.(3.2), i.e., 1, indicates the initial transmission which is mandatory. The second term indicates the product of the initial transmission failure probability, i.e., $q_{i-1,i}$, and the expected number of transmissions under the failure of the initial transmission, i.e., $\frac{1}{p_{i,i}}$. Therefore, the equation simply implies that a low cETX value is achievable where there are two conditions: (i) a small chance of the failure in the initial transmission, and (ii) a high chance of recovering from the failure in case the initial transmission fails.

Special Case: Eq.(3.2) shows that cETX is a function of spatiotemporal ($q_{i-1,i}$) and temporal ($p_{i,i}$) correlation parameters. However, the initial link e_1 is a special case as there is no link preceding it. In other words, e_1 has no link to be spatiotemporally correlated to it. Therefore $q_{0,1}$ is expressed sorely in terms of temporal correlation parameters, $p_{1,1}$ and $q_{1,1}$, to be the steady state probability of F_1. Please note that $q_{1,1}$

is found in the original Gilbert Model (q in Figure 3.7). Borrowing the steady state probability in Eq.(3.1) and rewriting with my notations, I have

$$q_{0,1} = \frac{q_{1,1}}{p_{1,1} + q_{1,1}} \tag{3.3}$$

The rationale behind this equation is as follows. The first transmission on the initial link is triggered at an arbitrary time instant, whenever the sender has a packet to transmit. This indicates that I need to estimate the link state at time t_n without the knowledge of the state at t_{n-1}. In this case, my best estimate becomes the steady state probability of S_1, which is simply the ratio of the time e_1 spent in S_1. Now, I can finally obtain $cETX_{e_1}$ by plugging in Eq.(3.3) to Eq.(3.2).

$$cETX_{e_1} = 1 + \frac{q_{1,1}}{(p_{1,1} + q_{1,1}) \cdot p_{1,1}} \tag{3.4}$$

Path cETX: my metric retains the additive property of ETX, indicating that path cETX is the sum of link cETXs in the path. This serves as the key that allows us to find the optimal path with cETX with appropriate modifications to traditional shortest path algorithms, which will be discussed in later part of the chapter. For a proof on the additive property, let us again consider the network and its model in Figures 3.10 and 3.11, where $cETX_E$ represents the cETX of the entire N-hop path. Then, $cETX_E$ is the expected first passage time from state I to S_N. Let the first passage time from state I to S_1 be denoted by a random variable Y_1, from S_1 to S_2 as Y_2, and so on until Y_N. Then, by definition $cETX_E$ is the expected value of the sum of all Y_i, i.e., $cETX_E = E[\Sigma_{i=1}^{N} Y_i]$. Due to the linearity of expectation and since $cETX_{e_i} = E[Y_i]$, I have

$$cETX_E = \sum_{i=1}^{n} cETX_{e_i} \tag{3.5}$$

Hence the additive property holds. I also note that cETX is indeed a generalized version of ETX, and it simply reduces to ETX under independent packet receptions. Proof on the relationship between cETX and ETX under different degrees of correlations is given in the appendix at the end of the chapter.

3.5 Implementation Issues

This subchapter describes how to obtain the parameters p and q for cETX computation, and shows *burst probing* enables precisely measuring the true performances of wireless links.

e_1	F	F	S	F	S	F	F	S	S	F	F	S
e_2	F	F	F	F	S	F	F	S	S	S	S	S
	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}

Figure 3.12: Parameter Computation Example

3.5.1 Parameter Computation

Among the parameters required by my design, those related to temporal correlation within a link are p and q. I again use S_n and F_n to denote the transmission success and failure at t_n. Let us compute the p and q values for e_2 in Figure 3.12. Maximum likely estimation for these parameters can be obtained simply by counting the number of transitions between success and failure, which turns out to be $p_{2,2} = Pr(S_2^n|F_2^{n-1}) = 1/3$ and $q_{2,2} = Pr(F_2^n|S_2^{n-1}) = 1/5$ for $0 < n \leq 11$. Similarly, the spatiotemporal correlation parameter when e_1 precedes e_2 is also computed as $q_{1,2} = Pr(F_2^n|S_1^{n-1}) = 1/2$.

3.5.2 Measurement in Practice

Probing refers to the technique of transmitting a small number of packets to support some fundamental functions including link quality measurement and neighbor discovery. Thus probes are commonly utilized in various state-of-the-art link metrics [29, 35, 36, 37]. Specifically, they adopt the classical periodic probing, where a single probe message is broadcast every fixed interval. However this mechanism suffers from limitations in measuring link performances; I observe that immediate retransmissions and relays induce a stream of packets within a short time interval to impose correlations between transmissions, which cannot be captured by periodic probes. To address this issue, I propose burst probing, where a batch of consecutive probes are sent at a short time interval. In the meantime, to keep the overhead at a reasonable level, the interval between probe bursts is set such that the number of probe packets per unit

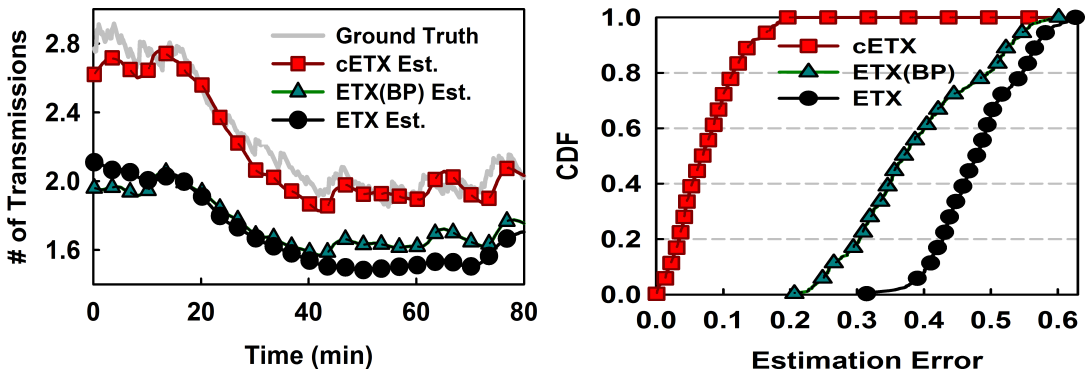
time is kept the same as that of periodic probing. This ensures the equivalent amount of overhead when cETX replaces ETX.

Unlike temporal parameters, measuring the spatiotemporal parameter via probing incurs additional overhead. This is because the spatiotemporal parameter for a link needs to be measured for every preceding link. To avoid such overhead, I obtain spatiotemporal parameters via the normal data traffic. Recall that the spatiotemporal correlation parameter is simply the failing probability of the initial transmission, just after the packet is received from the preceding link. This statistic can be easily collected; whenever a node receives a data packet, it records the preceding link, as well as the transmission success/failure of the initial transmission. This information is piggybacked on probes to be broadcast to its neighbors. To sum up, cETX assumes spatiotemporal independence under data traffic volume insufficient to obtain spatiotemporal parameter, where the accuracy of cETX nevertheless exceeds that of ETX due to the consideration of temporal correlation. Spatiotemporal correlation is reflected as the traffic increases, where multiple data flows in the network offer cETX with more opportunity to select better routes considering spatiotemporal correlation.

3.5.3 Overhead under Link Dynamics

I run an experiment to compare the link estimation performances of cETX and ETX in relation to the overhead they induce. To summarize, my result indicates estimation error for cETX is only 1/3 of ETX, at the cost of only 0.7% additional overhead. Experimental details are as follows: Burst probing in cETX occurs every 40s, where five consecutive probes are sent each time. ETX sends one probe every 8s. Both probing schemes induce overhead of 1/8 probe per second, indicating a fair comparison. To further maintain fairness, both metrics use the history of up to 90 probes, the value suggested by ETX [27]. In other words, when there is a data packet to be forwarded (every 10s in my case), both metrics use the history of the last 90 probes to estimate the number of (re)transmissions until a successful delivery. Then, the estimations are compared to the number of transmissions that actually occurs, to verify their accuracies. The 80-minute experiment consists of 11 MICAz nodes, v_1 through v_{11}. One node (v_1) serves as the sender while all others are receivers.

Figure 3.13(a) shows the result on a link between v_1 and v_8. The ground truth



(a) Estimation under dynamics (b) Error distribution

Figure 3.13: Link estimation with cETX

indicates the actual number of transmissions, while other curves are estimations based on probe history. Furthermore, Figure 3.13(b) exhibits the distribution of the estimation error (i.e., the absolute difference from the ground truth). I note that there are two versions of ETX in the figure, where one is the original ETX with periodic probing and the other is the modified ETX with burst probing. From the figures, I find that the estimate of cETX is far better than both versions of ETX. The average estimation error of cETX, ETX with burst probing, and original ETX in all ten links are 0.16, 0.42, and 0.48. The performance of ETX with burst probing sits in between cETX and the original ETX because it benefits from burst probing which mimics the data (re)transmission scenario, but it fails to accurately capture the correlation within. I note that the degree of cETX's advantage in accuracy applies to the link metrics built on top of ETX, such as 4B [35].

The overhead incurred by probes is approximately 3.2% of the total energy consumed. Since probing is adopted in the majority of the link metrics, its overhead should be amortized. On the other hand, exclusive overhead introduced by my design is the extra information piggybacked on probe packets. This takes up a tiny portion of about 0.7% of the total energy consumption.

3.5.4 Compatibility with Different MAC

The generality of cETX allows it to be applied to networks running various MAC protocols. This is because (i) both contention-based [38, 39, 40] and reservation-based [41, 42] MACs adopt a quick-retransmission policy for lost packet, which incurs temporally correlated, consecutive loss captured by cETX. For example CSMA commonly serves as the basis function for contention-based MACs, where the retransmission for a lost packet is made after a short duration of ACK timeout – 864us and 372us for ZigBee [43] and Wi-Fi [44], respectively. (ii) Spatiotemporal correlation captured via data traffic in cETX naturally reflects the forwarding scheme adopted by the network. That is, the corresponding degree of spatiotemporal correlation is captured regardless of immediate or delayed (e.g., duty cycled networks) packet forwarding. Immediate forwarding is likely to exhibit spatiotemporal correlation. On the other hand, when the delay is sufficiently large, cETX reflects the independence between consecutive links (i.e., no spatiotemporal correlation) where cETX still remains more accurate than ETX due to its consideration on temporal correlation.

3.6 Unicast/Broadcast over cETX

Replacing the popular metric of ETX, cETX can be generically applied to many protocols. This subchater presents how optimality is maintained in unicast protocols, followed by the benefit to broadcast schemes.

3.6.1 Application to Unicast

When spatiotemporal correlation is considered, the cETX value of a link is dependent on the preceding link. In this subchapter I demonstrate via an example, how optimal shortest path can be found in a distributed manner using distance vector with two-hop information.

Let us consider a simple four-node network in Figure 3.14, where $cETX(i, j)$ indicates cETX from node v_i to v_j. Initially, nodes attain local information about their two-hop neighbors by exchanging information between one-hop neighbors. Then the problem becomes: How cETX beyond two hops can be obtained through further iterative exchanges. That is, how the gray box in v_i's distance vector (i.e., $cETX(i, l)$)

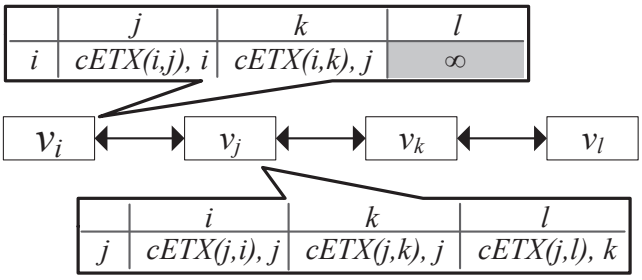


Figure 3.14: cETX in Unicast – Example Scenario

can be found. First, I note that $cETX(i,l) \neq cETX(i,j) + cETX(j,l)$, as $cETX(j,l)$ does not reflect the effect of correlation among the link between v_i and v_j and the link between v_j and v_k. Instead, $cETX(i,l)$ can be obtained via the below relationship:

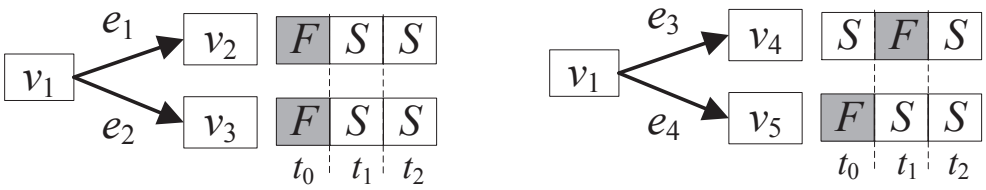
$$cETX(i,l) = cETX(i,k) + cETX(j,l) - cETX(j,k)$$

where last the term, $cETX(j,k)$, is the cost of link between v_j and v_k *not considering* the correlation with the link between v_i and v_j (i.e., the special case in Subchapter 3.4.4). Referring back to Figure 3.14, the above equation implies that v_i can obtain $cETX(i,l)$ via exchange of distance vector with its direct neighbor, v_j. In fact, the equation can be generally applied to finding cETX to any multiple hops (i.e., v_l of any hop distance from v_i), given that v_j and v_k are one and two-hop neighbors to v_i.

The case of convergecast: Support of unicast, the most fundamental data delivery mechanism, indicates cETX's applicability to a wide range of protocols whom unicast serves as their basis operation. Convergecast is one example, which consists of unicasts whose destinations are the sink. In convergecast the impact of spatiotemporal correlation is determined by immediate or delayed forwarding, where cETX inherently covers both the cases via its unicast support.

3.6.2 Application to Broadcast

This subchapter presents how cETX can help improve the performance of broadcast protocols. First, let's consider the positively correlated case in Figure 3.15(a). If v_1 starts broadcast at t_0, the two receivers will both receive the second packet at t_1, causing two transmissions to deliver the packet to both receivers. If the broadcast starts at t_1, the



(a) Positively correlated (b) Negatively correlated

Figure 3.15: Benefit of cETX in broadcast

two receivers immediately receive the packet, indicating that a single transmission is consumed. Similarly, at t_3, only one transmission is required. Therefore, an average of $(2+1+1)/3 = 1.25$ transmissions are needed for the network in Figure 3.15(a), while it is $(2+2+1)/3 = 1.66$ in Figure 3.15(b). Here I calculate the expected number of transmissions – $E(T)$ for a source node to reliably broadcast one packet to all the receivers with the consideration of temporal and spatial correlation using cETX. Without loss of generality, I assume that the link quality of the N receivers satisfies $Pr(S_1^n) \geq Pr(S_2^n) \geq \ldots \geq Pr(S_N^n)$. Since the node with a better link receives most of the packets earlier than the node with a worse link [45], I have

$$\begin{aligned} E(T) &= \frac{1}{Pr(S_1^n)} + \sum_{i=2}^{N} \frac{Pr(F_i^n | S_{i-1}^n)}{Pr(S_i^n)} \\ &= \sum_{i=1}^{N} cETX_{e_i} - N + 1 \end{aligned} \tag{3.6}$$

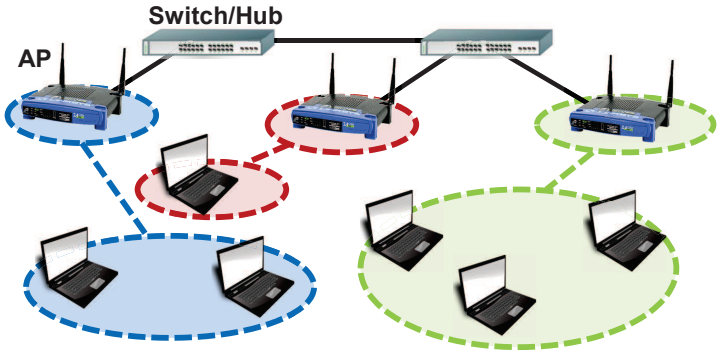
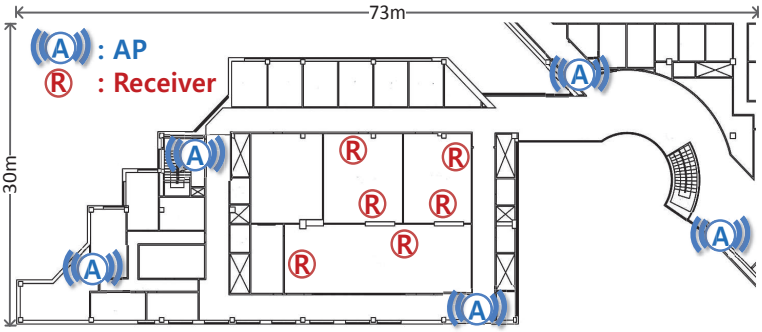


Figure 3.16: Collaborative broadcast in Wi-Fi

This lets cETX to be integrated within various broadcast protocols by selecting forwarder which minimizes $E(T)$.

Eq.(3.6) can be further applied to a novel communication paradigm for Wi-Fi networks, demonstrated in Figure 3.16. In collaborative broadcast multiple APs connected via wires share the same data to collaboratively deliver it to all users. Grouping and assigning users to APs are done via Eq.(3.6). Both ordinary and collaborative broadcasts are further described and evaluated in the next subchapter.

(a) ZigBee (lab)



(b) Wi-Fi (university building)

Figure 3.17: ZigBee and Wi-Fi Testbeds

3.7 Testbed Experimentation

In this subchapter I show that mydesign can be universally applied to various wireless technologies under different scenarios: I evaluate the performance of cETX in indoor

testbeds on ZigBee (802.15.4) and Wi-Fi (802.11) platforms, under unicast and broadcast scenarios with twelve protocols.

3.7.1 Baseline

Various representative studies in unicast and broadcast domains are implemented for cETX evaluation.

• **Unicast**: I adopt cETX in three representative unicast routing protocols. **(i) LQSR** [5]: Extends the DSR [46] protocol by taking unreliable links into account. **(ii) srcRR** [4]: Similar to LQSR, but without path switching to improve network robustness. **(iii) OLSR** [6]: Routing is executed only through a subset of nodes (i.e, Multi Point Relays) to reduce the control overhead. The three protocols offer different degrees of freedom in path selection to affect the benefit of cETX.

• **Broadcast**: I integrate cETX to nine classical broadcast protocols with diverse underlying network structures, showing the wide applicability of cETX. **(i) tree based**: C-Tree [7], **(ii) cluster based**: forwarding node cluster (Cluster) [13], **(iii) Multi-Point Relay**: MPR [12], **(iv) pruning based**: Self Pruning (SP [10]), Dominating Pruning (DP [10]), Partial Dominating Pruning (PDP [11]), and Total Dominating Pruning (TDP [11]), **(v) RNG based**: RNG relay subset (RNG [8]), and finally, **(vi) network coding based broadcast protocols**: CODEB [9].

3.7.2 Performance Metrics

The following performance metrics are adopted through my evaluation.

• **Number of Transmissions**: The average number of (re)transmissions to deliver one packet to the destination. Less number of transmissions indicate energy savings.

• **End-to-End Delivery Delay**: The average latency for a packet to be delivered to the destination.

• **End-to-End Delivery Ratio**: The success probability of a packet delivered end-to-end. That is, the number of (re)transmissions in each link in a path do not exceed the retransmission limit.

3.7.3 Experiment Setup

Experimental results in this subchapter are obtained from two physical testbeds, ZigBee and Wi-Fi, with below settings.

• **ZigBee**: I randomly deploy 22 MICAz motes running TinyOS as shown in Figure 3.17(a). Transmission power is set to be -25dBm to ensure multi-hop network, up to 5 hops. Node placement is kept the same for all evaluated metrics for fairness. Experiment for each metric lasts for two hours, during which an end-to-end packet delivery is made every four seconds, for a total of 1,800 deliveries per metric. CSMA is performed for each (re)transmission with random delay, where the average interval between retransmissions is approximately $6ms$. I limit the maximum number of retransmissions for each packet to be 7 (unless otherwise stated) following most practical systems to safeguard networks from being overwhelmed by retransmissions and collisions. Lastly, time stamps and sequence numbers of the delivered packets are logged in the nodes' flash, and uploaded to a PC at the end of the experiment.

• **Wi-Fi**: Depicted in Figure 3.17(b), five APs are installed in the corners of the floor, where I use laptops with the `Lorcon2` packet injection library [47] to play the role of AP generating downstream traffic. It is a reasonable approach since laptops/PCs are frequently used as APs in practice via software AP [48]. Six receivers are placed in three different rooms, separated by concrete walls. Although mainly 802.11g on channel 6 is used for its popularity, analyses later in the subchapter implies a similar benefit in 802.11b and 802.11n as well. A total of 10^5 packet transmissions are made in 40 minutes, with the power of 20dBm. Links with packet reception ratios from 10% to 95% are observed, where I commonly apply EWMA (Exponential Weighted Moving Average) on every metric. Control overhead of different probing (i.e., burst probing in cETX and ETX(BP), and periodic probing in ETX) is kept consistent for fairness.

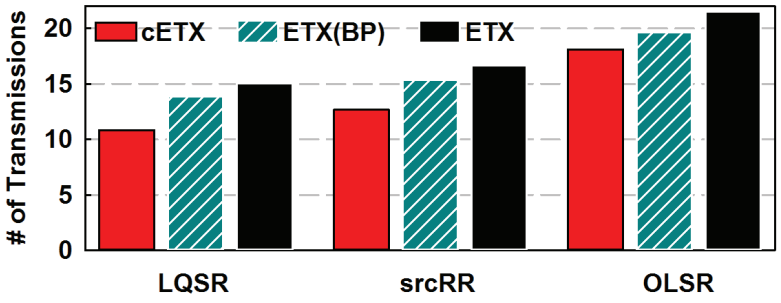
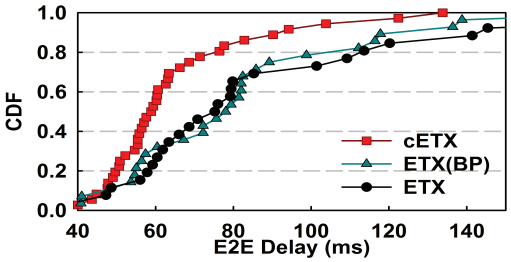
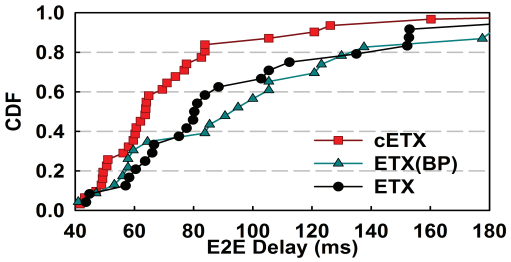


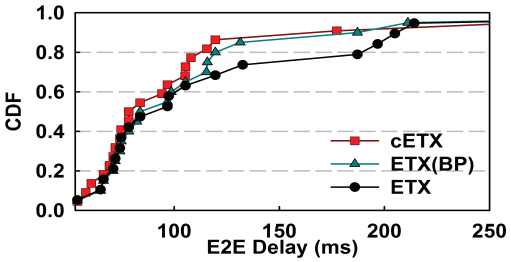
Figure 3.18: cETX in unicast



(a) LQSR

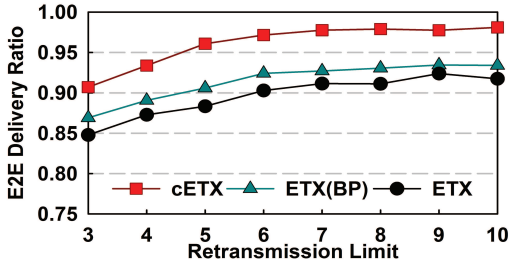


(b) srcRR

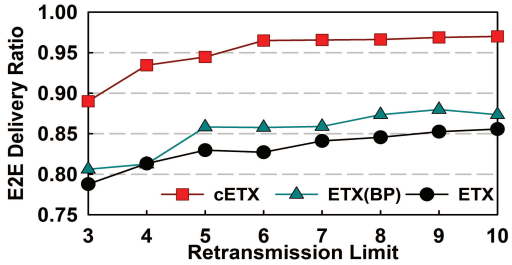


(c) OLSR

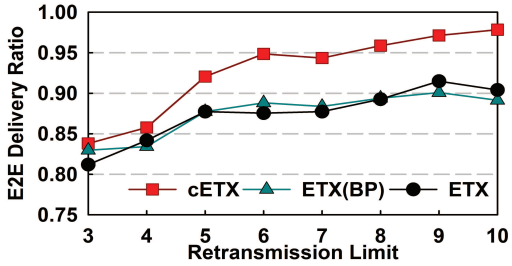
Figure 3.19: End-to-End Delay



(a) LQSR



(b) srcRR



(c) OLSR

Figure 3.20: End-to-End Delivery Ratio

3.7.4 Unicast

Figure 3.18 depicts the performance of cETX on unicast, obtained from the ZigBee testbed in Figure 3.17(a). Compared to ETX, cETX saves 27.8%, 24.1%, and 15.8% of transmissions for LQSR, srcRR, and OLSR. This demonstrates that cETX makes wiser routing decisions than ETX. Compared to ETX(BP), the percentage of savings becomes 21.6%, 17.5%, and 7.7%. This is because ETX(BP) benefits from burst probing for a slight performance enhancement over the original ETX. Furthermore, the advantage of cETX is the largest for LQSR and smallest for OLSR. The reason for this lies in the number of paths to choose from. While LQSR grants the most freedom in selecting paths, srcRR and OLSR restrict the frequent path switch to avoid high control overhead. Especially, OLSR only allows paths with MPR (Multi Point Relay) nodes. Figure 3.19 shows the end-to-end delay of metrics in the three protocols. cETX again exhibits the best performance, where the average reduction in delay ranged from 10.7% to 26.7%. Again LQSR obtains the biggest performance improvement due to the largest selection of paths.

• **Impact of Retransmission Limit:** Now I vary the limitation in the number of retransmissions from 3 to 10 and investigate its effect on the performance. Packet in delivery is dropped whenever the number of transmissions in a single link exceeds the limit, indicating failure in the end-to-end delivery. Note that exceeding the limit requires a link exhibiting continuous failures, which cETX precisely avoids to select. On the other hand, ETX blindly selects paths without considering this issue, leading to a worse delivery ratio. As shown in Figure 3.20, regardless of protocol cETX consistently achieves delivery ratios of over 94% for retransmission limit of 7 or higher.

3.7.5 Broadcast

The experimental results on nine classical reliable broadcast protocols, tested on ZigBee testbed, are shown in Figure 3.21. On average, the protocols need 20.5 transmissions to guarantee delivery of a packet to the entire network, while the number becomes 14.7 and 12.2 for ETX and cETX. Knowing that CODEB saves 31.6% of transmissions compared to the schemes without network coding, my design makes a further 21.4% improvement upon CODEB due to following reasons: a low link correlation may cause

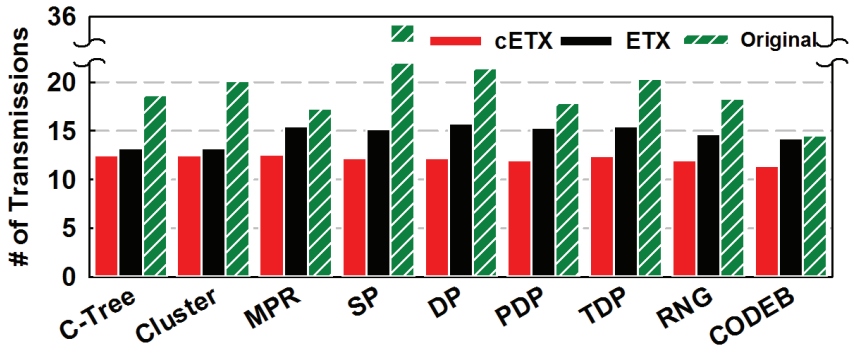
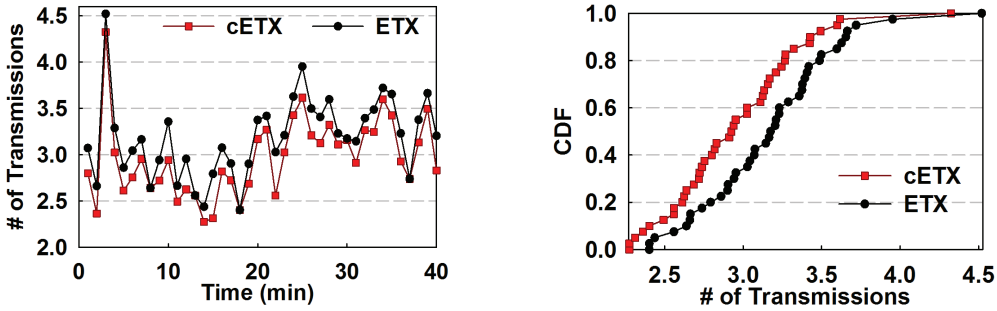


Figure 3.21: The Performance in Broadcast



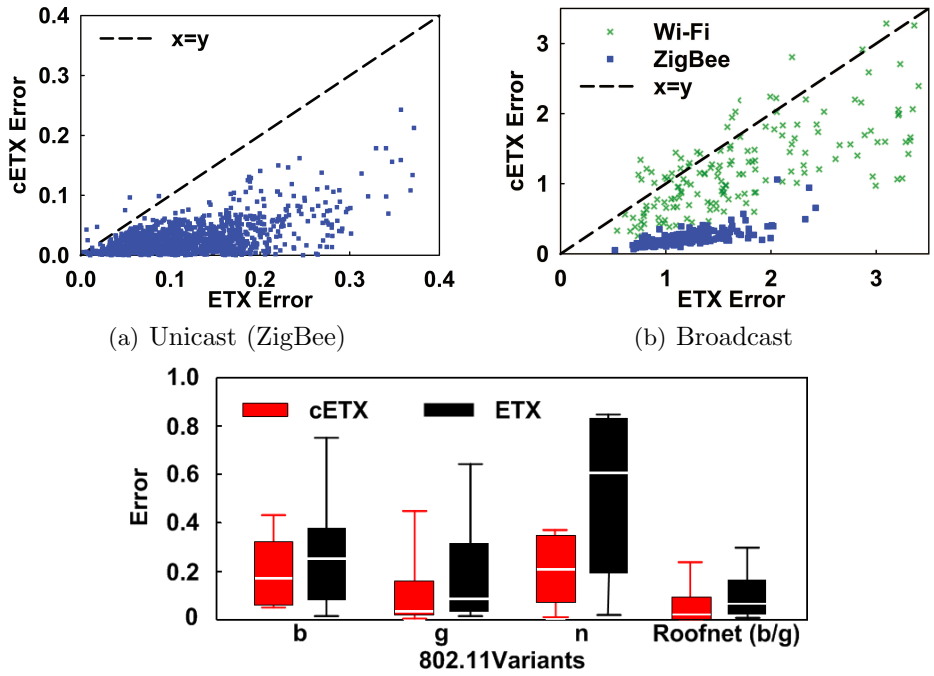
(a) Number of Transmissions

(b) Distribution

Figure 3.22: Collaborative broadcast performance

the nodes in a cluster to lose different packets, causing more retransmissions. With the consideration on spatial and temporal correlations, the protocols benefit from cETX to select forwarders with the minimum transmission cost (i.e., $min(\frac{E(T)}{N})$), thus forming clusters with high correlations.

Figure 3.22 shows the improvement achieved for collaborative broadcast on Wi-Fi testbed depicted in Figure 3.17(b). Five APs collaboratively deliver 10^5 packets to all six receivers. As cETX allows grouping highly correlated receivers to minimize retransmissions, its transmission cost is consistently less than or equal to that of ETX. The reduction can be as large as 34.1%.



(a) Unicast (ZigBee)

(b) Broadcast

(c) Unicast (Wi-Fi)

Figure 3.23: Interpretation error

3.7.6 Performance Insights

This subchapter provides analysis on *why* cETX performs better than ETX. Results on ZigBee platform are obtained from the 11-node experiment in subchapter 3.5.3. For Wi-Fi, experiments with six receivers and 2×10^5 packets are analyzed for each 802.11 variant: b, g, and n. I also analyze large-scale public Wi-Fi traces (5.6×10^5 packets from 560 links) from the MIT's Roofnet project [33] to validate the generality of my design.

•**Interpretation accuracy in unicast:** I first show the metrics' abilities to accurately interpret the probe reception history. That is, given the history, I observe how precisely a metric captures the performance of the corresponding link. This is done by comparing ETX and cETX values to the true value, introduced in subchapter 3.2.2. I use *error* as the measure of interpretation quality, which is the absolute difference between the metric and the true value. The result on ZigBee platform is depicted in Figure 3.23(a). Error for cETX is less than that of ETX for most of the time, where it is 70.2% less on

average. This is because ETX fails to capture highly correlated burst failures. This trend continues to Wi-Fi in Figure 3.23(c) regardless of its variants; ETX errors exceed those of cETX by 45.1%, 84.7%, and 160.3% in 802.11b, g, and n. Analysis on the Roofnet trace, which is a mixture of 802.11b and g, indicate the error is 49.6% larger in ETX. I note that error in Roofnet trace is generally smaller compared to my experiments; Roofnet is measured in outdoor deployment, where the degree of interference (i.e., the main cause of burst failures) is limited.

•**Interpretation accuracy in broadcast:** In both traditional and collaborative broadcasts, multiple receivers form a cluster in which members receive the packets from the designated sender. (i.e., APs) A smart grouping of receivers and assigning them with the right sender play a key role in determining the efficiency of the network. With the consideration of link correlation, cETX shows an improved accuracy compared to ETX in computing the expected number of transmissions required for a cluster for a given sender, as presented in Figure 3.23(b). Similarly to the unicast case, error is defined to be the absolute value of the difference of the true value and the values computed by cETX and ETX. The figure shows adopting cETX reduces the error by up to 67.6% in Wi-Fi where in ZigBee it is as high as 91.4%. This is because multiple low-power ZigBee links can easily be victims of cross-network interferences, inducing correlation among links. On average, cETX reduces the error by 21.5% and 75.3% in Wi-Fi and ZigBee, respectively. This advantage provides a rational explanation of the transmission (i.e., energy) saving achieved in the broadcast protocols shown earlier in the subchapter.

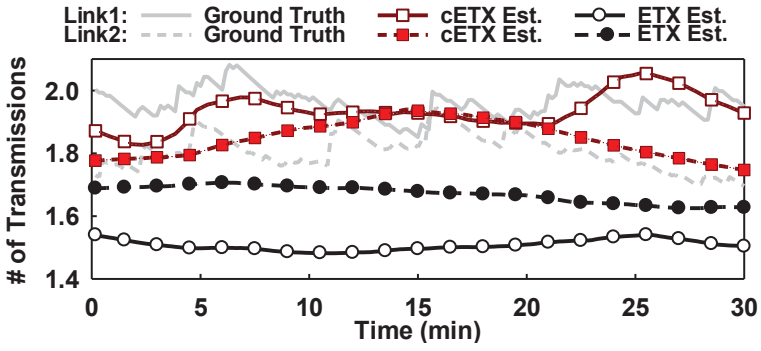
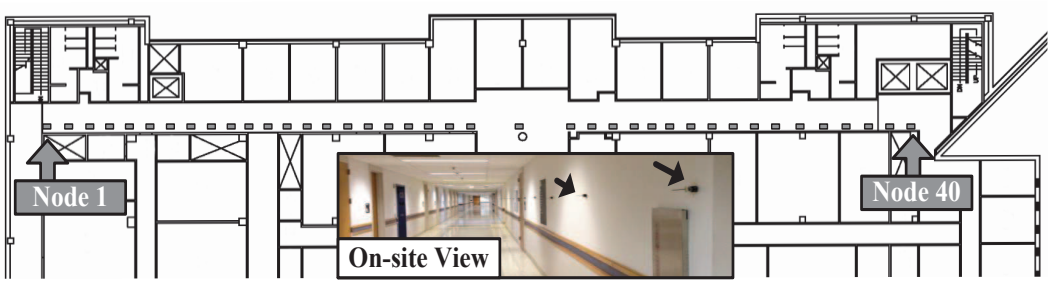
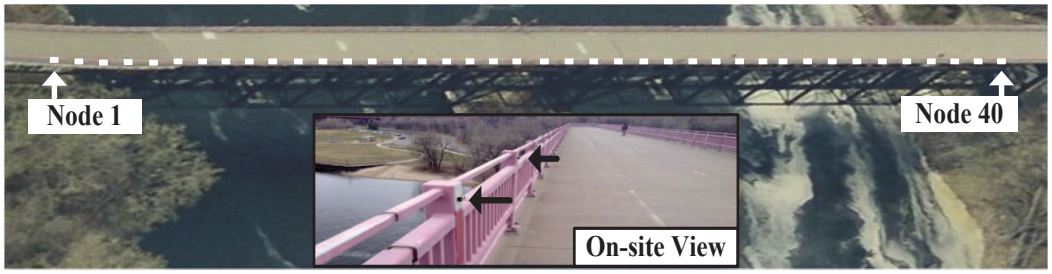


Figure 3.24: Insight into cETX performance

•**Estimation accuracy and route selection:** An inaccurate interpretation leads to



(a) Corridor



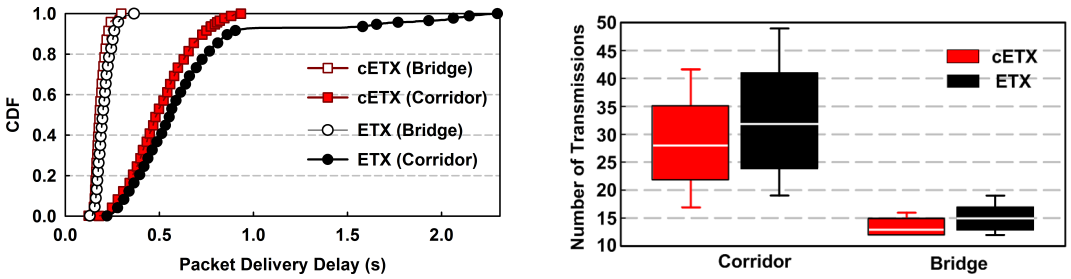
(b) Bridge

Figure 3.25: Indoor and outdoor experiment setup

an imprecise link quality estimation and further, a sub-optimal route selection. I show how this occurs in practice via Figure 3.24, showing the estimated and real number of transmissions on two distinct links for 30 minutes. The figure conveys two ideas: (i) cETX estimation is more precise and thus (ii) cETX makes better routing decision. When making a selection between links 1 and 2, ETX suggests link 1 should be chosen over link 2, due to the lower ETX value. However, the ground truth indicates otherwise, i.e., link 2 always has a better or similar (at 17^{th} min) performance than link 1. cETX is free from this problem, explaining the performance gain obtained in the unicast and broadcast algorithms in the earlier parts of the subchapter.

3.8 In-situ Deployment

In this subchapter I conduct experiments under practical indoor and outdoor settings, in the aim to give a better understanding on the impact of my design in real-world applications. Forty MICAz nodes are deployed in a corridor in a university building and



(a) Delay distribution (b) Number of transmissions

Figure 3.26: In-situ evaluation results

on a bridge crossing a river – common sites for popular applications including tracking and maintenance/safety monitoring – as shown in Figure 3.25. Nodes are deployed every $1.8m$ spanning $70m$ end to end. The transmission power is set to be -15dBm to assure multiple neighbors, where each node roughly has 10-12 neighbors (i.e, 5-6 neighbors to the node's right and left). Other parameters are kept the same as previous experiments. A thousand end-to-end (i.e., from node 1 to 40) packet deliveries are made, via LQSR embedding either ETX or cETX.

Results on delivery delay, shown in Figure 3.26(a), suggests that cETX's performance gain is more significant in the corridor (21.5% reduction) than on the bridge (8.9% reduction). This is due to the degree of wireless interference; In the university building hundreds of Wi-Fi users are connected to over fifty Wi-Fi access points, creating excessive amount of wireless interference. This causes burst failures in the experiment, forcing ETX to select poorly performing paths suffering from long delivery delays well above 1 second. Similar observation can be made in Figure 3.26(b). The average number of transmissions are 28.8 and 36.9 for cETX and ETX in the corridor, respectively, indicating 21.9% reduction. Due to the same reason as in the delay case, smaller degree of reduction (9%) is observed on the bridge, compared to what is achieved in the corridor.

3.9 Related Work

My work is built on top of two phenomena addressed in separate studies, namely, temporal and spatiotemporal correlation. I briefly summarize how the two phenomena

have been studied so far in relation to my work.

The impact of temporal correlation in ZigBee network, as well as in Wi-Fi, has been demonstrated clearly via extensive measurements and analyses in [49], where the authors propose a metric to quantify the degree of correlation. Furthermore, the works including BRE [50], 4C [36], and TALENT [37] exploit short-term dynamics based on packet overhearing to utilize intermittent and burst links. In RNP [29] and mETX [51], temporal properties of links were considered in routing, by counting consecutive failures and taking the variance of failure probability, respectively. However, the works in this category pays little or no attention to correlation across different links.

The comprehensive study on packet-level spatial correlation in [32] suggests a metric to predict performances of diversity-based protocols based on the intensity of correlation. To date, the phenomenon has been exploited in a wide range of wireless domains including ZigBee [52], Wi-Fi [32], and cellular networks [37]. For instance, it was utilized in network coding for efficient multicast [53] and broadcast [54]. Furthermore, Spatial correlation was studied in the presence of constructive interference to achieve fast data dissemination [55]. In this work I consider spatiotemporal correlation, which expands the effect of spatial correlation to the time domain. The unified design allows us to improve the unicast performance as well as the broadcast performance. In addition, the previous studies focus on particular protocol designs, while this work provides a generalized form of the widely-adopted ETX. My work can broadly improve a wide range of protocols.

3.10 Conclusion

My empirical study on temporal and spatiotemporal correlations in packet receptions shed light on the possibility of further improvement in network performance. In this work, I reveal that the most widely adopted link metric, ETX, fails to reflect the true link performance due to the ignorance of correlations. To address this issue, I propose cETX, a generalized version of ETX capturing not only temporal correlation but also spatiotemporal correlation. Because my design keeps the intuitive idea of ETX intact, I can simply replace ETX with cETX in existing protocols to obtain better routing performance. My evaluations on both ZigBee and Wi-Fi testbeds demonstrate that cETX

cuts down the metric error by 75.8% and 62.1%, and saves 22% and 37% communication cost for three unicast and nine broadcast protocols with only 0.7% additional overhead.

Appendix: Entropy Computation

Introduced in the *information theory*, entropy defines the the amount of uncertainty contained in a random variable. Let X^n be a bernoulli random variable indicating reception success or failure at time instant t_n. Then $H(X^n)$ represents the entropy within X^n, where it is computed as

$$H(X^n) = -\sum_{x^n \in \{succ, fail\}} Pr(x^n) log_2 Pr(x^n) \tag{7}$$

which is a marginal entropy as it only considers one random variable, without any given conditions. Furthermore, let X^{n-1} and X^{n-2} be bernoulli random variables for reception success or failure at one and two time instants prior to X^n. Then, the conditional entropy is the degree of uncertainty in X^n, when the value of X^{n-1} or both X^{n-1} and X^{n-2} are known a priori, denoted as $H(X^n|X^{n-1})$ and $H(X^n|X^{n-1}, X^{n-2})$. The conditional entropies are found by the following equations.

$$\begin{aligned} H(X^n|X^{n-1}) =& H(X^n, X^{n-1}) - H(X^{n-1}) \\ H(X^n|X^{n-1}, X^{n-2}) =& H(X^n, X^{n-1}, X^{n-2}) \\ & - H(X^n|X^{n-1}) - H(X^{n-2}) \end{aligned}$$

where the joint entropies, $H(X^n, X^{n-1})$ and $H(X^n, X^{n-1}, X^{n-2})$, are computed similarly to Eq.(7). That is, by replacing the marginal probability $Pr(x^n)$ by joint probabilities, $Pr(x^n, x^{n-1})$ and $Pr(x^n, x^{n-1}, x^{n-2})$.

Appendix: The Properties of cETX

In this appendix I provide the relationship between cETX and ETX, under various degrees of correlations.

• **Under independence:** cETX is a generalized version of ETX, where cETX simply reduces to ETX when packet receptions are independent. For simplicity, I omit the

subscript for the following parts of the subchapter. For an arbitrary link, spatiotemporal independence turns $cETX$ from Eq.(3.2) to the form of Eq.(3.4). Then, when transmissions are temporally independent,

$$Pr(S^n|F^{n-1}) = Pr(S^n|S^{n-1})$$

which leads to $p + q = 1$. By plugging in this condition within Eq.(3.4), I have

$$cETX = \frac{p+q}{p} = \frac{1}{Pr(S^n)} = ETX$$

where I used Eq.(3.3) and the rationale behind it. Thus, cETX is equivalent to ETX when transmissions are independent.

• **Effect of correlation:** Here I show how cETX is determined in relation to the correlation. For clarity, in direct comparison to ETX, I use cETX in Eq.(3.4). When link faces failure burst, the following condition holds:

$$Pr(F^n|F^{n-1}) > Pr(F^n|S^{n-1})$$

I thus have $p + q < 1$. Plugging in the condition within Eq.(3.4), I obtain

$$cETX = \frac{q}{(p+q)\cdot p} + 1 > \frac{q}{p} + 1 = ETX$$

Therefore, cETX is larger than ETX when the link experiences burst failure. In other words, ETX normally underestimates the transmission cost because of the ignorance of correlations.

Chapter 4

Broadcast under Coexistence

4.1 Introduction

Currently, most existing radio link status studies [56, 57, 58, 59, 60, 61, 62, 63] focusing on the individual link or path quality are based on the assumption that receptions among links are independent. However, the unprecedented popularity of wireless technology has created an emerging need for taking link correlation into consideration because of (i) the increasing cross-technology interference due to co-existence of different wireless technologies and (ii) the correlated shadowing introduced by dynamic environments.

Correlation of concurrent receptions at nearby receivers was experimentally revealed in recent works [64, 65], contradicting the popular assumption that wireless links are independent. Then, the significant impact of link correlation was demonstrated, where noticeable performance gain were achieved by incorporating correlation-aware designs into a wide range of applications including: flooding [66, 65], code dissemination [67], reliable broadcast [45], opportunistic routing [68], protocol performance analysis [64], and more.

Despite various representations of link correlation introduced in the works up to date, they all share the common idea of measuring link correlation at the link layer; correlation is solely determined by the similarity of reception success and failure patterns between links. For example, by representing reception success and failure as 0 and 1, measuring link correlation simply turns to be a comparison between two Bernoulli processes. Although the link level correlation measurement is simple, it provides limited information on the phenomenon due to the ignorance on the root cause.

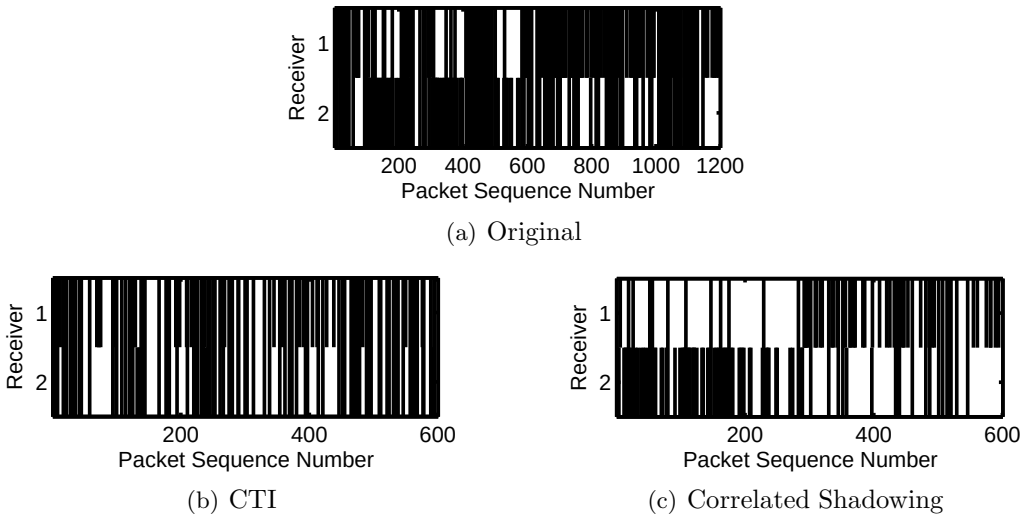
In this work I propose CorrModel, a framework that addresses the issues in the previous studies, to accurately capture and model link correlation. CorrMoe utilize SINR, which embeds fine-grained PHY layer information, to replace binary reception status to obtain high granularity measurement. I note that SINR is a well-known metric. Therefore, instead of reinventing the wheel, the focus of this work is in exploring the causes and intensities of link correlation using SINR, and showing how they are modeled. My contributions are three-fold:

- Although the phenomenon of link correlation has been mentioned in previous literatures, I provide the first extensive study to explore the root causes of link correlation.

- Through test-bed experiments consisting of 41 MICAz nodes, I show SINR's superiority in capturing correlations in comparison with RSSI (Received Signal Strength Indicator) and LQI (Link Quality Indicator). I then demonstrate how correlations are modeled via SINR for in-network use.

- I apply CorrModel to two popular routing protocols, opportunistic routing and network coding, on a test-bed with 15 MICAz nodes. Experiment results show that my design helps opportunistic routing and network coding to save about 25% and 15% transmissions on average.

The rest of the chapter is organized as follows: Subchapter 4.2 presents what motivated this work. Means to capturing the link correlation and distinguishing the root causes follows in subchapters 4.3 and 4.4. Subchapter 4.5 illustrates the link correlation model, where its performance is evaluated in Subchapter 4.6. Subchapter 4.7 demonstrates two applications. Related works are discussed in Subchapter 4.8. Finally, Subchapter 4.9 concludes the chapter.



(a) Original

(b) CTI

(c) Correlated Shadowing

Figure 4.1: Independent effect of two correlations

4.2 Motivation

In this subchapter I first introduce link correlation, where the root causes of the effect is experimentally revealed. Then, the importance of identifying the causes and their intensities, which the existing studies fall short on, is illustrated.

4.2.1 Existence and Causes of Link Correlation

Link correlation is a phenomenon where broadcast packet receptions among closely-positioned receivers are not independent. That is, successful packet reception at one receiver strongly indicates success (i.e., positive correlation) or failure (i.e., negative correlation) in nearby receivers. The existence of link correlation was reported and its effect were exploited in recent works [64, 45, 65]. However they do so without considering the underlying causes, hindering accurate measurement of correlation and thus limited utilization of the phenomenon. Here I introduce the two causes, which are empirically revealed in the following subchapter.

• **CTI (Cross-Technology Interference)**: Multiple wireless technologies used today share unlicensed band. For instance 802.11b/g/n and 802.15.4, all reside on the 2.4 GHz ISM band. This leads to CTI, as high-power wireless networks (e.g., Wi-Fi) introduce

destructive interference in low-power networks (e.g., ZigBee) [69] throughout a large region, causing correlated packet loss in multiple links simultaneously.

• **Correlated Shadowing**: Wireless signals suffer shadow fading caused by the presence of obstacles in the propagation path of the radio waves. This affects closely positioned receivers altogether, leading to correlation among them [70].

4.2.2 Impact of the Two Causes

I now demonstrate the two distinct causes of link correlation, i.e., CTI and correlated shadowing. Based on the experimental result I show that they ought to be treated separately in order to accurately capture correlation. I run an experiment with three MICAz nodes, equipped with IEEE 802.15.4 compliant TI CC2420 radio. Two receivers are placed 1.6m apart while the sender sits in between them. While the sender broadcasts 1,200 packets, the two receivers listen and record the sequence numbers of the received packets, which is downloaded to a PC at the end of the experiment for analysis.

This experiment has two states to isolate CTI from correlated shadowing, and vice versa. In state one, nodes tune to channel 26 to avoid CTI from Wi-Fi, while people walk freely between sender and receivers, causing shadow fading. On the other hand, state two uses channel 12 to take the effect of CTI into account, while no obstacles are allowed. The nodes periodically switch between the two states every 100 packets. The reception traces obtained from the two receivers is shown in Figure 4.1(a), where the correlation is hardly observed. However, the trend becomes much more clear when CTI and correlated shadowing are distinguished from each other, as shown in figures 4.1(b) and 4.1(c). In Figure 4.1(b) the two receivers are positively correlated via CTI, due to powerful Wi-Fi signal propagating throughout a long distance to corrupt packets at multiple 802.15.4 receivers. Negative shadowing correlation in Figure 4.1(c) is a result of obstacles that block either one of the two. I note that, This result not only shows the existence of correlation induced by CTI and correlated shadowing, but also offers insight on the importance of differentiating one cause from another for an accurate correlation measurement.

The walk-through example in Figure 4.2 clarifies that the lack of the capability of differentiating the two causes leads to incorrect correlation measurements. In the

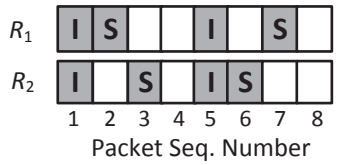


Figure 4.2: Example packet reception history

example I have two receivers, R_1 and R_2, receiving broadcast packets from a sender. The white boxes indicate successful receptions. The black boxes imply failures, where the letters indicate the cause; i.e., I and S stand for CTI and correlated shadowing. It is clear that receivers are perfectly positively correlated in CTI, and negatively correlated by correlated shadowing. However, treating CTI and correlated shadowing altogether falsely diagnose the receivers as independent. Let H_1 and H_2 denote R_1 and R_2's histories, where 1 indicates a success (white box) and 0 indicates a failure (black box). Then, I have $H_1 = \{0, 0, 1, 1, 0, 1, 0, 1\}$ and $H_2 = \{0, 1, 0, 1, 1, 1, 0, 0\}$. In this case, the correlation, computed by Pearson's rho, becomes $\rho = 0$, indicating independence.

4.2.3 The Need for High-granularity Measurement

Existing works on link correlation measures the phenomenon at the link layer, where they simply consider the relationship between the reception success or failure on different links, without taking the *quality* of the reception into account. Let us assume link correlation is to be measured between R_1 and R_2. Under existing approach, observing link correlation is simplified to measuring the chance of reception failure on R_2 given failure on R_1. Although simple and intuitive, this method lies on a rather strong assumption: all reception failures on R_1 implies the same chance of failure on R_2. However, this is often not the case in practice. The probability of failure on R_2 when R_1's reception has failed, is highly dependent on the intensity of the cause for the failure. For example, if failure on R_1 has occurred due to high-power CTI, the chance of R_2 facing a failure is high. Low-power CTI imposes lower chance of failure on R_2. The same applies to correlated shadowing. The degree of the cause can be obtained from the reception quality at R_1, from the PHY layer. In other words, link correlation should account for information from both link and PHY layer for accuracy.

4.3 Obtaining SINR

This subchapter shows how SINR is obtained. Computed for each packet based on the RSSI, I adopt an improved RSSI sampling to overcome the limitation present in the default sampling method.

4.3.1 Partial-sampling Problem

In 802.15.4-compliant radios (e.g., CC2420), RSSI is measured for the first 8 symbols, following the very beginning of a packet indicated by SFD (Start of the Frame Delimiter) [71]. Therefore RSSI only reflects a small initial portion of the packet, and is unable to detect any event that occurs in the following parts; an issue I call *the partial-sampling problem*. This imposes a significant limitation on accurately presenting the quality of a packet reception, especially when Wi-Fi is identified as the main source of CTI in low power networks [72]. Specifically, there exists a high possibility of partially corrupted packet in 802.15.4 networks, as Wi-Fi packets usually have much shorter in-air durations than those of 802.15.4. For instance, the duration of the maximum-sized packet in 802.11b and 802.11g networks are 1,906μs and 542μs, whereas 802.15.4 packets can span up to 4,256μs [69]. To address the partial-sampling problem, I directly access the RSSI register on the radio chip to acquire series of RSSI samples throughout the whole duration of a packet. I refer to this approach as *full-sampling*, which is discussed in detail in the following subchapter.

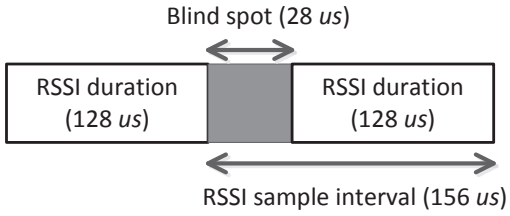


Figure 4.3: 6.4kHz RSSI Sampling

4.3.2 Full-sampling

Packet duration can be detected via SFD pin, where it rises upon beginning of a packet reception and falls at the end. RSSI series that cover the entire packet can be obtained by sampling RSSI values between SFD pin rise and fall at a high frequency. My 6.4kHz RSSI sampling rate is achieved using 32.768kHz watch crystal on MICAz's Atmega128 [73]. The rate is chosen to appropriately capture CTI induced by the minimal 802.11g packets. Referring to Figure 4.3, an RSSI sample is computed as an average during 8 symbols, which corresponds to 128μs. The 6.4kHz sampling rate indicates a sample interval of 156μs. Therefore, the duration of the *blind spot* where I cannot sense the interference comes to be only 28μs, which is sufficiently small to ensure capturing of a minimum-sized 802.11g packet (194μs) or even ACK (112μs).

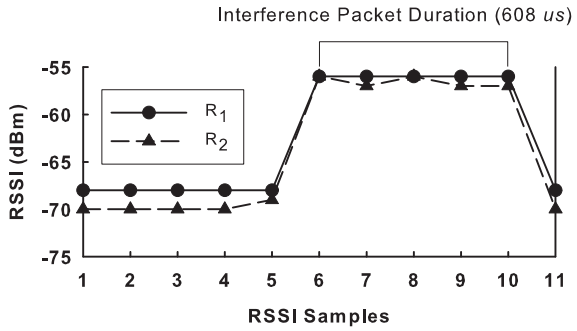


Figure 4.4: Full-sampling of RSSI to capturing CTI

The effectiveness of full sampling is validated via an experiment. A pair of closely-positioned MICAz receivers, R_1 and R_2, listen to a broadcast from a sender in the presence of a Wi-Fi interferer that generates CTI. Figure 4.4 depicts RSSI series obtained from the receivers. The figure demonstrates two properties of full sampling: (i) It is capable of capturing CTI in partially corrupted packets. It exhibits a steep increase for the duration of CTI, which is 608μs (456-byte PHY Protocol Data Unit at 6Mbps data rate) in the figure. (ii) Link correlation caused by CTI is clearly observable via similar trend on the two series. In fact, the packet did not pass CRC checks on both receivers, indicating a correlated packet loss.

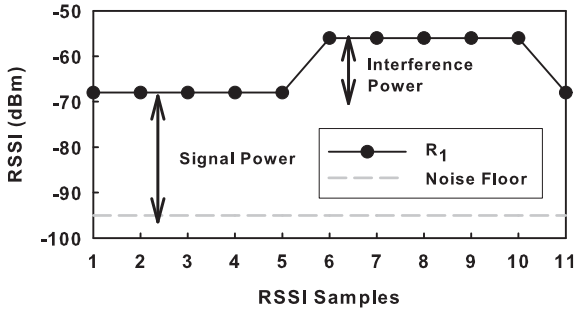


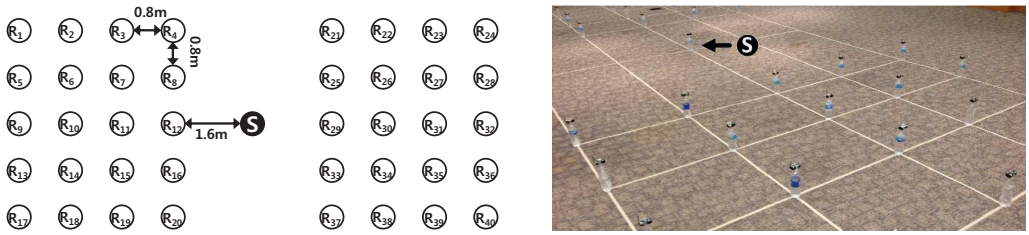
Figure 4.5: Acquiring parameters for SINR computation

4.3.3 Computing SINR

Now I describe how SINR can be computed from RSSI series obtained via full-sampling, with the help of the noise floor. Figure 4.5 shows R_1's RSSI series borrowed from Figure 4.4. The signal corresponds to the packet being received. Note that when CTI occurs, its power simply adds up on top of the signal. The noise floor is acquired by the RSSI value as soon as the link turns idle after packet reception, in order to best reflect the noise floor at the time of reception. Let $\mathbf{R} = \{r_1, r_2, ..., r_m\}$ indicate the RSSI series obtained during receiving a packet when r_i represents the i-th sample in the series. Finally, let r_n denote the noise floor level. Then, I can compute the power of signal p_s, noise p_n, and CTI p_i as follows:

$$\begin{aligned} p_n &= 10^{r_n/10} \\ p_s &= 10^{min(\mathbf{R})/10} - p_n \\ p_i &= 10^{max(\mathbf{R})/10} - p_s - p_n \end{aligned} \tag{4.1}$$

where $min(\mathbf{R})$ and $max(\mathbf{R})$ indicate the minimum and maximum elements in $\mathbf{R}$. The minimum value is used for the signal power simply to exclude the effect of CTI. The rationale behind taking the maximum value for The CTI power is to consider a part of the packet that is the most vulnerable to corruption. This is because 802.15.4 does not support correction codes, and thus an error in any part of the packet directly leads to packet loss. Then SINR in dB can be computed as:



(a) Experiment Setting (b) On-site view

Figure 4.6: Test-bed Deployment

$$SINR_{dB} = 10log_{10}\frac{p_s}{p_n + p_i}$$
$$= 10log_{10}\frac{10^{min(\mathbf{R})/10} - 10^{r_n/10}}{10^{max(\mathbf{R})/10} - 10^{min(\mathbf{R})/10} + 10^{r_n/10}}$$

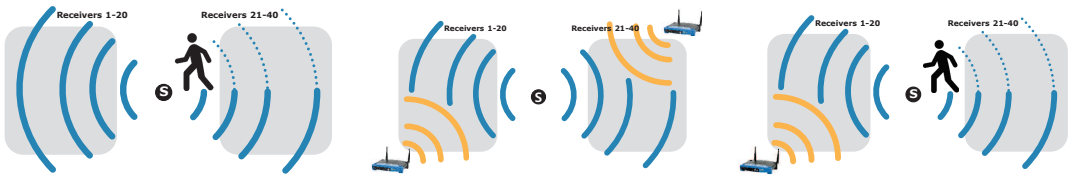
where the subscript $_{dB}$ is omitted from now on for simplicity. The strength of SINR lies in its capability to capture both correlated shadowing and CTI. Moreover, I note that the computation for SINR offers CTI power (i.e., p_i) as a byproduct. This allows us to distinguish the causes of correlation, which is another key point for accurate correlation measurement. Correlation capturing as well as differentiating the causes are further discussed in the next subchapter.

4.4 Performance of SINR

In this subchapter I experimentally demonstrate that SINR successfully captures both correlated shadowing and CTI. In comparison, I validate the performance of two well-known metrics, RSSI and LQI. I also show how SINR can express the distinct correlation trends with the help of CTI power, p_i.

4.4.1 Experimental Setup

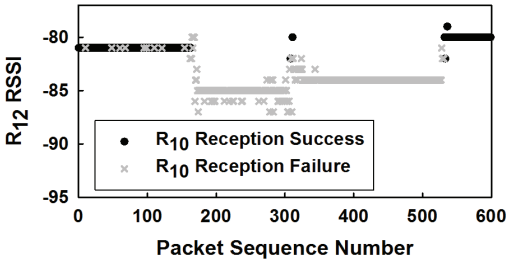
SINR, RSSI, and LQI are tested in three distinct experiments under correlated shadowing, CTI, and both. Each experiment consists of 41 MICAz nodes, where one is the sender and the rest are receivers. Deployment scenario depicted in Figure 4.6(a) is installed on an indoor test-bed as in Figure 4.6(b). Specifically, the test-bed takes



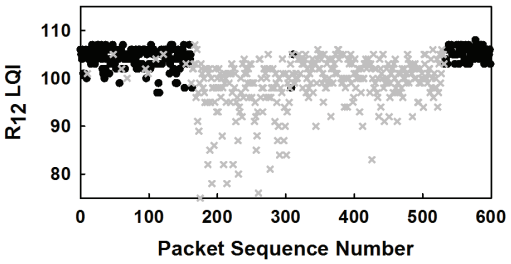
(a) Correlated Shadowing (b) CTI (c) Real-life

Figure 4.7: Experimental Scenarios

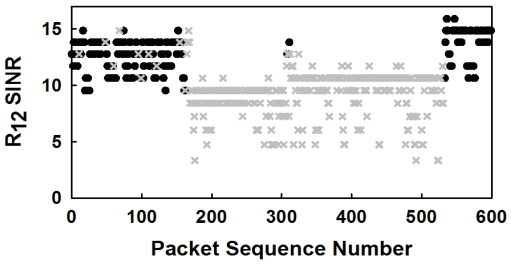
place in a large conference room in a university building where a high volume of Wi-Fi traffics (i.e., CTI) with more than 20 access points are present.



(a) RSSI

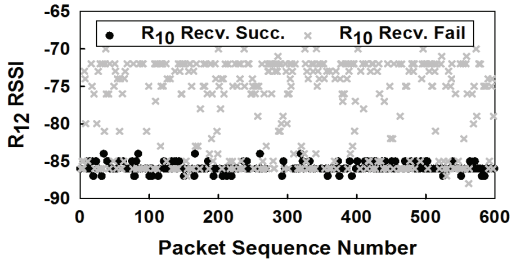


(b) LQI

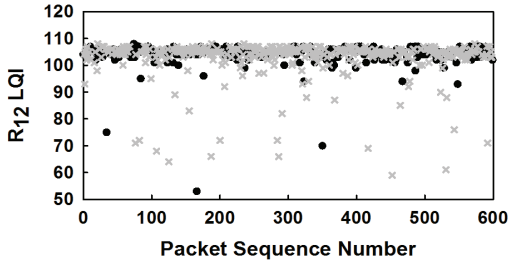


(c) SINR

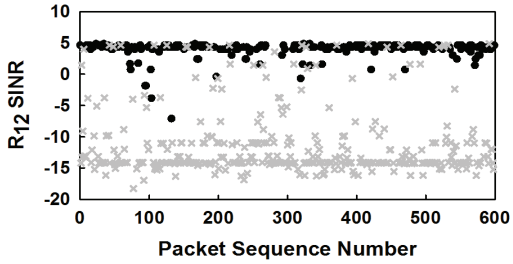
Figure 4.8: Capturing corr. shadowing



(a) RSSI



(b) LQI



(c) SINR

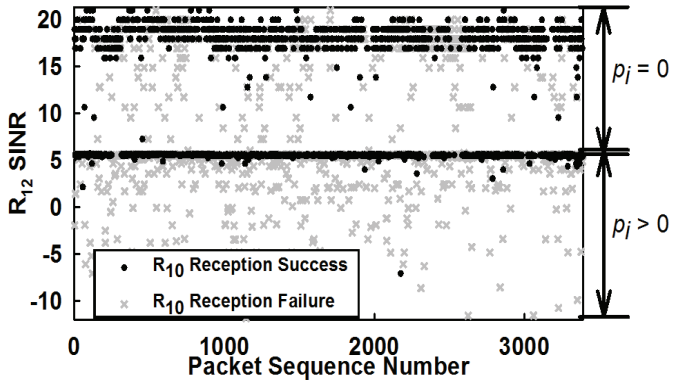
Figure 4.9: Capturing CTI correlation

In all three experiments, the transmission power of the sender is tuned to 0dBm. The sender periodically broadcasts 54-byte payload packet at a rate of a packet per second for an hour for the scenarios in figures 4.7(a) and 4.7(b), and for 4 hours for the case in Figure 4.7(c) (a total of 21,600 packets). In the correlated shadowing scenario depicted in Figure 4.7(a) I allow people to walk or stand around the sender to induce shadowing. I avoid the CTI by running the test-bed on channel 26, which does not overlap with Wi-Fi. I do the opposite in the second experiment in Figure 4.7(b) where I tune the test-bed to channel 12, which overlaps with Wi-Fi channel 1 to introduce CTI. Meanwhile, this time I avoid obstacles. Finally, in the last experiment I test the case where both CTI and correlated shadowing are present; the scenario where most of the real-life situations fall into.

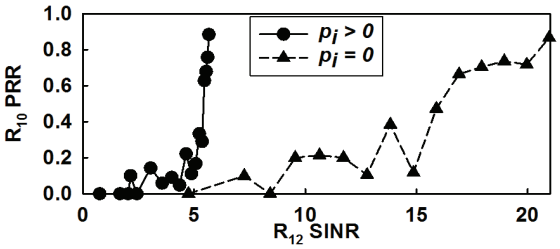
4.4.2 Capturing Correlation

Figure 4.8 shows the correlation between receivers 10 and 12 (i.e., R_{10} and R_{12}), which share the same LOS path from the sender. The three results for RSSI, LQI, and SINR are from the same series of broadcast packets for a 10-minute duration, to enable direct comparison of the performances. RSSI and SINR in figures 4.8(a) and 4.8(c) show a clear drop under correlated shadowing, between sequence numbers of 200 and 500 during which R_{10} experiences continuous packet reception failures. For example, when the RSSI of R_{12} is below to -80dBm, PRR (Packet Reception Ratio) on R_{10} drops to 28.6%. Similarly, for R_{12} SINR smaller than 13dB, R_{10} PRR is only 22.6%. This trend is also observable on LQI in Figure 4.8(b) but is not as clear compared to RSSI and SINR.

The limitation of RSSI and LQI is clearly revealed in Figure 4.9. In this result obtained from the CTI scenario in Figure 4.7(b), both metrics fail to show a clear relationship between R_{10} and R_{12} (figures 4.9(a) and 4.9(b)). This is because both RSSI and LQI suffer from partial-sampling problem, where only the first 8 symbols of the packet are considered. Partial-sampling brings fundamental limitation in detecting CTI, which often corrupt packets partially. Meanwhile, Figure 4.9(c) demonstrates that SINR successfully addresses this problem and effectively detects correlation under CTI: At R_{12} SINR above 2dB, R_{10} PRR is 90.5%, where otherwise is only 4.8%.



(a) R_{12} SINR vs R_{10} Reception



(b) R_{12} SINR vs R_{10} PRR

Figure 4.10: Correlation capturing under real-life scenario.

4.4.3 Distinguishing the Causes

Now that I know SINR is capable of detecting both correlated shadowing and CTI separately, I look into the real-life scenario in Figure 4.7(c) where both phenomena occur simultaneously. The result for 1 hour duration is depicted in Figure 4.10(a) in which two trends are observable, which can be distinguished via the value of p_i. Recall that p_i is a parameter in computing for SINR, representing CTI power; therefore $p_i = 0$ indicate CTI-free case (i.e, under shadowing) while $p_i > 0$ implies the existence of CTI. This essentially enables SINR to offer clear trends from the two independent phenomena as shown in Figure 4.10(b). Therefore, SINR is capable of capturing correlations in full under mixture of the two phenomena.

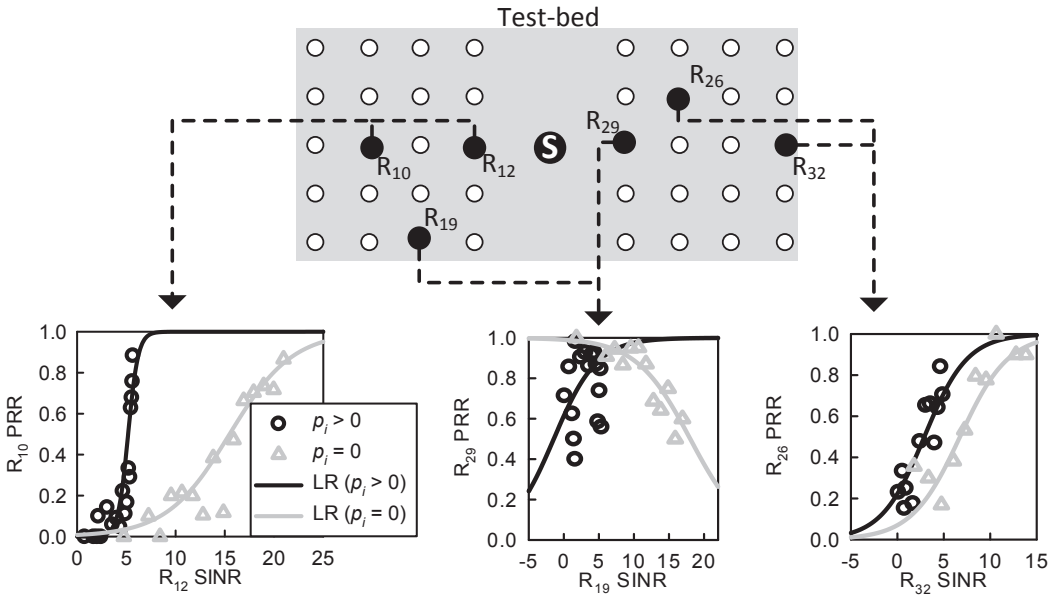


Figure 4.11: Test-bed logistic regression results

4.5 Link Correlation Model

This subchapter discusses how the effects of CTI and correlated shadowing are modeled with SINR, which we call CorrModel.

4.5.1 Applying Logistic Regression

As demonstrated in Figure 4.10(b), the relationship between SINR and PRR of two receivers induces two separate curves due to correlated shadowing and CTI. I adopt logistic regression to model each curve, a technique widely adopted especially in the field of machine learning. To offer a better idea on the technique, I first show the results in Figure 4.11, where the left-most plot in the figure shows the result when it is applied between R_{10} and R_{12} (cf. Figure 4.10(b)). Black curve indicate CTI correlation ($p_i > 0$), whereas correlated shadowing ($p_i = 0$) is presented in gray. The figure clearly shows that two logistic regression curves are very different, again justifying the need to separately model the two correlations. Note that R_{19} and R_{29} are negatively correlated by shadowing. This is because the two receivers are on the other side of the sender, where obstacle are only big enough to block one side of the sender.

Among various modeling techniques, logistic regression was selected due to its exclusive advantages for my purpose: (i) The y-axis of the model ranges from 0 to 1, which can be directly interpreted as probability. (i.e., PRR) (ii) Computational simplicity makes logistic regression feasible even under strict resource constraints, such as in sensor nodes [74]. Now I move on to how logistic regression is applied to express the relationship between R_A's PRR and R_B's SINR. Let Y be a Bernoulli random variable denoting reception success and failure on R_A, where 1 and 0 indicate reception success and failure, respectively. Next, let X be a 2×1 vector, $[X_0\ X_1]^T$, where X_0 is fixed to 1 (intercept term) and X_1 is SINR measured at R_B. Finally, according to logistic function, the success and failure probabilities of R_A given R_B's SINR is denoted as:

$$\begin{aligned} Pr(Y=1|X) &= \frac{1}{1+e^{-(\theta^T X)}} = h_\theta(X) \\ Pr(Y=0|X) &= \frac{e^{-(\theta^T X)}}{1+e^{-(\theta^T X)}} = 1 - h_\theta(X) \end{aligned} \tag{4.2}$$

where $\theta = [\theta_0\ \theta_1]^T$ indicates the parameters to be adjusted; that is, θ_1 is the slope of the function and the function shifts along x-axis according to $\frac{\theta_0}{\theta_1}$. Now, let $(X^i,\ Y^i)$ $1 \leq i \leq m$ indicate m training samples, where each sample is obtained from a packet. Then, training, or adjustment of θ, is done such that it maximizes the log likelihood below:

$$J(\theta) = \sum_{i=1}^{m} Y^i log(h_\theta(X^i)) + (1-Y^i) log(1-h_\theta(X^i)) \tag{4.3}$$

I apply Newton's method to solve for the maximum, where for each iteration θ is updated to

$$\theta \Leftarrow \theta - H^{-1} \nabla_\theta J \tag{4.4}$$

where H is Hessian matrix and $\nabla_\theta J$ is the gradient vector with respect to θ. They are

computed as

$$\begin{aligned} \nabla_\theta J &= \sum_{i=1}^{m} (Y^i - h_\theta(X^i))X^i \\ H &= \sum_{i=1}^{m} h_\theta(X^i)(h_\theta(X^i) - 1)(X^i)(X^i)^T \end{aligned} \tag{4.5}$$

4.5.2 Optimization

As shown in Eq. 4.3 and 4.4, logistic regression adjusts its parameter, θ, such that it closely matches the measured relationship between SINR of one node and PRR of the other (by maximizing log-likelihood). There are different optimization techniques that can achieve this, including batch gradient descent, SGD(stochastic gradient descent), and Newton's method. Among these I chose Newtons's method for several reasons, described in the following.

The main advantage of Newton's method is its converge speed. In my data, the method took less than 3 iterations to converge, sufficiently light to be adopted even on platforms with low computational power including wireless sensor nodes. SGD was also tested on the same dataset; it returned worse curve fit even after taking more than 30 iterations under step size of 0.1. Moreover, Newton's method does not require step size parameter, which is a considerable benefit in terms of adaptability and simplicity. Meanwhile, the main disadvantage of the technique is the heavy computational complexity in obtaining the inverse of H in Eq. 4.4. However, this is not a severe issue in my case, since H is only a square matrix of size 2×2.

I note that, at the first glance, techniques that allow on-line learning (e.g., SGD) might appeal as a better candidate, due to their advantage in accommodating newly obtained samples without the need to reconstruct the entire model. However, I argue that such on-line update techniques suffer from keeping the old samples intact, which degrades their performance in the face of dynamically changing correlation over time. This issue will be further discussed in Subchapter 4.6.4 when I examine the effect of temporal dynamics.

4.6 Model Evaluation

4.6.1 Baseline and Terminologies

My model is evaluated against *conditional reception*, or CR in short. Recent studies have shown that CR offers considerable amount of benefit in reception estimation [68, 66, 65]. For a brief idea on CR, let there be two receivers R_A and R_B that hear a broadcast packet from a sender. When the event of reception success at each receiver is indicated as S_A and S_B, $CR = Pr(S_A|S_B)$. Note that CR differs from $PRR = Pr(S_A)$ due to correlation between S_A and S_B. CR is a link-level metric, as it only uses the information on reception success or failure. On the other hand, my model, which I refer to as CM(correlation model), takes advantage of information from the PHY layer.

Denoting the dataset from which both CM and CR is computed as the *training set*, whereas the dataset on which estimation is performed as the *test set*, my model is evaluated in terms of accuracy, estimation performance, and the cost of maintaining estimation quality in the face of temporal link dynamics. The results in this subchapter is based on the data obtained from the real-life experiment shown in Subchapter 4.4, Figure 4.7(c). Specifically, since the estimation is very obvious for links with very high and very low PRRs, evaluation is performed among 18 links (i.e., $\binom{18}{2} = 153$ link pairs) that exhibit intermediate PRRs from 19.2% to 80.8%, with an average of 63.9%.

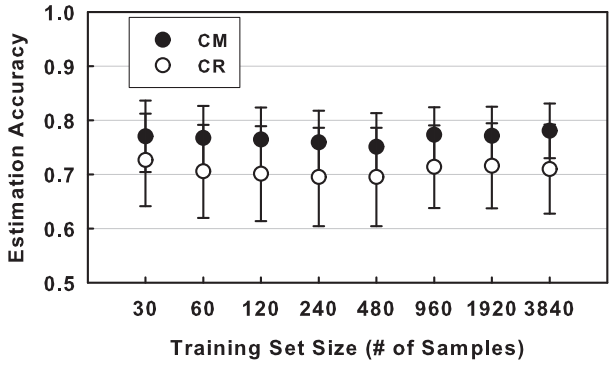


Figure 4.12: Model Accuracy

4.6.2 Accuracy Verification

I first show how accurately my model reflects the training set. That is, I verify the estimation performance of CM when the test set is equal to the training set. After constructing CM, estimation is done per-packet basis, by taking SINR of one packet at a receiver and probabilistically guessing if reception on the other receiver is a success or failure. Specifically, this is done in two steps: (i) Given a SINR sample, a receiver first finds the corresponding PRR of the correlated receiver via CM. (ii) Then, I estimate the reception to be a success when PRR$\geq$0.5 and failure otherwise. In other words, the logistic regression (i.e., CM) is used as a linear classifier, based on a simple rationale where a reception is more likely to be a success when PRR is above 0.5. For a fair comparison, CR is evaluated similarly to CM, where the reception is estimated to be a success when CR$\geq$0.5.

Figure. 4.12 shows the result with respect to different training set sizes. Error bars indicate 95% confidence intervals. The figure offers two observations: (i) The accuracy of CM is stable across various set sizes. This demonstrates the fast convergence of CM which only requires a small training set, cutting down the cost for model construction. Moreover, performance enhancement with large training set size indicate the model's robustness to link dynamics. (ii) There exist consistent gaps between the accuracies of CM and CR. This is especially notable knowing that CM does not require additional transmission overhead compared to CR.

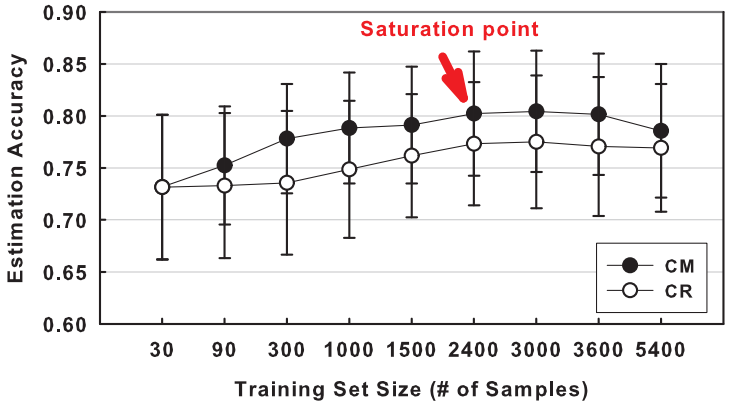


Figure 4.13: Estimation Performance

4.6.3 Estimation Performance

I now move on to the estimation performance when CM and CR are examined on the test set which is different from the training set. Specifically, I set test set to be a record of 3,600 samples (*i.e.*, 1 hr) *immediately following* the training set. Simply put, this evaluation shows whether the model built on the past history (training set) can be applied to well estimate the future (test set) for an hour. Note that this test is directly related to the performance of CM and CR when applied to real networks, where they most likely face temporal variance in links. In other words, the relationship between correlated receivers dynamically changes over time.

Figure 4.13 presents the result. It shows that, with a very small training set of 30 samples, the accuracy of CM and CR is approximately equivalent. Performance is enhanced on both techniques as the size grows, where CM improves at a faster rate than CR. The figure exposes two interesting observations. First, the performance of CM reaches the peak at training set size of 2,400 and is kept stable until 3,600. I call 2,400 samples training set as the *saturation point*, as obtaining additional samples does not help to improve performance beyond this point. In practice, the saturation point can be regarded as the maximum number of samples to be obtained. Moreover, this point also indicates the size of the buffer required to fully utilize my model to its maximum. I note that 2,400 samples only take 9.6KB of memory. This indicates that my model can easily fit into sensor networks; Off-the shelf sensor platforms, for instance MICAz and TelosB, are equipped with 512KB and 1MB flash storage. Another interesting observation is that CM's performance starts to degrade at the size of 5,400. This is an effect of time-varying property of links. As 5,400 samples indicate 90 minutes, samples obtained obtained earlier start to be outdated and force CM to make false decisions. The effect of time-varying link dynamics is further evaluated in the following subchapter.

4.6.4 Impact of Temporal Link Dynamics

Knowing that the saturation point indicates the most effective training set size, I explore the effect of temporal link dynamics on the performance of CM with the corresponding training set size. Figure 4.14 shows the trend of the accuracy for every 15 minute duration, beginning an hour after the model is initially constructed, until two hours

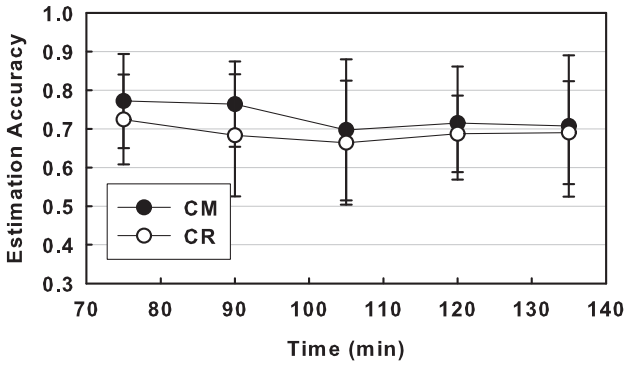


Figure 4.14: Effect of Temporal Link Dynamics

and 15 minutes. The figure clearly shows that the performance slowly degrades as the model gets outdated, agreeing with my previous observation. From practical standpoint, this figure suggests an optimal sampling frequency to maintain the performance to the maximum; if 2,400 samples can be collected during 90 minutes, CM with the maximum performance can be continuously rebuilt, without allowing performance degradation. This yields a moderate sampling rate of a sample (i.e., a packet reception) every 2.25 seconds.

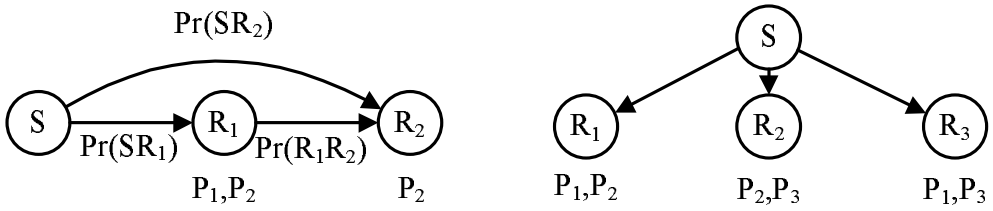
4.7 Applications

My model can be used to efficiently predict the nodes' all outgoing links' status, which provides us great opportunities to help a wide range of upper layer protocols obtain better performance. To illustrate its versatility, I show two applications on (i) opportunistic routing, and (ii) network coding.

4.7.1 Application Examples

Opportunistic Routing

Opportunistic routing protocols, such as ExOR [75], defer the selection of the next hop for a packet until these protocols have knowledge about the set of nodes that have actually received that packet. By doing so, opportunistic routing could utilize links



(a) Opportunistic routing (b) Network coding

Figure 4.15: Illustration examples

that become temporarily available. I use the example in Figure 4.15(a) to illustrate how opportunistic routing works. In this figure, three nodes form a two-hop network and the best shortest ETX path is "$S \rightarrow R_1 \rightarrow R_2$". With opportunistic routing, I can obtain a better results than ETX if node R_2 overhears the packets (e.g., P_2) from S.

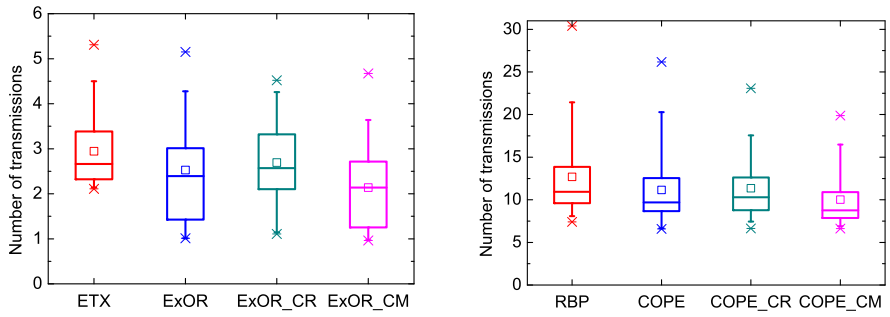
An issue here is how does the source node, S, finds out if the link "SR_2" is available or not. In opportunistic routing, node R_2 explicitly or implicitly sends back an ACK when it overhears packets from S. In reality, when opportunistic routing applies, the link quality of "SR_2", i.e., $Pr(SR_2)$ is not high (otherwise ETX will choose path "$S \rightarrow R_2$" instead of "$S \rightarrow R_1 \rightarrow R_2$"). Note that the reversed link is not symmetric especially when the link quality of the outgoing link is low [76], opportunistic routing loses many opportunities even if R_2 does overhear the packets from S. With my model, source node S can tell the reception status of the long distance link, e.g., SR_2 from a close node, e.g., R_1's reception information.

Network Coding

Network coding allows the nodes of a network to take several packets and combine them into one transmission, thus improving the throughput. For example, in COPE [77], as shown in Figure 4.15(b), the source node S broadcasts three packets, i.e., P_1, P_2, and P_3, to the three receivers. The first receiver R_1 loses packet P_3, the second receiver R_2 loses packet P_1, and the third receiver R_3 loses packet P_2. Instead of transmitting all the three lost packets, COPE sends an XORed packet $P_1 \oplus P_2 \oplus P_3$. Despite the fact that they lost different packets, all three receivers can retrieve the three original packets using the XORed packet.

Now the question is: how does the source node know what packets its neighbors

have? In COPE, it estimates the probability that a particular neighbor has a packet as the link quality. Here, my model can help COPE obtain more precise information on neighbor's possessed packets, thus offering more coding opportunities. In detail, one receiver will send back an ACK with its SINR value. Then, the source node computes all its neighbors' packet reception information with my model.



(a) Opportunistic routing (b) Network coding

Figure 4.16: Evaluation results

4.7.2 Performance Evaluation

To further confirm the analysis, I conduct two experiments with fifteen MICAz nodes which form a two-hop network. The testbed size is 8m by 2.5m. In the experiments, a control node is used to remotely configure the channel and transmission power. The default channel is 16 and the default power is -25dBm. Based on these radio settings, each node sends 10^4 packets in turn. Each packet is sent with a time interval of 0.3 second. The received packets's information, e.g., RSSI, is recorded in MICAz nodes flash memory. When the nodes finish sending packets, they report their packet reception information to a sink node which is connected to a PC. I thus obtain the information required by my model. Then, the corresponding nodes in the testbed are selected as forwarders for opportunistic routing or network coding.

In the opportunistic routing experiment, I compare the performance of ETX [57], ExOR [75], ExOR_CR, and ExOR_CM. In ExOR_CR, I use conditional probability to estimate whether node R_2 received a packet from S or not, given R_1's reception

status. In my method, i.e., ExOR_CM, node R_1 will send back an ACK including its SINR information to estimate the nearby links' receiving status. In the network coding experiment, I compare my design COPE_CM, with a traditional broadcast protocol, namely, RBP [78], and two network coding protocols, COPE [77] and COPE_CR.

Results on Opportunistic Routing

Figure 4.16(a) plots the number of transmissions for the source node to send a single packet to the destined receiver using different algorithms. On the average, the source node needs 3.0, 2.5, 2.7, and 2.1 transmissions for ETX, ExOR, ExOR_CR, and ExOR_CM, respectively, to deliver one packet to the destination. The three opportunistic routings perform better than ETX because the source node learns that the destination overhears the packets and sends the packets without passing through the relay node. ExOR_CM obtains a 30% improvement compared to ETX, which is better than ExOR and ExOR_CR. That is because my model provides a more precise information compared to ExOR and ExOR_CR in predicting the availability of the temporarily unstable links.

Results on Network Coding

Figure 4.16(b) shows the experiment results with RBP, COPE, COPE_CR, and COPE_CM. From the figure, the average number of transmissions for RBP to reliably broadcast one packet to the 14 receivers are 12.7, while COPE, COPE_CR, and COPE_CM is 11.2, 11.4, and 10.0 respectively. My model helps network coding obtain more benefits. Besides, I find that the gain of COPE is not very high in this scenario. The trace records show that this is because the broadcast packet loss is highly correlated; that is, the receivers lose the same packet and COPE has few opportunity to save transmissions.

4.8 Related Work

This chapter is related to link correlation and link quality estimation. Link correlation has been recently enlightened in the networking community. Previous work experimentally revealed the phenomenon and presented protocols that can effectively utilize it to

achieve high efficiency in flooding [66, 65], opportunistic routing [68], reliable broadcast [45], and code dissemination [67]. Another pioneering work on link correlation proposed a quantitative index to indicate the degree of the correlation [64]. My work is fundamentally different from the existing studies for the following three reasons: It is the first to (i) observe the phenomenon using PHY layer information and (ii) explore the two causes of link correlation. Lastly, (iii) instead of proposing a protocol, I focus on accurate capturing and modeling that can benefit a wide range of protocols.

Among a large body of works on link quality estimation, my work is most related to the recent studies that adopt logistic regression for their modeling. In [79], authors model link estimator taking PRR and PHY information including RSSI, LQI, and SNR. This work was improved in [74], where instantaneous prediction was made possible to avoid the need for off-line training. However, my work is inherently different from these studies as my work makes estimation across links, rather than within the same link.

4.9 Conclusion

To further improve network performance, this chapter aims to provide a framework named CorrModel that precisely captures the phenomenon of link correlation in reality, under both CTI and correlated shadowing. CorrModel utilizes SINR that is capable of capturing link correlation in full, while conventional metrics, RSSI and LQI, fail to do so. Using SINR I successfully model link correlation, where its estimation performance is consistently better than the conditional reception, which is the metric introduced in the in the state-of-the-art works [45, 65] as a means to measure correlation. Moreover, I show that CorrModel can be built and maintained with a moderate cost in the face of temporal dynamics of links. Lastly, my model was applied to opportunistic routing and network coding protocols for an average energy savings of 25% and 15%.

Chapter 5

Cross-technology Collaboration

5.1 Introduction

This chapter begins with the recognition that this coexistence is indeed double side: although it may cause inefficiency and unfairness in channel/spectrum utilization, it also provides new opportunities because the standards for individual technologies are specialized and hence possess strengths in different areas that are, often the weaknesses of the others. For example, while WiFi has access to a virtually unlimited amount of information via the Internet, it consumes a considerable amount of power, causing battery problems in mobile devices [80, 81]. Conversely, the ZigBee network often operates as a stand-alone and has limited information, but is extremely energy efficient. Thus, both networks can be enhanced via mutual supplementation, demonstrating the positive side of coexistence.

In this chapter, I propose FreeBee, a cross-technology communication framework that is generic and transparent (i.e., no extra traffic). My design aims to mitigate the detrimental effect of coexistence while exploring the opportunities behind it. As such, FreeBee sheds the light on the opportunities that cross-technology communication has to offer including, but not limited to, cross-technology interference mitigation and context-aware smart operation. Specifically I achieve this via embedding symbol into beacons by shifting their transmission timing. Although the concept of modulating via signal timings is known as PPM (Pulse Position Modulation), legacy PPM supports only communication between homogeneous devices and requires precise pulse timing, which can be hardly satisfied in wireless coexistence environments with mainly contention-based MACs.

Existing cross-technology communication works [82, 83] are technology-specific and require dedicated packets for communication, burdening already-crowded channels with further overhead. In contrast, Freebee utilizes mandatory beacons widely adopted among wireless technologies [84, 44, 43], achieving a generic and free-side-channel design. In summary, my original contribution is three-fold:

- I propose FreeBee, a novel cross-technology communication framework that allows direct communication between heterogeneous senders and receivers. In addition, FreeBee allows heterogeneous devices to receive broadcast simultaneously from a sender with overlapping frequencies (e.g., Bluetooth to WiFi and ZigBee) and

support a sender with a wider bandwidth (e.g., WiFi) to reach multiple narrower-band receivers (e.g., WiFi to multi-channel ZigBee).

- FreeBee requires no hardware modification and does not introduce dedicated traffic. Its existence is transparent to legacy wireless systems. My new *interval multiplexing scheme* supports concurrent transmission and reception of multiple signals.
- I present three prototype implementations: WiFi, ZigBee, and Bluetooth. Results suggest that FreeBee offers reliable symbol delivery within less than a second and supports mobility up to 30mph and duty cycle operation of under 5%. I also demonstrate a practical use of FreeBee: Inspecting real WiFi deployment patterns in a shopping mall area, FreeBee was found to save 78.9% of the energy otherwise wasted by the WiFi interface.

The chapter is organized as follows. Subchapter 5.2 presents motivations. The FreeBee design and features are introduced in Subchapters 5.3 and 5.4. Subchapters 5.5 and 5.6 evaluate performance analytically and empirically, followed by a FreeBee application in Subchapter 5.7. I summarize related work in Subchapter 5.8 and conclude in Subchapter 5.9.

5.2 Motivation

This subchapter demonstrates a wide range of benefits the FreeBee technology has to offer.

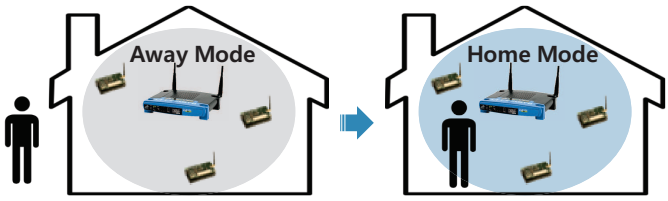


Figure 5.1: Context-aware home automation

• **Benefit to ZigBee – Smart homes**: FreeBee enables direct sharing of information without assistance from costly (>150 USD) dual-radio gateways that are mostly unavailable in real-life settings. One such type of information is user presence, which is accessible by WiFi AP by observing the nodes associated to it. Sharing this information enables other networks to provide context-aware service. Figure 5.1 demonstrates the example of a smart home with ZigBee-assisted appliances. Home WiFi AP first determines whether the resident is away or home (i.e., his/her smart phone is associated or not). Using FreeBee, this information is broadcast from the AP to all the ZigBee nodes inside the home to drive them to the appropriate operation mode. For instance, once the resident leaves they turn to "away mode", such as lowering home temperature to an energy-economic value.

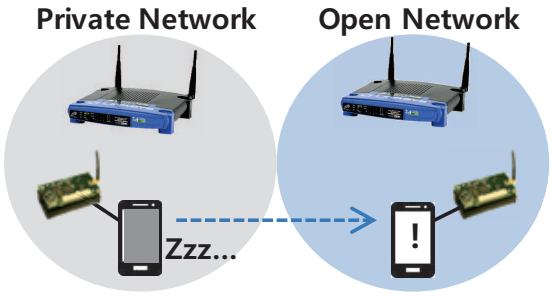


Figure 5.2: Wake on selective WLAN

• **Benefit to WiFi – Mobile devices**: Operating a WiFi Network Interface Card (NIC) continuously depletes precious energy in portable devices [80, 81]. To tackle this,

ZiFi [85] suggests attaching a low-power ZigBee radio to the device to wake up the WiFi NIC whenever it detects the existence of *any* WiFi AP. While this approach can significantly reduce standby energy, I believe that further savings can be achieved.

Nowadays most of metropolitan areas are overloaded with WiFi APs, many of which are private. Thus, it is still a waste of energy to blindly wake up a WiFi NIC when an arbitrary AP shows up, without knowing if it is accessible or not. As shown in Figure 5.2, I avoid this issue by embedding 1-bit accessibility information (i.e., open/private) into WiFi beacons and allowing the attached low-power radio (Bluetooth or ZigBee) to capture this information through FreeBee. Accordingly, the WiFi NIC can wake up only when it finds an open AP. I refer to this function as *wake on selective WLAN*. I note that this approach can be extended to limiting WiFi wakeup only to discovering an AP that fits the user's interest; for instance, when there is an AP that matches a user-defined SSID.

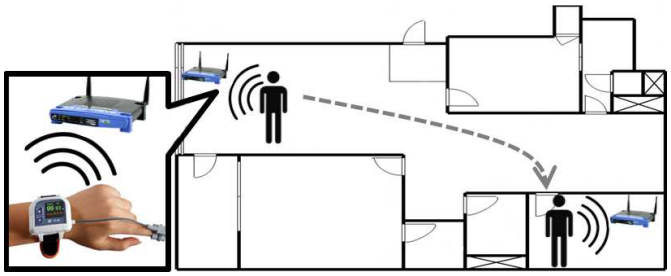


Figure 5.3: Real-time patient monitoring

• **Benefit to Bluetooth – Health care**: Taking advantage of its low-power operation, Bluetooth technology is widely used in portable medical devices, including glucose and heart monitors [86]. Although FreeBee is not designed to transfer a large volume of medical data, it enables health alerts by embedding urgent information into Bluetooth beacons. The ubiquitous WiFi coverage in most indoor environments today provides a continuous alert service even if patients are away from their Bluetooth-enabled medical station, as shown in Figure 5.3. This figure also shows that FreeBee can offer the location of the patient via geolocation provided by WiFi AP, allowing accurate and timely medical actions in case of an emergency.

• **Benefit to all – Channel efficiency**: All WiFi, ZigBee, and Bluetooth networks can

Figure 5.4: Cross-technology coordination

benefit from FreeBee via cross-technology channel coordination. As a result of evolving separately without considering each other, channel access schemes in heterogeneous wireless technologies are incompatible, leading to a severe CTI [87, 69, 83, 88]. This cross-technology channel access problem can be addressed by allowing explicit communication among different technologies. As demonstrated in Figure 5.4, FreeBee essentially allows TDMA or FDMA among heterogeneous wireless platforms, alleviating CTI. For instance, FreeBee realizes mechanisms similar to NAV (Network allocation vector) or RTS/CTS in WiFi for spectrum allocation that is global across technologies.

5.3 Framework Design

This subchapter presents an overview of FreeBee, followed by design specifics such as modulation and demodulation techniques. Without loss of generality, I use communication between WiFi (sender) and ZigBee (receiver) to illustrate the generic design of FreeBee.

5.3.1 Design Overview

I propose a cross-technology communication framework in which symbols are embedded into the timing of beacon frames. Specifically, I slightly shift the transmission time (advance or delay) of beacon frames, a configurable setting in most WiFi APs deployed today, simply via HTTP protocol. The shift is made in the units of 1.024ms, compliant to the 802.11 standard unit used in beacon scheduling, known as TBTT (Target Beacon Transmission Time) [44]. To ensure a free side-channel operation, FreeBee shifts timing in such a way that the average interval remains the same as the original setting. Thus, the proposed communication framework is not only transparent, but also does not consume additional bandwidth or energy. This unique aspect enables important

information to be broadcast continuously, safely reaching mobile and/or duty-cycled ZigBee receivers whose presence or active periods are a priori unknown to the sender.

However, WiFi beacons cannot be directly captured and recognized by ZigBee nodes, due to incompatible PHY layers. Instead, as WiFi coexists with ZigBee on the 2.4GHz ISM band, a ZigBee receiver statically detects the position of the WiFi beacons in the wireless channel, using its RSSI sensing capability. I note that, as a foundational function for MAC techniques including CSMA, RSSI sampling is common among varieties of wireless standards (e.g., Bluetooth).

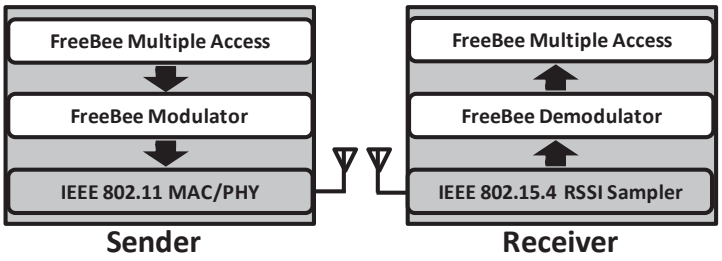


Figure 5.5: FreeBee architecture and scope

Figure 5.5 depicts the overall architecture and the scope of this work. My design spans modulation/demodulation techniques, as well as a multiple access scheme for concurrent communications. I note that the WiFi-ZigBee FreeBee design is based on 802.11 and 802.15.4 standards, and my design is compatible with all 802.11 variants (i.e., a/b/g/n) and 802.15.4-compliant nodes (e.g., TelosB and MICAz) and requires no hardware modification. In fact, FreeBee can be adopted to enable communication between any heterogeneous wireless platforms as long as the channel is shared. In the following, I first propose a basic version of FreeBee with the simplest form, followed by elaborated designs that enhance the basic version.

5.3.2 Basic FreeBee

In this subchapter, I describe the basic version of modulation/demodulation I refer to as *FreeBee*. This design assumes that the unmodulated position of the beacon is found during the initial network setup, which I refer to as the *reference position*. This position can be simply obtained by running FreeBee when the AP is sending beacons in their original timing. Modulated beacon positions found later are compared to the reference

position for symbol interpretation (i.e., demodulation).

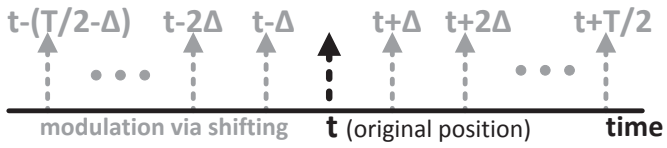


Figure 5.6: Symbol embedding with beacon shift

Modulation

For timely advertisement, the 802.11 standard requires APs to periodically broadcast beacons. FreeBee establishes a free side-channel by embedding symbols within the *transmission timing* of these mandatory packets. Referring to Figure 5.6, let's consider a beacon whose reference position is at t, where the interval is T. Applying FreeBee, I *shift* the beacon from its reference position in the range of $(-T/2, T/2]$ to indicate the symbol to be delivered. The amount of information that can be embedded is determined by T and the granularity of shift, indicated by Δ in Figure 5.6. I set Δ as $1.024ms$ following the beacon scheduling granularity as defined by the 802.11 standard (I note that the information amount can easily be increased by adopting smaller Δ). Under this setting the typical T of $102.4ms$, adopted in the majority of legacy WiFi APs, corresponds to 100Δ, indicating that the beacon can be positioned at 100 different time instances. Thus, beacon shift can express $\lfloor log_2 100 \rfloor = 6$ bits.

Due to incompatible PHY layers, the ZigBee receiver is unable to decode the beacons and thus cannot detect the presence of beacons directly. Therefore I statistically locate beacons by their periodic repetition. For instance, to deliver a FreeBee symbol corresponding to $t - \Delta$ in Figure 5.6, multiple consecutive beacons are shifted for the same amount (i.e., beacons are transmitted at $t + T - \Delta$, $t + 2T - \Delta$, and so on). The required number of beacon repetitions per symbol is decided by the channel noise, which is analyzed in detail in later subchapters. Lastly, I note that the beacon interval still remains at T in the process of FreeBee transmission, indicating free side-channel operation.

Demodulation

Here I describe how FreeBee captures and interprets position-modulated beacons for successful demodulation, especially under channel noise (i.e., other ongoing traffic within the same band). FreeBee demodulation starts from sampling the energy in the channel. This is done by consecutively recording the values obtained from the RSSI register on an 802.15.4-compliant RF chip on the ZigBee node. Upon recording a stream of RSSIs, the captured values are quantized to binaries–0 if below threshold and 1 if above–to indicate clear and busy channels. The threshold is set to be -75dBm following the CCA (Clear Channel Assessment) threshold for the 802.15.4 standard [43]. I note that WiFi also runs CCA, where the threshold is -82dBm [44]. For simplicity, I hereafter will refer to the binary value simply as RSSI. Furthermore, as a RSSI sample in ZigBee is a measurement spanning for $128us$, the sampling rate is set to be $1/128us = 7.8KHz$ to avoid time gaps between samples while keeping the rate at its minimum.

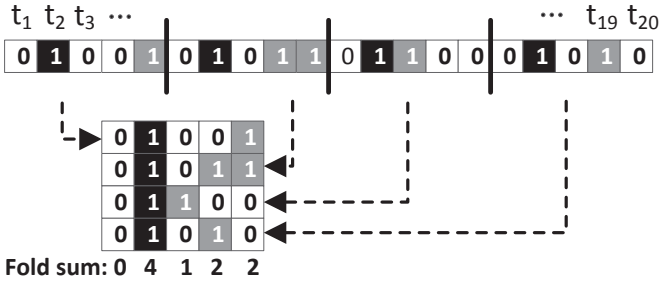
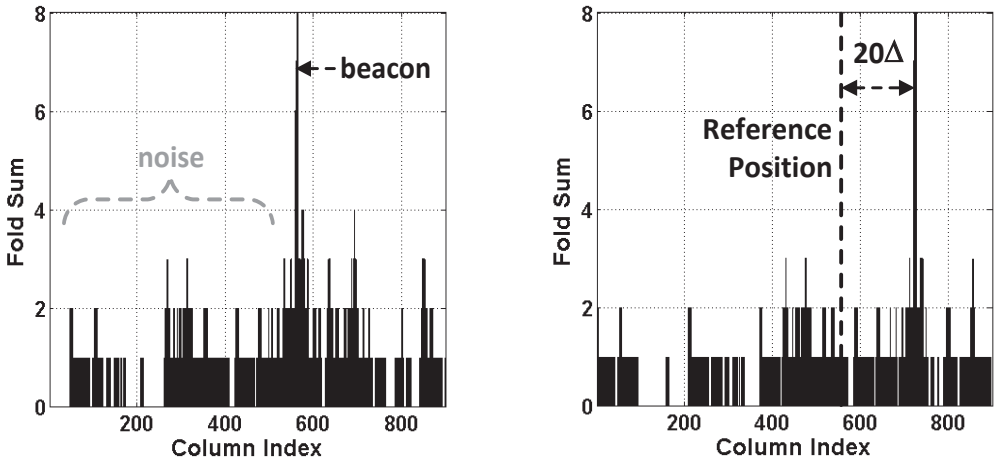


Figure 5.7: Folding example

I then apply *folding* to the obtained RSSI vector, a signal processing technique that allows detecting periodic signal under noise. I note that this technique was originally introduced in [89] and was recently featured by ZiFi [85] to detect the presence of WiFi AP. Given a sampled RSSI vector, folding by P simply cuts the vector into sub-vectors of equal lengths of P and stacks them to yield a matrix. An example of folding is shown in Figure 5.7, where a sampled RSSI vector of length 20 is considered. Let the interval of the beacons captured in the vector be T seconds, and the number of samples in T as λ In the example $\lambda = 5$, and upon folding by $P = \lambda$, RSSIs of beacons (in black) align in a column. the column-wise sum is referred to as the *fold sum*, where the column with the highest fold sum indicates the position of the beacon. Note that the fold sums

are likely to be smaller in other columns, as they are induced by either random (thus aperiodic) traffic or beacons with different intervals.



(a) Reference Position (b) Modulated Position

Figure 5.8: FreeBee demodulation in practice

Figure 5.8 presents an example of demodulating FreeBee symbol (20Δ) in practice. To sum up, FreeBee demodulation is process of finding the column corresponding to beacon position, which can easily be achieved by folding and simply picking the column with the maximum fold sum. This same process is used to learn the reference position of the beacon during network initialization and to find the modulated position. The difference of the two positions indicates the symbol within. Other harmonic analysis techniques, such as FFT (Fast Fourier Transform) and autocorrelation, do not yield the position (i.e., phase) of the beacon, and hence are not suitable for my purpose.

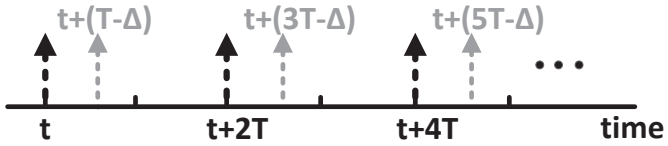


Figure 5.9: A-FreeBee modulation

5.3.3 Enhanced Feature #1: Asynchrony

The basic FreeBee design embeds the symbol as a beacon time shift from the reference position, a concept that requires learning the position beforehand. I relax this condition to introduce A-FreeBee (Asynchronous FreeBee), freeing my design from any prior knowledge to offer instantaneous communication.

Figure 5.9 shows Modulation of A-FreeBee. Applying A-FreeBee to beacons with an interval of T, beacons are scheduled to construct two streams of beacons (black and gray) with the same interval of $2T$, where one stream (i.e., gray beacons) is a shifted version of the other (i.e., black beacons). This is achieved by shifting *every other* beacon by the amount that corresponds to the symbol to be delivered. The figure demonstrates a case where the symbol corresponds to Δ, indicating one unit of shift. I note that A-FreeBee is also a free-side-channel, as the average interval between consecutive beacons is still T.

Under A-FreeBee design, the reference position is no longer required; it simply looks for two beacon streams with the same period by folding with $P = 2\lambda$. The embedded symbol is interpreted directly from the phase difference, i.e., Two columns with the first and second highest fold sums are found, where the distance between them indicates A-FreeBee symbol.

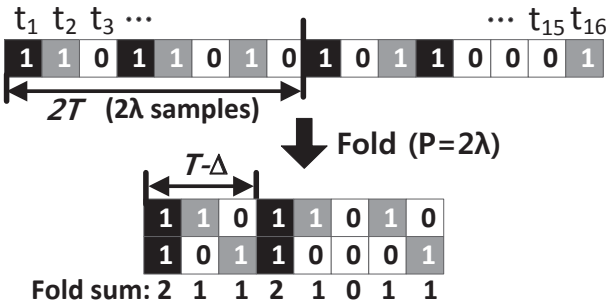


Figure 5.10: Folding example for A-FreeBee

A demodulation example of A-FreeBee is shown in Figure 5.10 for a RSSI vector of length 16. Two beacon streams are depicted in black boxes, and the gray boxes represent noise. Two columns with high fold sums are found by folding with $P = 2\lambda = 8$, where the distance between the two columns is $T - \Delta$. Noting that the distance would simply be T before modulation, the amount of shift (i.e., the symbol) is therefore Δ. Figure 5.11

below demonstrates A-FreeBee demodulation in practice, where the conveyed symbol is 20Δ.

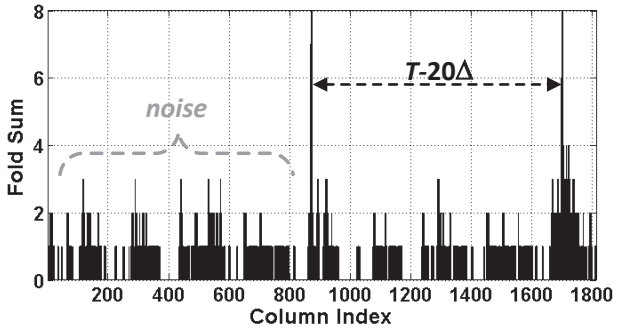


Figure 5.11: A-FreeBee demodulation

The asynchronous version features several advantages over the basic FreeBee, including that it (i) requires no synchronization, (ii) is robust to clock drifts, and (iii) supports instantaneous communication without prior knowledge. I note that all these improvements come with a trade-off in data rate; A-FreeBee, compared to the basic FreeBee, requires collecting a higher number of beacons to form *two* high fold sums instead of one. The relationship between the performances of FreeBee and A-FreeBee are analyzed both theoretically and empirically in later parts of the chapter.

5.3.4 Enhanced Feature #2: Concurrency

In the face of multiple (A-)FreeBee senders, selecting the same or arbitrary intervals may lead the signals to tangle and cause errors in demodulation. In this subchapter, I address this issue to allow *simultaneous transmissions* of an arbitrary number of (A-)FreeBee symbols such that each of them can be safely demodulated at the receiver.

Interval Multiplexing

Recall that, according to the 802.11 standard, beacon intervals are defined in the unit of $\Delta(=1.024ms)$. Let beacon intervals of n neighboring APs be $x_1\Delta$, $x_2\Delta$, ..., $x_n\Delta$. Then, FreeBee allows simultaneous communication of n APs if x_1, x_2, ... x_n are pair-wise co-primes. I refer to this as *interval multiplexing*.

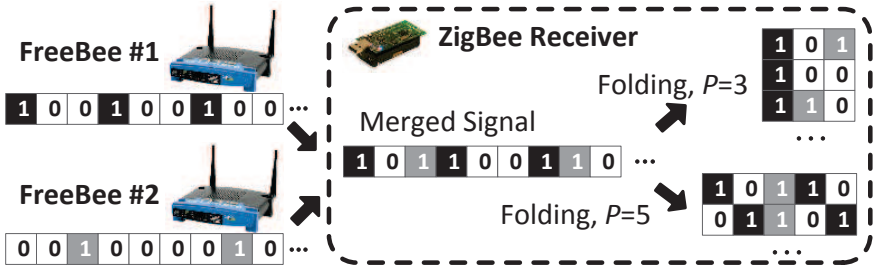


Figure 5.12: Demodulating interval-mutiplexed FreeBee symbols.

Figure 5.12 demonstrates a scenario of interval multiplexing/demultiplexing, where two FreeBee senders with intervals $T_1 = 3\Delta$ (in black) and $T_2 = 5\Delta$ (in gray) introduce a vector of merged RSSI signals at a receiver, and this receiver utilizes interval demultiplexing to demodulate. Specifically, by folding with $P = 3$ and 5 and looking for the column with the highest fold sum, the receiver can detect the position of the beacons as if the other signal does not exist. This is because 3 and 5 are co-primes, and no longer holds when they are not; for example, consider a sampled RSSI vector including beacons with intervals $T_1 = 2\Delta$ and $T_2 = 4\Delta$. When folded by $P = 4$, both beacons will form high fold sums, causing demodulation error. I note that while the figure shows only two senders for clarity, this idea can be extended to n senders as long as the intervals are pairwise co-prime. The rationale behind this scheme is given in the following proposition:

Proposition 5.3.1. *For FreeBee signals with co-prime intervals, folding for one signal restricts the fold sum of the others to a maximum of 1.*

Proof. Let $T_1 = x_1\Delta$ and $T_2 = x_2\Delta$ be two beacon intervals where x_1 and x_2 are co-primes. When a sampled RSSI vector containing beacons of interval T_2 is fold by $P = x_1$, beacon repeats in a column every LCM (least common multiple) of x_1 and x_2, which is $x_1 \times x_2$. Since the total length of the sampled RSSI vector is much smaller than $x_1 \times x_2$, beacons with the interval of $x_1\Delta$ cannot be folded into the same column when folded by x_2 (and vice versa), yielding the maximum fold sum of 1. □

The proposition states that the cross-interference between FreeBee signals is effectively suppressed when intervals are co-prime, essentially granting orthogonality between signals for concurrent communication. I note that this holds for both basic FreeBee and

A-FreeBee.

In practice, to avoid the overhead in computing for co-prime numbers, I allow APs to select among a set of pre-stored *prime* numbers instead. Moreover, the same interval should not be chosen among neighboring APs, which can be easily achieved in *listen before talk* fashion; they listen to each other to acquire the beacon interval information (i.e., x in $T = x\Delta$) that, according to the 802.11 standard, is recorded within beacons. After storing a set of intervals used by the neighboring APs, an unoccupied prime number is chosen as its interval. Conversely, coordination via wired connection (i.e., WLAN or the Internet) may be preferred, which avoids the hidden terminal problem.

Implicit Addressing Feature of FreeBee

This part discusses the unique addressing scheme used in FreeBee. As a reminder, demodulating each interval-multiplexed FreeBee signals remains the same as the case for a single signal; folding with P yields the corresponding FreeBee signal with P.

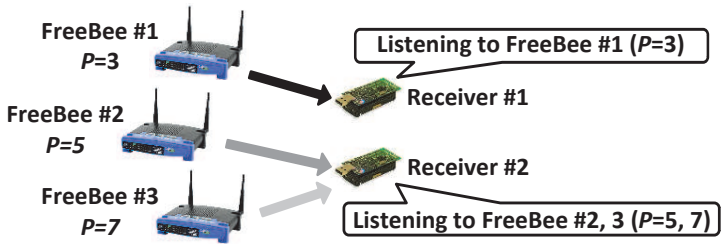


Figure 5.13: Implicit addressing via interval multiplexing

• **For stationary deployment**: As shown in Figure 5.13, each interval P is allocated to one freeBee sender, implicitly addressing that sender. Hence a receiver may select a subset of P's that corresponds to the sender(s) of interest.

• **For mobile deployment**: In mobile scenarios, mobile devices are not aware of Ps used by nearby FreeBee enabled devices. I need to fold for all Ps in the prime set to receive from every sender. This is not as computationally heavy as it may sound, as the number of primes, by the prime number theorem [90], is limited to $\frac{x_{max}}{ln(x_{max})} - \frac{x_{min}}{ln(x_{min})}$ when x_{max} and x_{min} are the maximum and minimum values in the prime set. For example, there are 20 primes in the interval range of $x_{min} = 53$ and $x_{max} = 149$.

From a practical point of view, I emphasize that implicit addressing is a unique

and effective feature of my design: as each sender is required to select different beacon interval, symbols demodulated with the same P are ensured to be from the same sender. This allows safely constructing a long symbol by appending received symbols. However, this is not the case for all other cross-technology techniques. In Esense [82] and HoWiES [88], a sender ID has to be embedded along with information in order to concatenate separate symbols correctly, leading to a large overhead in such a low-rate cross-technology communication.

5.4 Practical Issues

In this subchapter, I discuss practical issues and their effects on my design.

• **Channel access delay**: Although beacons are prioritized over data packets and hence queueing delays are negligible [44], they *do* suffer from channel access delays according to CSMA. This is in fact a challenge uniquely imposed on my design, which is different from traditional the PPM environment where all pulses are transmitted at their exact times. Upon long delays, a beacon may fail to contribute to folding. This is precisely why beacon repetitions are needed (e.g., four beacons in Figure 5.7) for statistical performance guarantee. My empirical study in Subchapter 5.6.3 suggests that 5 beacons yield an error of less than 1%.

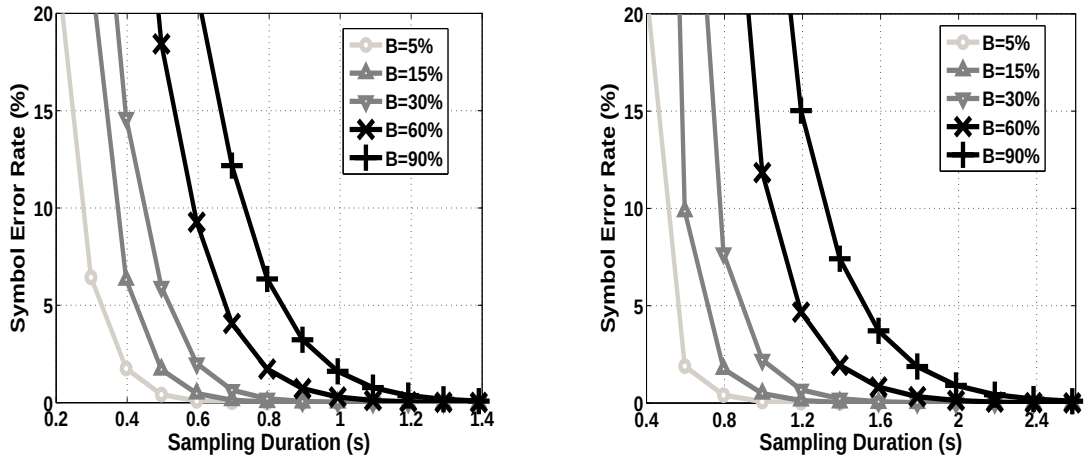
• **Noise**: Any non-beacon signal occupying the spectrum serves as noise and is a potential source of error. That is, as frequent 1's due to noise fill up the sampled RSSI, there is an increased chance of a large fold sum formed elsewhere to the beacon position, thus inducing demodulation failure. In other words, the performance of my design is enhanced by reducing the noise. This is simply done by taking only the first two RSSI samples for *any* packet including beacons and discarding (i.e., set to 0) the rest. The reason behind this approach is two-fold: (i) As data packets tend to be much longer than beacons, this reduces noise to 1/6 on average in my experiments. (ii) My empirical analysis indicates that the channel access delays of beacons are mostly (>90%) less than 256us, where a similar result was reported in a recent study [91]. Noting that the duration of a RSSI sample is 128us, this suggests the first two RSSI samples (256us) of beacons maintain a high chance of overlapping (i.e., contributing to fold sum) upon folding.

5.5 Analytics

In this subchapter, I provide a theoretical analysis of the performance of FreeBee and A-FreeBee.

5.5.1 SER vs. Sampling Duration

As noted earlier, the symbol error occurs upon demodulation failure, essentially when the fold sum of noise is higher than that of the beacons. For brevity, I defer a detailed SER (Symbol Error Rate) derivation to the appendix (at the end of the chapter) and here move directly to the results to demonstrate the impact of three system parameters: (i) T, the beacon interval. (i) ρ, the number of beacon repetitions for statistical demodulation; and (iii) Sampling duration, the sampling time to obtain a symbol, which is $\rho \times T$ for FreeBee and $\rho \times 2T$ for A-FreeBee.



(a) FreeBee (b) A-FreeBee

Figure 5.14: SER vs. sampling duration

Figures 5.14(a) and 5.14(b) show Symbol Error Rate (SER) when default beacon interval is set as $T = 97\Delta = 99.3ms$. These figures convey three ideas: (i) longer sampling (i.e., higher ρ) lowers SER, as more beacons are utilized to fight against noise; (ii) for a given duration, FreeBee achieves a lower SER than A-FreeBee; and (iii) higher channel occupancy, denoted by B, indicates more noise, thus higher SER. The figures have B up to 90% for completeness of analysis, where $B \leq 30\%$ was observed

in my experiment (under the threshold of -75dBm) in a university building with 50+ APs. Similar observations were reported in a recent study from a large-scale open WiFi trace [92, 85]. The figures suggest analytically that FreeBee and A-FreeBee achieve SER<1% for durations of $0.7s$ and $1.2s$ under such condition. I note this analytical result matches well with results from empirical experiments shown in Subchapter 5.6.3.

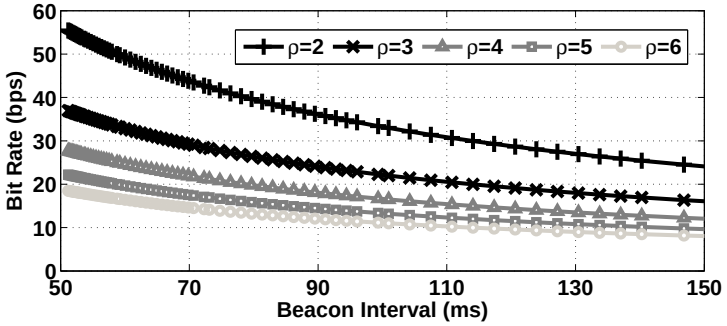


Figure 5.15: Bit rate vs. T for different ρ.

5.5.2 Transmission Rate vs. Beacon Interval

Beacon interval, T, is another factor that can affect (A-)FreeBee performance. The impact of beacon interval T can be observed with bit rate R. Intuitively, enlarging the beacon interval has two effects: (i) it offers more space for shift, or equivalently, yields more bits per symbol; and (ii) it requires more time to reach the same ρ. The bit rate for FreeBee can be computed as below:

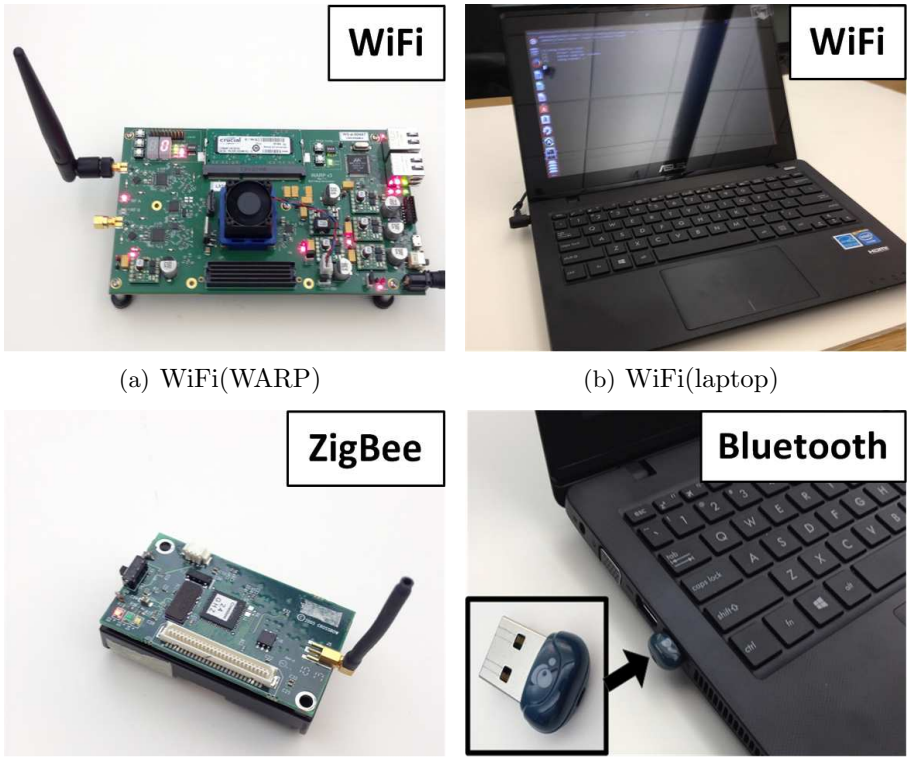
$$R = \frac{log_2 T/\Delta}{T \times \rho}\, bps \tag{5.1}$$

Noting that Δ (1.024ms in WiFi) defines the granularity of shift, the numerator in the Equ. 5.1 implies *bit per symbol*. Figure 5.15 shows the impact of beacon interval T on R in different scenarios for the range of practical intervals. In A-FreeBee, the rate is cut in half as it takes double sample duration (i.e., $\rho \times 2T$) to convey a same symbol. It is important to note that the rate given here is per sender, without bandwidth consumption (i.e., without incurring extra traffic). Due to interval multiplexing, the aggregated

throughput linearly increases according to the number of senders. Furthermore, boosting the throughput of a single sender by adopting additional beacons is also a viable option. Performances under such cases are shown via experimental evaluations in the next subchapter.

5.6 Performance Evaluation

I show the generality of (A-)FreeBee by evaluations performed across four platforms operating on three different wireless standards: WiFi, ZigBee, and Bluetooth.



(a) WiFi(WARP) (b) WiFi(laptop)

(c) ZigBee (d) Bluetooth

Figure 5.16: Four evaluation platforms

Table 5.1: Experiment Settings

Communication Direction	Tx Ch.	Tx Power	Rx Ch.	Dist.
WiFi → ZigBee	1,4	13 dBm	11-15	8m
ZigBee → WiFi	11-14	0 dBm	1	1.5m
Bluetooth → WiFi	37-39	4 dBm	4	3m
Bluetooth → ZigBee	37-39	4 dBm	15	3m

5.6.1 Experiment Settings

The experiment parameters are specified in Table 5.1, where the detailed settings are as follows.

• **Design generality:** As a generic cross-technology communication framework, my design mechanisms including mo-dulation, demodulation, and interval multiplexing commonly apply to different underlying technologies. This is possible because, according to the standards, (i) beacons are adopted in all three technologies of WiFi, ZigBee, and Bluetooth, (ii) they commonly allow modification of beacon timings, (iii) they reside on overlapping frequencies in the 2.4GHz band and finally, (iv) the light-weight design makes my design feasible even to low-end devices, as demonstrated later in the subchapter.

• **WiFi:** Figures 5.16(a) and 5.16(b) show the two WiFi platforms on which my design is implemented: WARP [93] and laptops. The former is a wireless research platform fully integrated with WiFi PHY/MAC layers. As a FPGA-based system allowing real-time operation, the evaluation on WARP enables precise exploration into FreeBee performance. Further implementation on six different laptops with various WiFi NICs from Qualcomm, Realtek, and Intel avoids hardware bias.

Evaluations via laptops utilize `Lorcon2` packet injection library [47] to schedule beacons according to FreeBee, which is a reasonable approach since laptops/PCs running software AP are frequently used in practice [48]. In Table 5.1, I use WiFi channel 1 (unless otherwise specified) for communication with ZigBee, one of the three most commonly used channels (i.e., 1, 6, and 11), granting generality to my result. I then tune to channel 4, which overlaps with a Bluetooth advertising channel (i.e., 38), for communication with Bluetooth.

• **ZigBee:** I use 30 ZigBee-compliant MICAz nodes (Figure 5.16(c)) for my experiments. When operating as a receiver, a MICAz node captures RSSI values (at 7.8KHz sampling rate by default) and records them within its 512KB on-board flash. The values are either processed (i.e., demodulated) within the node or flushed to a PC for analysis, depending on experiments. I use channels 11-15, overlapping with WiFi channels 1 and 4, and a Bluetooth advertising channel of 38.

• **Bluetooth:** My design is implemented on IOGEAR Bluetooth LE USB adapter, a cheap (∼12 USD) off-the-shelf product, shown in Figure 5.16(d). On this device, I utilize `AltBeacon` [94] library running under Linux's BlueZ driver for FreeBee embedding. Specifically, connectable directed advertising was used to generate FreeBee-enabled beacons on all three advertising channels of 37-39, which complies to the standard on Bluetooth beaconing.

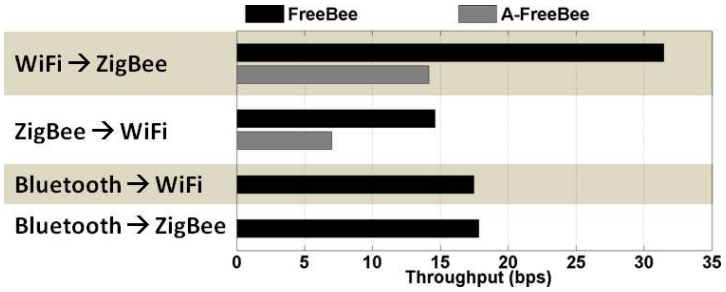
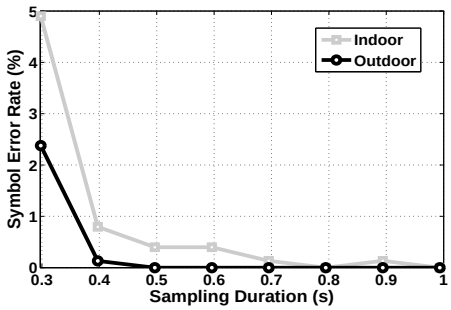


Figure 5.17: Per-sender throughputs

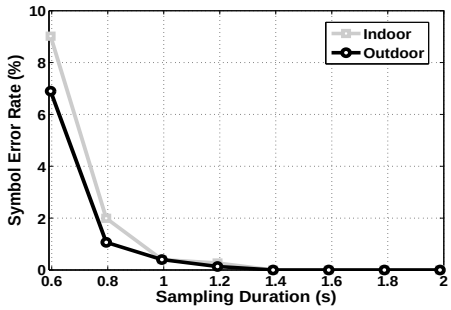
5.6.2 Per-sender Throughput

In this subchapter I demonstrate the data rate achievable per sender, under free side-channel as well as when the channel is fully utilized.

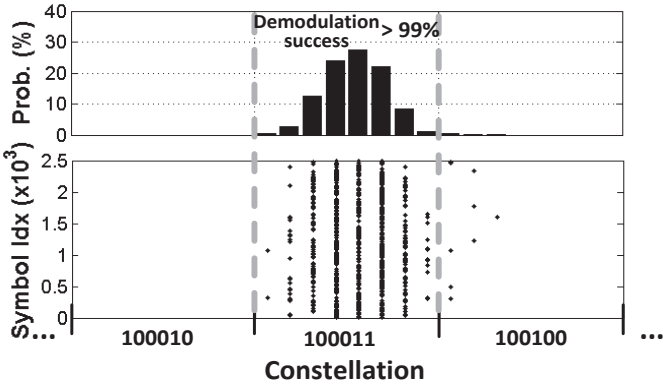
• **Free side-channel throughput:** Figure 5.17 illustrates per-sender data rate without introducing additional traffic (i.e., free side-channel). The experiment was conducted in a residential area with 20+ APs in proximity. Beacon intervals are set as 99.3, 76.8, and 78.75ms for WiFi, ZigBee, and Bluetooth, respectively, where the rate may easily be enhanced with shorter beacon intervals. The figure demonstrates two ideas: (i) The performance of A-FreeBee is slightly less than half of that of FreeBee, due to doubled sampling duration in A-FreeBee. Longer sampling duration increases the chance



(a) FreeBee



(b) A-FreeBee



(c) Constellation diagram and distribution

Figure 5.18: Indoor and outdoor performances

of larger fold sum of noise, hence yields higher SER in A-Freebee compared to FreeBee. This agrees with my theoretical analysis in Subchapter 5.5.1. (ii) Among different communication directions, WiFi to ZigBee exhibits the fastest rate of 31.5bps for FreeBee. The rate drops to 14.6bps for ZigBee to WiFi as the ZigBee standard enforces large unit shift (i.e., Δ=15.36ms), reducing the amount of information embedded in a symbol. While the Bluetooth standard defines a fine-grained beacon shift unit (Δ = 0.625ms), the random backoff ranging up to 10ms affects the performances of Bluetooth to WiFi and ZigBee communications where they show 17.5 and 17.8bps, respectively. While disabling the backoff functionality would increase the throughput significantly, the case is not considered in this work as it requires modification to the standard (lacks compatibility).

• **Upper bound throughput under ideal conditions:** I demonstrate the effectiveness of my design by comparing the maximum throughput of my design to that of Esense [82], a state-of-art cross-technology communication scheme. Esense, as a WiFi to ZigBee communication technique, conveys data by modulating WiFi packet durations, where its ideal maximum throughput is reported under the ideal condition of interference-free channel that is fully utilized by a single sender. Hence I adopt this setting in evaluating FreeBee. Evaluation parameters also follow the values proposed in Esense; That is, RSSI sampling rate of $32KHz$ (= sampling interval of $30.5us$), inter-frame duration of $90us$, and the maximum WiFi transmission rate of 54Mbps (802.11g).

To obtain the ideal FreeBee performance, I first note that beacons need not be repeatedly transmitted under the interference-free channel. That is, a single beacon frame conveys a symbol where SER=0. When the maximum shift is $x\Delta$, a symbol (i.e., a beacon) embeds $log_2(x+1)bits$, where the time it consumes consists of inter-frame duration, beacon transmission time, and the amount of shift, where the unit of shift may be as small as the sampling interval (i.e., $\Delta = 30.5us$). When $x = 4$, FreeBee yields the throughput of 10.2Kbps given the beacon length of 100Bytes. Esense, according to its researchers, achieves the throughput of 5.13Kbps under the same setting. This is because Esense requires to use long-length packets (with long air-time) up to 1,500Bytes to enable measurement of WiFi packet durations via a low-end ZigBee node. Meanwhile, FreeBee utilizes short beacons to offer higher channel efficiency.

5.6.3 Symbol Error Rate

Here I present the reliability of my design in practice by evaluating SER under both indoor and outdoor environments in a residential area. This experiment is based on WiFi(WARP) to ZigBee communication, where more than 2,500 symbols are sent and demodulated for SER computation. The results for FreeBee and A-FreeBee are depicted in Figures 5.18(a) and 5.18(b). Both designs reach $SER \leq 0.5\%$ when $\rho = 5$ (i.e., $0.5s$ and $1s$ for FreeBee and A-FreeBee), regardless of the environment. Furthermore, both designs perform better in outdoor environments, due to lower channel occupancy, B.

The lower figure in 5.18(c) illustrates the constellation, along with the demodulated positions of received symbols, when the 6bit symbol of '100011' is repeatedly sent 2,500 times. Demodulation is successful when a dot resides inside the region marked by gray

dotted lines in the constellation. The upper figure in 5.18(c) shows the distribution of the dots, in which >99% are successfully demodulated.

5.6.4 Cross-technology/channel Broadcast

This subchapter demonstrates FreeBee's unique capability to *broadcast* to receivers with heterogeneous technologies and channels.

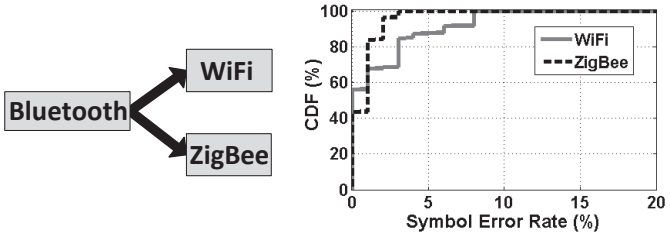


Figure 5.19: Simultaneous broadcast from Bluetooth to WiFi and ZigBee

• **Cross-technology broadcast**: Depicted in Figure 5.19, as a generic communication framework, FreeBee allows broadcast to heterogeneous receivers with overlapping frequencies. My generic design is effective in practice, since it avoids the complexity associated with technology-specific operation. In the experiment, WiFi and ZigBee were set to channels 4 and 15 to listen to Bluetooth's advertisement channel of 38 simultaneously. In this particular case shown in Figure 5.19, WiFi receiver, compared to ZigBee, suffers from larger SER. It is due to higher noise, as WiFi channel 4 overlaps with the popular WiFi channels of 1 and 6, while ZigBee channel 15 does not.

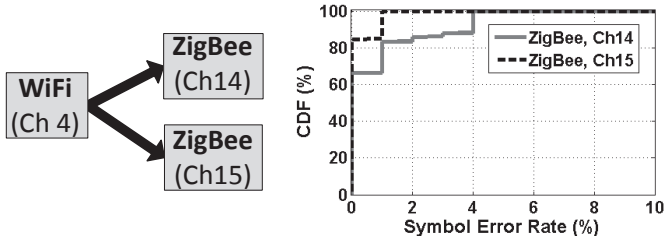
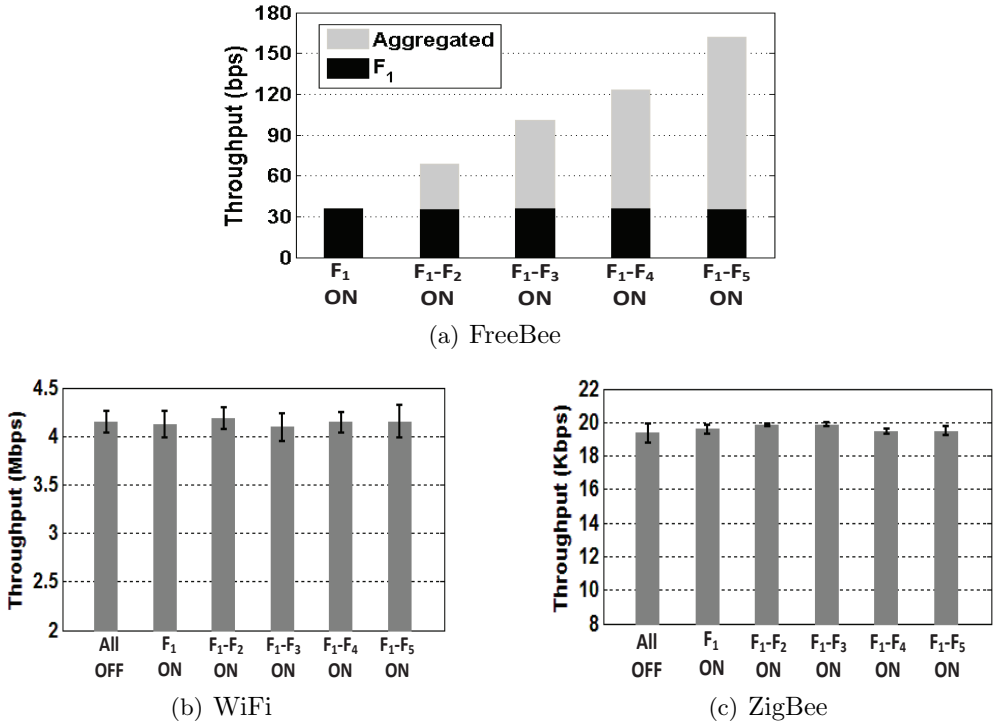


Figure 5.20: Simultaneous broadcast from WiFi to ZigBees on different channels

• **Cross-channel broadcast**: Figure 5.20 demonstrates another FreeBee's feature where a sender with wider bandwidth (WiFi) reaches multiple narrower-band receivers (ZigBee) on different channels with a single broadcast. WiFi's bandwidth spans 20MHz



(a) FreeBee

(b) WiFi

(c) ZigBee

Figure 5.21: FreeBee transparency

while it is only 2MHz for ZigBee. The experiment result in Figure 5.20 shows that SER on channel 14 is larger than that of channel 15. This is because channel 14 is affected by the noise from the heavily-used WiFi channel 1.

I note cross-technology broadcast and cross-channel bro-adcast can be combined to support more sophisticated scenarios where multiple heterogeneous senders deliver control symbols (through interval multiplexing) to multiple heterogeneous wireless receivers running under different channels. Such a capability would encourage further research on cross-technology coordination and control.

5.6.5 Transparency

I deploy five WiFi FreeBee senders, F_1-F_5, with beacon intervals of $\{91.1, 99.3, 103.4, 105.5, 109.6\}$ ms corresponding to $\{89\Delta, 97\Delta, 101\Delta, 103\Delta, 107\Delta\}$. Prime intervals indicate they are interval-multiplexed. Each sender is allowed to operate in FreeBee ON or OFF modes. A sender embeds FreeBee symbols only when it is ON. When OFF, the sender acts as

a legacy AP to simply transmit beacons periodically without symbol embedding. Under this setting, I observe the throughput of legacy wireless networks when a single to multiple FreeBee APs are operational, as well as the cross-interference among them. To do so, I first turn only F_1 to ON mode, while leaving all the others OFF. Then, I turn $F_2 - F_5$ to ON one by one every 10 minutes.

• **Transparency between FreeBees**: As the black bar in Figure 5.21(a) demonstrates, the throughput of F_1 is kept stable at an average of 35.5bps in the face of multiple concurrent FreeBee transmissions. This validates the performance of interval multiplexing in suppressing cross-interference among FreeBee signals. In fact simultaneous transmission helps to linearly increase the aggregated throughput up to 161.8bps, as shown by the gray bar. While the throughput can be enhanced by selecting shorter intervals for $F_1 - F_5$, I select ones close to that of a legacy AP ($102.4ms = 100\Delta$) to limit the channel usage to a similar level.

• **Transparency to legacy networks**: I again turn $F_1 - F_5$ to ON mode one at a time every 10 minutes, during which throughput between a pair of WiFi or ZigBee nodes are measured. The results are demonstrated in Figures 5.21(b) and 5.21(c). For WiFi, I use `Iperf` [95] to measure TCP throughput when operating in 802.11g. The WiFi sender was placed beside the FreeBees, 8m away from the WiFi receiver. For ZigBee measurement, sender and receiver were placed 5m apart, where 20byte packets were continuously transmitted at the fastest speed (i.e., with the minimum inter-packet delay). The figures show stable throughputs averaging 4.1Mbps and 19.6Kbps for WiFi and ZigBee, respectively, with small standard deviations. This suggests that both networks are unaffected by the presence of FreeBees, confirming that the free side-channel design successfully keeps FreeBee transparent to legacy networks.

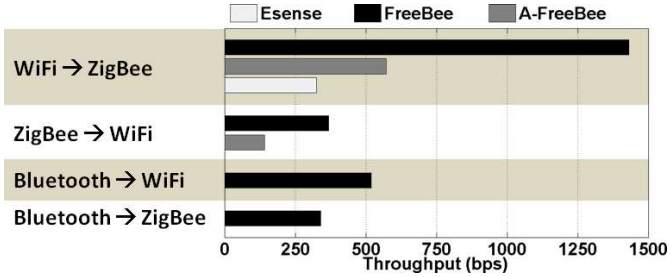
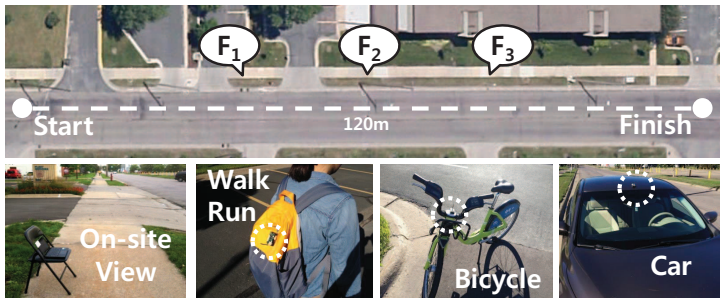


Figure 5.22: Aggregated throughput via interval multiplexing

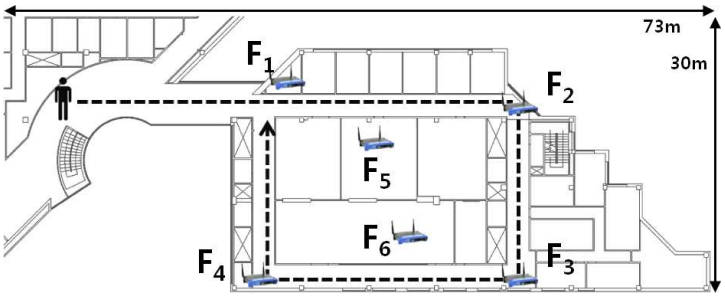
5.6.6 Aggregated Throughput

This subchapter investigates throughputs achievable via interval multiplexing, under different communication scenarios. Through interval multiplexing my design supports: (i) boosted aggregated throughput via concurrent transmission from multiple senders and (ii) adoption of additional beacons in a single sender to increase the throughput of that sender. I note that (i) and (ii) can be applied simultaneously; that is, multiple senders transmit concurrently while each sender freely adopts additional beacons in case of a need for higher throughput. Figure 5.22 depicts the throughputs for different communication scenarios, where I compare my design to the state of the art, Esense. According to the authors, when multiple Esense senders are present and the interference from the heterogeneous wireless systems are negligible, Esense achieves the bit rate of 1.63Kbps when a single WiFi packet is used to deliver one symbol, where five consecutive packets are needed per symbol for reliable communication [82]. This yields a throughput of 326bps. Following the same setting, Figure 5.22 demonstrates the rate achievable when the channel is shared among multiple (A-)FreeBee senders. FreeBee from WiFi to ZigBee shows the highest throughput of 1.4Kbps. The rate drops in other scenarios due to the same reasons discussed in the previous subchapter; that is, a large shift unit in ZigBee and random access delay in Bluetooth. It is notable that Bluetooth to WiFi shows higher throughput of 514bps, compared to 349bps in the Bluetooth to ZigBee case. This is because higher sampling rate achievable in WiFi compared to ZigBee (10MHz versus 7.8KHz in my setting) allows precise measurement of the beacon timings and reduces the chance of noise forming a high fold sum.

I note that the results in Figure 5.22 is achievable under a large number (∼hundreds) of concurrent senders, which can be realized by the following two techniques: (i) To allow numerous interval-multiplexed symbols, I choose beacon intervals from a set of numbers whose pair-wise LCM is longer than the length of the sampled RSSI vector. This clearly offers many more adoptable intervals compared to just using primes, while maintaining the effect of the interval multiplexing. (ii) As shown in the previous subchapter, interval-multiplexed FreeBee symbols are transparent to each other under usual volume of concurrent senders (e.g., tens of FreeBee WiFi APs). Here I consider the extreme case of hundreds of senders where I take cross-interference into account. I address this issue of cross-interference by eliminating the beacons from the RSSI vector after they



(a) Outdoor



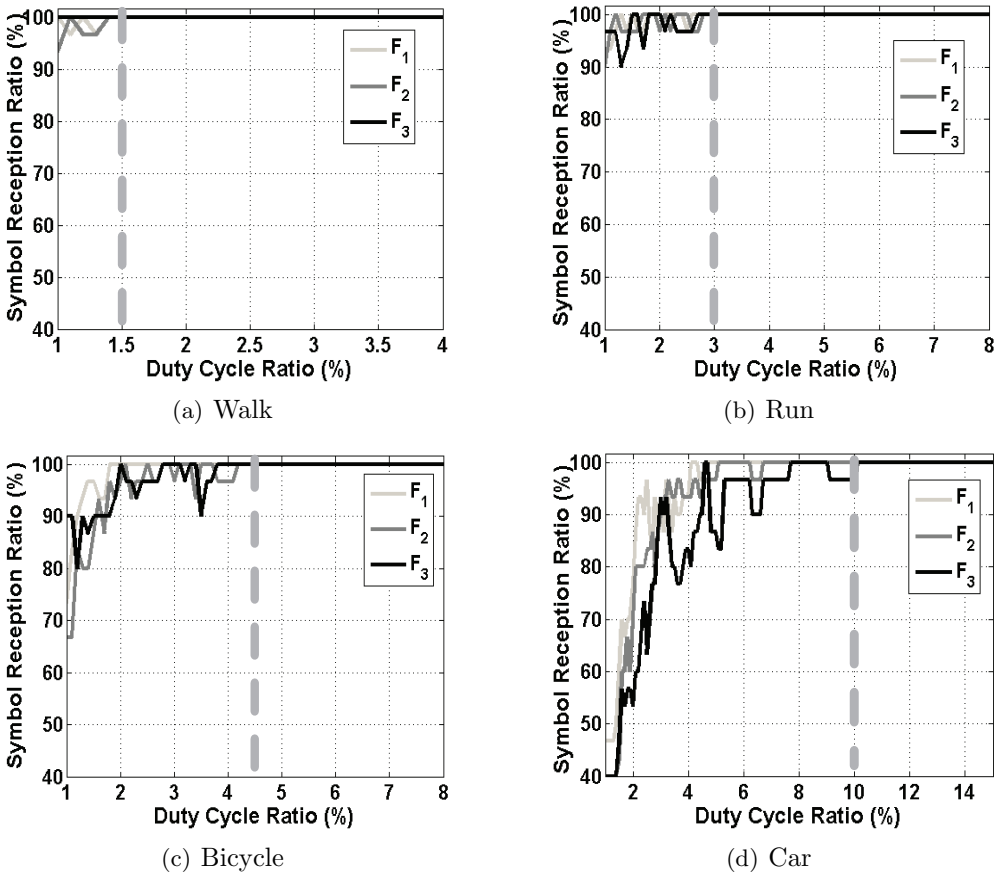
(b) Indoor

Figure 5.23: Mobile scenario evaluation

are interpreted. Specifically, I demodulate symbols in the ascending order according to their intervals. After demodulating each symbol, the beacons (i.e., RSSIs) corresponding to that symbol are eliminated (i.e., flipped from 1 to 0 in the RSSI vector). By doing so the cross-interference between symbols are effectively canceled to yield a high throughput. For instance, in WiFi to ZigBee FreeBee, SER is suppressed to be less than 5% when over 400 symbols with intervals between 91.1ms and 2s are concurrently sent.

5.6.7 Mobility and Duty Cycling

Here I consider a practical scenario of mobile receivers. For instance, this insight offers ideas on the feasibility of A-FreeBee for applications that requires location tracking. Furthermore, I take duty cycle into account where radios are turned off the majority of the time to preserve energy, a technique often adopted in battery-powered ZigBee networks to support long-term operations [96, 97, 98]. I deploy several WiFi A-FreeBee senders (laptop) in a university building and on a street, where the mobile receivers pass by the senders at different speeds: walking, running, on a bicycle, and in a car.



(a) Walk (b) Run

(c) Bicycle (d) Car

Figure 5.24: Mobile scenario performance

It is worth noting that this experiment takes advantage of my free-side channel design, as (A-)FreeBee symbols can be continuously broadcast without occupying the channel, thus reaching mobile and/or duty-cycled receivers whose presence or active periods are unknown a priori.

• **On a street**: In this experiment, I deploy 3 A-FreeBee senders, 20m apart, on a street as shown in Figure 5.23(a). The figure also shows ZigBee receivers with different degrees of mobility: walking (4.3 mph), running (6.8 mph), on a bicycle (10.8 mph) and in a car (30 mph). Thirty receivers were tested for each speed. Figure 5.24 informs that A-FreeBee symbols from all the senders can be safely received with duty-cycling of only 1.5%, 3%, 4.5% and 10% when walking, running, cycling and riding in a car,

respectively.

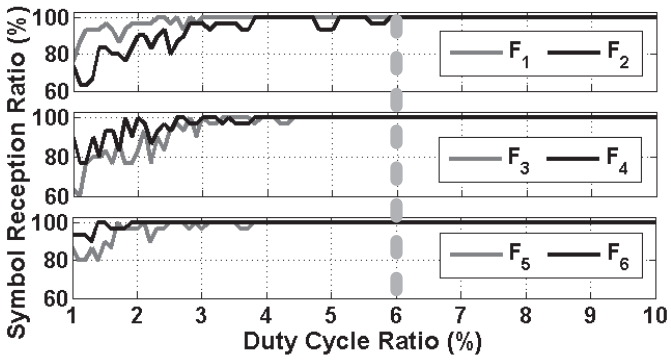


Figure 5.25: Performance under indoor mobility

• **In a university building**: A total of 6 FreeBee senders are deployed as shown in Figure 5.23(b) to show the impact of various positions, such as hallways and rooms with closed or open doors. A person walks at a moderate speed of 4.3 mph through the hallway following the arrow in the figure, with a ZigBee receiver tied to his/her backpack. The result after repeating for 30 different receivers is shown in Figure 5.25. Overall, as indicated by the gray dotted line, duty cycling of 6% ensures reception of A-FreeBee symbols from all the senders. A higher duty cycle ratio is required compared to the 1.5% when walking outside in the previous experiment due to the noisy environment with 50+ legacy WiFi APs deployed.

5.6.8 Receiver Overhead

In this subchapter I demonstrate light-weight storage and computation/energy overheads of (A-)FreeBee demodulation, making it affordable even to low-end devices (e.g., ZigBee mote) without disrupting the ongoing legacy protocols.

• **Storage**: Let us consider an example case where the sender is a WiFi AP with the beacon interval of $T = 97\Delta = 99.3ms$ (i.e., λ=776) and the receiver is a ZigBee mote. As explained earlier, the receiver samples RSSI every $128us$, where the total number of $\rho \times \lambda$ samples are collected for demodulation. As a RSSI value is represented as a single bit, under $\rho = 5$ FreeBee requires 5×776=3.88Kbits (i.e., 485Bytes) of memory. This takes up less than 1% of the storage offered in popular commodity ZigBee motes of MICAz and TelosB, where they offer 512KBytes and 1MBytes of on-board flash, respectively.

Moreover, it is also important to note the followings: First, concurrent demodulation of n interval-multiplexed FreeBee symbols whose intervals are $\lambda_1 < \lambda_2 < ... < \lambda_n$, is achieved by collecting $\rho \times \lambda_n$ samples. Second, it is possible to reduce the memory usage by storing only the timing of transitions (i.e., $0 \leftrightarrow 1$) in RSSI values. Third, A-FreeBee requires to store twice the number of samples compared to the basic FreeBee.

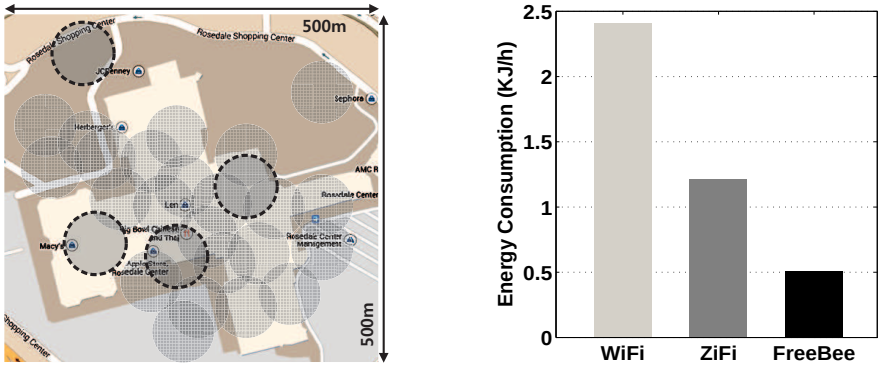
• **Computation/Energy**: The demodulation process can be decomposed into three parts: (i) sampling RSSI, (ii) adding to fold sum, and lastly, (iii) locating the position of the maximum fold sum. (i) and (ii) are repeatedly performed every $128us$, periodically incurring small overheads to perform a fetch and an addition operations. Upon collecting sufficient number (i.e., $\rho \times \lambda$) of samples (iii) takes place in which series of compare operations are executed in search for the maximum of the λ fold sums (2λ for A-FreeBee).

I demonstrate the overhead of the entire steps (i.e., (i)-(iii)) on a off-the-shelf, low-end system – ZigBee-compliant MICAz node. Specifically, I observe the execution time the node spends to process the demodulation. This is accurately measured by triggering a GPIO (general-purpose input/output) pin, whose activity is captured with Tektronix DPO 4054 oscilloscope. It is found that the demodulation of a symbol with an interval of $\lambda = 776$ costs $51.5ms$ of computation time, consuming only $1.2mJ$ of additional energy compared to sleeping. The light overhead allows FreeBee to run on commodity low-end hardwares without disrupting the legacy protocols. Finally, I note that demodulating multiple, interval-multiplexed FreeBee symbols requires more computation/energy where they still remain affordable; For instance, listening to 10 concurrent symbols costs $98.5ms$ of computation, indicating $2.4mJ$ of energy.

5.7 APP: Wake on Selective WLAN

In battery-powered mobile devices such as smart phones, WiFi NIC continuously scanning the channel for available APs, leading to significant energy drainage. To tackle this issue, recent studies utilize a low-power secondary radio to wake up the WiFi interface only when APs are present, since a WiFi interface consumes $669.9mW$ [99] when it is awake, but it drops to only $10mW$ once it starts to sleep. ZigBee, in contrast, requires only $75mW$ [100], even when active. Among the works in this category, ZiFi [85] is a

state-of-the-art solution that offers a simple, infrastructure-free approach to detect APs in proximity with a ZigBee SD card inserted in a smart phone.



(a) WiFi zones (b) Energy Consumption

Figure 5.26: PErformance of Wake on Selective WLAN

Although results in [85] are encouraging, I believe that further savings can be achieved with FreeBee. Given that most of APs nowadays are private, blindly waking up the WiFi without knowing if it is accessible or not leads to energy waste. This problem can be easily solved with FreeBee by allowing APs to broadcast binary information indicating if it is an open or a private WiFi. Figure 5.26(a) illustrates the benefit of this approach by considering a $500 \times 500 m^2$ area around a shopping mall. For a realistic evaluation, I observe all private and open WiFi APs currently running in the area[1] . Private WiFi APs are marked as light gray circles and open APs illustrated as outlined circles. The radius of the circles is set as 43.7m, following the value obtained in my experiments. A total of 27 WiFi APs take up 43.2% of the entire area. On the other, hand only 4 open WiFi APs are found, covering only 9.6%.

To ensure a fair comparison, the energy consumption of the ZigBee node is also considered in case of ZiFi and FreeBee. The results in Figure 5.26(b) indicate that the energy consumption for WiFi, ZiFi, and FreeBee are 2.41KJ/h, 1.21KJ/h, and 0.51KJ/h, respectively. In sum, FreeBee saves 78.9% and 58.2% of the energy required by WiFi and ZiFi, respectively, without sacrificing opportunities for open WiFi access.

[1] Real AP deployment information retrieved from *wigle.net* and *openwifispots.com*.

5.8 Related Works

The FreeBee design of embedding information within the beacon timing is inspired by PPM (Pulse Position Modulation), which has been widely studied in the field of optical and UWB (Ultra Wide Band) communications [101, 102, 103, 104, 105, 106]. The extensive research in this area, however, is not applicable to my scenario because they are designed for pulses (which correspond to beacons in my case) that occur at their exact timing and/or can be differentiated from noise via a matched filter. Neither of these conditions is met by the beacons, since they are prone to channel access delays due to CSMA and they are indistinguishable from other traffic for receivers with incompatible PHY layers.

Improvement through cross-platform interaction has been investigated in many recent works [91, 107, 108, 109, 110, 111, 85], while only a handful have focused on cross-platform direct communication, including Esense [82], HoWiES [88], and GSense [83]. Esense establishes communication channels from WiFi to ZigBee by modulating packet lengths to those are unlikely to be used in normal WiFi traffic. HoWiES extends the Esense mechanism to convey data with combinations of WiFi packets, and introduces a simple coding technique. Finally, GSense enables cross-technology communication by prepending a customized preamble to legacy packets containing sequence of pulses in which gaps between them embed the data to be delivered. FreeBee uniquely departs from the above works in a few aspects: (i) the use of the mandatory beacons allows free-side-channel communication, (ii) interval multiplexing allows concurrent communication between multiple senders and receivers with implicit addressing, (iii) does not require modification to the commodity hardware (in contrast to GSense), and (iv) provides the first running prototype showcasing a generic communication capability among three popular wireless technologies.

5.9 Conclusion

In this chapter I propose FreeBee, a cross-technology communication framework that aims to take advantage of the coexistence problem by adopting mutual supplementation. Extensive testbed experiments on three popular wireless technologies, WiFi, ZigBee, and Bluetooth, reveal that FreeBee offers reliable symbol delivery within less than a

second. Its unique free side-channel design and the utilization of short beacon frames grant high spectrum efficiency and feasibility; FreeBee exhibits 4.3× the rate compared to the state-of-the-art, Esense, and demonstrates strong support for highly mobile (in a car) and extremely duty cycled ($\leq 5\%$) receivers, implying its applicability to a wide range of practical applications. An examination of real WiFi deployment patterns in a shopping mall area finds that FreeBee can save 78.9% of the energy otherwise wasted by the WiFi interface.

Appendix: SER Derivation

Table 2: Parameters

T	Beacon interval in time
λ	# of RSSI sample in T
ρ	# of repetitions
B	Channel occupancy rate
B_f	Channel occupancy rate after filter
B_b	Probability of beacon fold

From Table 2, recall that beacon repetition ρ is the number of rows in the matrix upon folding. The number of columns after folding is λ and 2λ in FreeBee and A-FreeBee. B is the ratio of 1's in the sampled RSSI vector, reflecting the amount of traffic (i.e., noise) present in the channel. B_f stands for the ratio after taking only first two samples as introduced in Subchapter 5.4. Due to such filtering, beacon is folded (i.e., contributes to fold sum) only when the delay of the beacon is less than $256us$, where this probability is denoted as B_b.

• **FreeBee**: SER corresponds to the probability where fold sum of noise in *any* of $\lambda - 1$ columns (excluding the one with beacon) is greater than or equal to that of beacon. The probability of a column to have n fold sum follows binomial distribution, $\binom{\rho}{n} B_f^n (1 - B_f)^{\rho - n}$. Let N denote a random variable indicating the *maximum* of the noise fold sums. Then, N is distributed as:

$$Pr(N \leq n) = \left\{ \sum_{i=0}^{n} \binom{\rho}{i} B_f^i (1 - B_f)^{\rho - i} \right\}^{\lambda - 1}$$

$$Pr(N = n) = Pr(N \leq n) - Pr(N \leq n - 1)$$

Letting S indicate the fold sum induced by beacons, then it is also binomially distributed:

$$Pr(S = s) = \binom{\rho}{s} B_b^s (1 - B_b)^{\rho - s} \tag{2}$$

where SER can be computed from S and N.

$$\begin{aligned} SER &= Pr(S - N \leq 0) \\ &= \sum_{s=0}^{\rho} \sum_{n=s}^{\rho} Pr(S = s) Pr(N = n) \end{aligned}$$

• **A-FreeBee**: Symbol error occurs in A-FreeBee when fold sum of noise in any of $2\lambda - 2$ columns (excluding the two with beacon) is greater than or equal to that of either beacon fold sums. Maximum noise fold sum, N, follows below distribution:

$$\begin{aligned} Pr(N \leq n) &= \left\{ \sum_{i=0}^{n} \binom{\rho}{i} B_f^i (1 - B_f)^{\rho - i} \right\}^{2(\lambda - 1)} \\ Pr(N = n) &= Pr(N \leq n) - Pr(N \leq n - 1) \end{aligned}$$

The distribution of the beacon fold sum, S, is equivalent to Eq 2 where SER is found from S and N.

$$\begin{aligned} SER &= 1 - \{Pr(S - N > 0)\}^2 \\ &= 1 - \left\{ \sum_{n=0}^{\rho - 1} \sum_{s=n+1}^{\rho} Pr(S = s) Pr(N = n) \right\}^2 \end{aligned}$$

In summary, my analysis indicates that SER increases with B (or B_f) and reduces with larger ρ as also shown in Figures 5.14(a) and 5.14(b).

Chapter 6

Future Work

In the near future, I aim to investigate further opportunities behind wireless coexisting environment and cross-technology communication, to reach beyond the capability of each device. In the long run, I will focus on realizing extreme-scale collaborative network that integrates diverse wireless technologies, in the aim to bring ubiquitous computing in IoT to reality.

6.1 Short term plan

•**Transparent spectral adaptation:** Prior work addresses the interference due to coexisting heterogeneity in temporal and spatial perspectives. I plan to further push the coexistence technology to explore the opportunities available in the frequency domain. In other words, I will introduce an approach to reliably communicate under the densely occupied spectrum, transparently using the commercial, off-the-shelf devices for immediate impact at a low deployment cost. This improves the channel efficiency by utilizing spectrum fragments current systems fail to exploit, or in other words, waste.
•**Enhanced cross-technology communication:** As mentioned earlier, enabling communication between heterogeneous technologies is a fundamental building block for collaborative networking. I intend to improve the aforementioned scheme of FreeBee in diverse aspects; one example would be in throughput, which is limited to 1.4kbps in FreeBee. This can be achieved by exploiting communication opportunities behind abundant data packets, instead of the current design utilizing only beacons, which are strictly

limited in volume (<4% of the total traffic in WiFi). Enhancement in speed is expected to substantially improve the practical usage of the technique.

•**Global channel access protocols:** Many of channel access mechanisms, including RTS/CTS (Request to Send/Clear to Send) and NAV (network allocation vector), are defined assuming homogeneity and thus are prone to failure when different technologies compete for the medium. Resolving this issue requires extending the local mechanisms globally across wireless technologies, which can be realized by cross-technology communication schemes. Such designs enable explicit channel sharing among wireless technologies in a distributed manner, to offer reliability, fairness, and efficiency to coexisting networks.

•**Advanced applications via information sharing:** Another benefit of cross-technology communication is the real-time information sharing among platforms, to support diverse time-critical applications. For example, personalized and immediate emergency services are realized by combining geo-location in WiFi with health data available to Bluetooth-enabled heart monitors. I will explore and implement rich applications that assist our daily lives in various domains.

6.2 Long term plan

•**Collaborative Network Model:** Pervasive connectivity among wireless technologies essentially calls for definition of a new unified network model embracing all wireless devices, which I call the collaborative network model. It contains two notable differences from the traditional networks: (i) Legacy networks assume identicalness among devices, which no longer holds for the collaborative network; different wireless systems own disparate amounts of resources (e.g., computation power and storage) as well as accessible data. (ii) Intra and cross-technology links have distinct features; for example, a cross-technology link exhibits significant asymmetry due to considerable variation in the transmission powers across different wireless technologies - WiFi's radio power is more than two orders of magnitude higher than ZigBee and Bluetooth. The two properties impose significant challenges and provide extensive research opportunities.

•**Abstract User Interface:** For full utilization of the collaborative network and rapid

development of its applications, I plan to work on abstracting the sophisticated architecture of the network to provide a high-level interface. Hiding the complexity the interface should provide intuitive methods to exploit the functions of the network. Ideally, the interface should be made transparent by mimicking that of traditional networks which developers/users are familiar with. For instance, unidirectional links in the collaborative network (due to asymmetry) shall be abstracted as bidirectional where the underlying reverse link, in fact, involves multi-hop communication.

•**Collaborative Sensor Fusion:** The technique of sensor fusion blends diverse information acquired from various sources to gain a precise view of the physical world. This plays a critical role in constructing reliable IoT, especially for accuracy-sensitive applications such as those related to health and safety. From this standpoint, a collaborative network has two unique advantages over traditional networks. First, universal fusion across wireless technologies can be performed online and on-demand. This strongly supports a real-time and distributed operation of IoT, while avoiding overburdening centralized servers with traffic and computation. This is especially useful for highly-responsive applications such as augmented reality. Second, a collaborative network enables data upload to clouds, even from devices without direct Internet access, which is achieved via collaborative, indirect connections. This considerably enriches the volume of collectable data, subsidizing Big Data with high confidence.

Chapter 7

Conclusion

This dissertation proposes techniques for efficient networking under the IoT environment, where heterogeneous networks coexist. To support reliable unicast under arbitrary interference, cETX explores temporal white spaces within the interfered channel by taking various aspects of interference into consideration including volume, pattern, and temporal dynamics. Following work of CorrModel supports efficient broadcast under coexistence. This work spatial interference patterns are exploited to accurately infer packet deliveries across nearby receivers to avoid unnecessary spectrum usage, while keeping the network reliable. Both cETX and CorrModel techniques are generically designed to support networks under a wide range of settings: They are demonstrated to improve (i) diverse routing protocols (ii) running different wireless technologies, (iii) deployed in various indoor and outdoor environments. Evaluations on twelve existing protocols on WiFi and ZigBee grant 20-30% performance gain. Lastly, in the light of the benefit of coexistence by cross-technology collaboration, FreeBee, a novel cross-technology communication framework was proposed. Extensive testbed experiments on three popular wireless technologies, WiFi, ZigBee, and Bluetooth, reveal that FreeBee offers reliable communication, with 4.3× the rate compared to the state-of-the-art. Moreover, its unique free side-channel design and the utilization of short beacon frames grant high spectrum efficiency and feasibility.

References

[1] Song Min Kim, Shuai Wang, and Tian He. cetx: Incorporating spatiotemporal correlation for better wireless networking. In *The 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2015.

[2] Song Min Kim, Shuai Wang, and Tian He. Exploiting causes and effects of wireless link correlation for better performance. In *The 34th IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[3] Song Min Kim and Tian He. Freebee: Cross-technology communication via free side-channel. In *The 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2015.

[4] Daniel Aguayo, John Bicket, and Robert Morris. Srcrr: A high throughput routing protocol for 802.11 mesh networks. In *Ph.D. Diss, MIT*, 2004.

[5] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom*, 2004.

[6] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE INMIC*, 2001.

[7] K.M. Alzoubi, P.J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Hawaii Int. Conf. System Sciences*, 2002.

[8] J.Cartigny, F. Ingelrest, and D. Simplot. Rng relay subset flooding protocols in mobile ad hoc networks. *International Journal of Foundations of Computer Science*, 2003.

[9] E. L. Li, R. Ramjee, M. M. Buddhikot, and S. C. Miller. Network coding-based broadcast in mobile ad-hoc networks. In *INFOCOM*, 2007.

[10] H. Lim and C. Kim. Flooding in wireless ad hoc networks. In *Computer Communications Journal*, 2001.

[11] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 2002.

[12] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS*, 2002.

[13] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. In *Wireless Comm. and Mobile Comp.*, 2003.

[14] Dave Evans (Cisco). The internet of things how the next evolution of the internet is changing everything. *http://www.cisco.com*, 2011.

[15] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the rf smog: making 802.11n robust to cross-technology interference. In *SIGCOMM*, 2011.

[16] Bozidar Radunovic, Ranveer Chandra, and Dinan Gunawardena. Weeble: enabling low-power nodes to coexist with high-power nodes in white space networks. In *CoNEXT*, 2011.

[17] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16, 2010.

[18] Xinyu Zhang and Kang G Shin. Enabling coexistence of heterogeneous wireless systems: case for zigbee and wifi. In *MobiHoc*, 2011.

[19] Xinyu Zhang and Kang G Shin. Cooperative carrier signaling: harmonizing coexisting wpan and wlan devices. *Networking, IEEE/ACM Transactions on*, 21(2):426–439, 2013.

[20] Zenghua Zhao, Xuanxuan Wu, Xin Zhang, Jing Zhao, and Xiang-Yang Li. Zigbee vs wifi: Understanding issues and measuring performances of their coexistence. In *IPCCC*, pages 1–8, 2014.

[21] Peng Guo, Jiannong Cao, Kui Zhang, and Xuefeng Liu. Enhancing zigbee throughput under wifi interference using real-time adaptive coding. In *INFOCOM*, 2014.

[22] Shouling Ji, Raheem Beyah, and Zhipeng Cai. Snapshot/continuous data collection capacity for large-scale probabilistic wireless sensor networks. In *Infocom*, 2012.

[23] Jian Tang, Guoliang Xue, Christopher Chandler, and Weiyi Zhang. Interference-aware routing in multihop wireless networks using directional antennas. In *INFOCOM*, 2005.

[24] Peng-Jun Wan, Zhu Wang, Hongwei Du, Scott C.-H. Huang, and Zhiyuan Wan. First-fit scheduling for beaconing in multihop wireless networks. In *INFOCOM*, 2010.

[25] Guoliang Xing, Minming Li, Hongbo Luo, and Xiaohua Jia. Dynamic multiresolution data dissemination in wireless sensor networks. *IEEE Trans. Mob. Comput.*, 2009.

[26] Reza Banirazi, Edmond A. Jonckheere, and Bhaskar Krishnamachari. Heat-diffusion: Pareto optimal dynamic routing for time-varying wireless networks. In *INFOCOM*, 2014.

[27] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom*, 2003.

[28] Jie Wu, Mingming Lu, and Feng Li. Utility-based opportunistic routing in multi-hop wireless networks. In *ICDCS*, 2008.

[29] Alberto Cerpa, Jennifer L. Wong, Miodrag Potkonjak, and Deborah Estrin. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. In *MobiHoc*, 2005.

[30] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM TOSN*, 6(2), 2010.

[31] S Pattem, B Krishnamachari, and R Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM TOSN*, 2008.

[32] Kannan Srinivasan, Mayank Jain, Jung Il Choi, Tahir Azim, Edward S. Kim, Philip Levis, and Bhaskar Krishnamachari. The kappa factor: Inferring protocol performance using inter-link reception correlation. In *MobiCom*, 2010.

[33] Mit roofnet.

[34] E. N. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39(2), 1960.

[35] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. Four-bit wireless link estimation. In *HotNets*, 2007.

[36] Tao Liu and Alberto Cerpa. Foresee (4c): Wireless link prediction using link features. In *IPSN*, 2011.

[37] Tao Liu and Alberto Cerpa. Talent: Temporal adaptive link estimator with no training. In *SenSys*, 2012.

[38] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys*, 2006.

[39] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys*, 2004.

[40] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 2003.

[41] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-mac: A hybrid mac for wireless sensor networks. In *SenSys*, 2005.

[42] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *SenSys*, 2006.

[43] Wireless Personal Area Network (WPAN) Working Group. Ieee standard part 15.4: Low-rate wireless personal area networks (lr-wpans). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, Sept 2011.

[44] Wireless LAN Working Group. Ieee standard part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[45] Shuai Wang, Song Min Kim, Yunhuai Liu, Guang Tan, and Tian He. Corlayer: A transparent link correlation layer for energy efficient broadcast. In *MobiCom*, 2013.

[46] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, 1996.

[47] Lorcon wireless packet injection lib. `https://code.google.com/p/lorcon/`.

[48] Hostapd. `http://hostap.epitest.fi/hostapd/`.

[49] Kannan Srinivasan, Maria A. Kazandjieva, Saatvik Agarwal, and Philip Levis. The beta-factor: Measuring wireless link burstiness. In *SenSys*, 2008.

[50] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. Bursty traffic over bursty links. In *SenSys*, 2009.

[51] Can Emre Koksal and Hari Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11), 2006.

[52] Gaoyang Guan Jiajun Bu Tao Gu Zhiwei Zhao, Wei Dong and Chun Chen. Modeling link correlation in low-power wireless networks. In *INFOCOM*, 2015.

[53] Abdallah Khreishah, Issa M. Khalil, and Jie Wu. Distributed network coding-based opportunistic routing for multicast. In *MobiHoc*, 2012.

[54] S. M. Iftekharul Alam, Salmin Sultana, Y. Charlie Hu, and Sonia Fahmy. Link correlation and network coding in broadcast protocols for wireless sensor networks. In *SECON*, 2012.

[55] Manjunath Doddavenkatappa, Mun Choon Chan, and Ben Leong. Splash: Fast data dissemination with constructive interference in wireless sensor networks. In *nsdi*, 2013.

[56] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *MOBICOM '05*, 2005.

[57] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A High Throughput Path Metric for Multi-Hop Wireless Routing. In *MOBICOM*, 2003.

[58] E. Miluzzo, X. Zheng, K. Fodor, and A. T. Campbell. Radio Characterization of 802.15.4 and its Impact on the Design of Mobile Sensor Networks. In *EWSN*, 2008.

[59] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2001.

[60] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks. In *Technical Report SING-06-00*, 2006.

[61] M. Zuniga, and B. Krishnamachari. An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links. In *TOSN*, 2007.

[62] Kannan Srinivasan, Maria A. Kazandjieva, Saatvik Agarwal, and Philip Levis. The beta-factor: measuring wireless link burstiness. In *SENSYS*, 2008.

[63] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SENSYS*, 2003.

[64] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. The κ-factor: Inferring protocol performance using inter-link reception correlation. In *MOBICOM*, 2010.

[65] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *NSDI*, 2010.

[66] Shuo Guo, Song Min Kim, Ting Zhu, Yu Gu, and Tian He. Correlated flooding in low-duty-cycle wireless sensor networks. In *ICNP*, 2011.

[67] Zhiwei Zhao, Wei Dong, Jiajun Bu, Tao Gu, Chun Chen, Xianghua Xu, and Shiliang Pu. Exploiting link correlation for core-based dissemination in wireless sensor networks. In *SECON*, 2014.

[68] Anas Basalamah, Song Min Kim, Shuo Guo, Tian He, and Yoshito Tobe. Link correlation aware opportunistic routing. In *INFOCOM*, 2012.

[69] Chieh-Jan Mike Liang, Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *SenSys*, 2010.

[70] Muhannad Quwaider, Jayanthi Rao, and Subir Biswas. Transmission power assignment with postural position inference for on-body wireless communication links. *TECS*, 2010.

[71] Texas Instruments. 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver datasheet. *http://www.ti.com/lit/ds/symlink/cc2420.pdf*, 2013.

[72] Marina Petrova, Lili Wu, Petri Mähönen, and Janne Riihijärvi. Interference measurements on performance degradation between colocated ieee 802.11g/n and ieee 802.15.4 networks. In *ICN*, 2007.

[73] Atmel. Atmega128/l datasheet. *http://www.atmel.com/Images/doc2467.pdf*, 2011.

[74] Tao Liu and Alberto Cerpa. Talent: temporal adaptive link estimator with no training. In *SENSYS*, 2012.

[75] S. Biswas and R. Morris. Exor: Opportunistic multi-hop routing for wireless networks. In *SIGCOMM*, 2005.

[76] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *SECON*, 2004.

[77] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *TON*, 2008.

[78] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza. Rbp: Robust broadcast propagation in wireless networks. In *SENSYS*, 2006.

[79] Tao Liu and Alberto Cerpa. Foresee (4c): Wireless link prediction using link features. In *IPSN*, 2011.

[80] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proc. ACM IMC*, 2009.

[81] Ning Ding, Daniel Wagner, Xiaomeng Chen, Abhinav Pathak, Y Charlie Hu, and Andrew Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *ACM SIGMETRICS Performance Evaluation Review*, volume 41, pages 29–40. ACM, 2013.

[82] Kameswari Chebrolu and Ashutosh Dhekne. Esense: communication through energy sensing. In *MOBICOM*, 2009.

[83] Xinyu Zhang and Kang G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *INFOCOM*, 2013.

[84] ios: Understanding ibeacon. `http://support.apple.com/kb/HT6048/`.

[85] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. Zifi: wireless lan discovery via zigbee interference signatures. In *MOBICOM*, 2010.

[86] Randy Carroll, Rick Cnossen, Mark Schnell, and David Simons. Continua: An interoperable personal healthcare ecosystem. *Pervasive Computing, IEEE*, 6(4):90–94, 2007.

[87] Alex Chia-Chun Hsu, David S. L. Wei, and C. C. Jay Kuo. Coexistence wi-fi MAC design for mitigating interference caused by collocated bluetooth. *IEEE Trans. Computers*, 64(2):342–352, 2015.

[88] Yifan Zhang and Qun Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *INFOCOM*, 2013.

[89] D.H. Staelin. Fast folding algorithm for detection of periodic pulse trains. *Proceedings of the IEEE*, 57:724 – 725, 1969.

[90] Albert Edward Ingham. *The Distribution of Prime Numbers*. Cambridge University Press, 1932.

[91] Tian Hao, Ruogu Zhou, Guoliang Xing, and Matt Mutka. Wizsync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks. In *RTSS*, 2011.

[92] Crawdad: A community resource for archiving wireless data at dartmouth. `http://crawdad.cs.dartmouth.edu/`.

[93] Wireless open-access research platform. `http://warpproject.org/trac/`.

[94] Altbeacon. `http://altbeacon.org/`.

[95] Iperf. `https://iperf.fr/`.

[96] Zhichao Cao, Yuan He, and Yunhao Liu. L 2: Lazy forwarding in low duty cycle wireless sensor networks. In *INFOCOM*, pages 1323–1331, 2012.

[97] Dora Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and Chiara Petrioli. Beyond Duty Cycling: Wake-up Radio with Selective Awakenings for Long-lived Wireless Sensing Systems. In *INFOCOM*, 2015.

[98] Zhenjiang Li, Mo Li, and Yunhao Liu. Towards energy-fairness in asynchronous duty-cycling sensor networks. *TOSN*, 10(3):38, 2014.

[99] Cisco aironet 802.11a/b/g datasheet. `http://www.cisco.com`.

[100] Micaz datasheet. `http://www.memsic.com/`.

[101] Cecilia Carbonelli and Umberto Mengali. M-ppm noncoherent receivers for uwb applications. *Wireless Communications, IEEE Transactions on*, 5(8):2285–2294, 2006.

[102] Antonio A D'Amico, Umberto Mengali, and Eva Arias-de Reyna. Energy-detection uwb receivers with multiple energy measurements. *Wireless Communications, IEEE Transactions on*, 6(7):2652–2659, 2007.

[103] Z Ghassemlooy, AR Hayes, NL Seed, and ED Kaluarachchi. Digital pulse interval modulation for optical communications. *Communications Magazine, IEEE*, 36(12):95–99, 1998.

[104] Robert C Qiu, Huaping Liu, and Xuemin Shen. Ultra-wideband for multiple access communications. *Communications Magazine, IEEE*, 43(2):80–87, 2005.

[105] Da-shan Shiu and Joseph M Kahn. Differential pulse-position modulation for power-efficient optical communication. *Communications, IEEE Transactions on*, 47(8):1201–1210, 1999.

[106] Jianjun Wu, Haige Xiang, and Zhi Tian. Weighted noncoherent receivers for uwb ppm signals. *Communications Letters, IEEE*, 10(9):655–657, 2006.

[107] Tao Jin, Guevara Noubir, and Bo Sheng. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. In *INFOCOM*, pages 1593–1601, 2011.

[108] Wenxian Li, Yanmin Zhu, and Tian He. Wibee: Building wifi radio map with zigbee sensor networks. In *INFOCOM*, 2012.

[109] Rajesh Mahindra, Hari Viswanathan, Karthik Sundaresan, Mustafa Y Arslan, and Sampath Rangarajan. A practical traffic management system for integrated lte-wifi networks. In *MOBICOM*, pages 189–200, 2014.

[110] Sayandeep Sen, Tan Zhang, Milind M Buddhikot, Suman Banerjee, Dragan Samardzija, and Susan Walker. A dual technology femto cell architecture for robust communication using whitespaces. In *Dynamic Spectrum Access Networks (DYSPAN)*, 2012.

[111] Cigdem Sengul, Mehedi Bakht, Albert F. Harris, Tarek Abdelzaher, and Robin H. Kravets. On the feasibility of high-power radios in sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):37–39, 2008.