EXPLOITING CROSS LAYER OPPORTUNITIES FOR SECRECY AND EFFICIENCY IN WIRELESS NETWORKS

by

Sriram Nandha Premnath

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

May 2013

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Sriram Nandha Premnath**

has been approved by the following supervisory committee members:

Sneha Kasera	, Chair	**02/12/2013** Date Approved
Rajeev Balasubramonian	, Member	**02/06/2013** Date Approved
Robert Ricci	, Member	**02/06/2013** Date Approved
Neal Patwari	, Member	**02/06/2013** Date Approved
Behrouz Farhang-Boroujeny	, Member	**02/06/2013** Date Approved

and by **Alan Davis**, Chair of the School of **Computing**

and by Donna M. White, Interim Dean of The Graduate School.

ABSTRACT

Cross layer system design represents a paradigm shift that breaks the traditional layer-boundaries in a network stack to enhance a wireless network in a number of different ways. Existing work has used the cross layer approach to optimize a wireless network in terms of packet scheduling, error correction, multimedia quality, power consumption, selection of modulation/coding and user experience, etc. We explore the use of *new* cross layer opportunities to achieve secrecy and efficiency of data transmission in wireless networks. In the first part of this dissertation, we build secret key establishment methods for private communication between wireless devices using the *spatio-temporal variations* of *symmetric-wireless channel measurements*. We evaluate our methods on a variety of wireless devices, including *laptops, telosB sensor nodes, and Android smartphones*, with diverse wireless capabilities. We perform extensive measurements in real-world environments and show that our methods generate high entropy secret bits at a significantly faster rate in comparison to existing approaches.

While the first part of this dissertation focuses on achieving secrecy in wireless networks, the second part of this dissertation examines the use of special pulse shaping filters of the filterbank multicarrier (FBMC) physical layer in reliably transmitting data packets at a very high rate. We first analyze the *mutual interference power* across subcarriers used by different transmitters. Next, to understand the impact of FBMC beyond the physical layer, we devise a distributed and adaptive *medium access control protocol* that coordinates data packet traffic among the different nodes in the network in a best effort manner. Using extensive simulations, we show that FBMC consistently achieves an *order-of-magnitude performance improvement* over orthogonal frequency division multiplexing (OFDM) in several aspects, including *packet transmission delays, channel access delays*, and *effective data transmission rate* available to each node in static indoor settings as well as in vehicular networks.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Sneha Kumar Kasera, who laid the foundation for my research experience. He taught me how to conduct research, the art of writing papers, and more importantly, how to give technical presentations. He introduced me to work on several interesting problems, including secret key extraction, dynamic spectrum access networks, and radio tomographic imaging. He immensely helped in building my resume by presenting me with opportunities to serve as a teaching assistant for his Computer Networks class, to present lectures in his Network Security classes, to intern at Bell Labs, and to serve as a reviewer for various top journals and conferences. Through his help, I had an opportunity to participate in the prestigious ACM MobiCom09 conference, which also paved the way for obtaining a prize in the student research competition at that conference. Second, I would like to thank Professor Neal Patwari for his constant support, for his help with all my papers, for always being available to answer any question, and for inspiring me to continue with my research. His meticulously designed assignments in the Random Processes class, in particular, have helped me in visualizing seemingly hard-to-understand concepts and have greatly shaped my thinking on the secret key extraction research and beyond. Next, I would like to thank Professor Behrouz Farhang-Boroujeny for his inputs and numerous thought provoking discussions throughout my work on the dynamic spectrum access networks problem. His constant motivation and insightful weekly reviews of my plots and results ultimately resulted in three publications. Finally, I would like to thank my committee members, Professor Rajeev Balasubramonian and Professor Robert Ricci, for their valuable inputs and suggestions on my research problems.

I would like to thank my collaborators. I enjoyed working with Suman Jana throughout his stay in Utah. Collaboration with Suman resulted in the publication

of my first paper, which also represented a major stepping stone for my secret key extraction research. Collaborations with Daryl Wasden and Jessica Croft resulted in obtaining solid real-world measurements data, further boosting the research contributions of my papers. During the recent few semesters, I have gained significant research experience by working with Prarthana Gowda, Dustin Maas, and Peter Hillyard.

I would like to thank my previous and current lab mates, Jun Cheol Park, Junxing Zhang, Arijit Banerjee, Saurav Muralidharan, Mojgan Khaledi, Hema Bhatia, for numerous conversations. Finally, I would like to thank Ann Carlstrom and Karen Feinauer for handling all my paper work and keeping me up to date with my due progress requirements through all these years.

Above all, I would like to thank my family and friends for their constant support and motivation.

CHAPTER 1

INTRODUCTION

Cross layer system design represents a paradigm shift that breaks the traditional layer-boundaries in a network stack to achieve desirable characteristics [1, 2]. Exploiting information across different layers offers numerous avenues for enhancing a wireless network. Some of the primary benefits of cross layer design include optimization/improvement in terms of packet scheduling [3], error correction [4], multimedia quality and power consumption [5], selection of modulation and coding [6], user experience [2], etc. In this work, we explore the use of *new cross layer opportunities* to achieve secrecy and efficiency of data transmission in wireless networks. Our systems-oriented, cross layer research enables pervasive wireless devices to efficiently establish private communication channels that are secure from adversaries with unlimited computational power. In addition, our work also enables these devices to efficiently utilize the available wireless spectrum. Our research work demonstrates how theoretical concepts can be transformed into real-life systems, which in turn can serve as a strong foundation for building innovative, mobile systems and applications.

In the first part of this dissertation, we build secret key establishment methods for private communication between wireless devices using the *spatio-temporal variations* of *symmetric-wireless channel measurements*. Using *physical layer measurements* of the wireless channel characteristics between any two nodes, we can establish a *secure upper layer communication channel* between these two nodes. We evaluate our methods on a variety of wireless devices, including *laptops, telosB sensor nodes*, and *Android smartphones*, with diverse wireless capabilities under a variety of real-world environments.

In the second part of this dissertation, we investigate the use of different *physi-*

cal layer pulse shapes for *efficiency of data transmission*, depending on the *type of upper layer data traffic* and the acceptable level of *system computational complexity*. For example, when different nodes in the network exchange *synchronous traffic*, an efficient, low complex system can be built using the well-known *rectangular pulse shape* of the orthogonal frequency division multiplexing (OFDM) physical layer [7, 8]. On the other hand, when applications primarily exchange *best-effort traffic*, where the transmissions of different nodes are *not necessarily synchronized*, we can achieve efficiency of data transmission, as we demonstrate in this work, through the use of special *square root Nyquist (SR Nyquist [9]) physical layer pulse shape* at the expense of only a slight increase in system complexity; this system is referred to as *filterbank multicarrier (FBMC [10])*.

1.1 Achieving Secrecy in Wireless Networks

Secret key establishment is a fundamental requirement for private communication between two entities. Currently, the most common method for establishing a private communication channel is by using public key cryptography. However, there has been growing interest in finding alternatives to public key cryptography owing to concerns with the security of the public keys. Essentially, public key cryptography methods become insecure against an adversary with unlimited computational power; i.e., they do not provide information-theoretic security. Quantum cryptography [11, 12] is an alternative to public key cryptography, which allows two parties to establish a secret/symmetric key. While quantum cryptography applications have started to appear recently [13], they are still very rare and expensive. We explore a *less expensive* alternative for secret key establishment between wireless nodes, which is capable of producing arbitrarily long secret keys, and which when used as one-time pad, can provide security against adversaries with unlimited computational power.

At any point in time, the multipath properties of the wireless channel (gains, phase shifts, and delays) are *identical* [14] on both directions of a link, because the radio waves from each direction of the link traverse the same set of multipaths and undergo identical radio wave propagation effects (reflection, refraction, scattering, diffraction, etc.). The wireless channel characteristics change over time due to the movement of

either end of the link and/or any intermediate objects in the environment. The channel properties are unique to the locations of the two endpoints of the link. Therefore, an eavesdropper, who is a few wavelengths away from either endpoint, will measure a different, uncorrelated radio channel [15]. Thus, the wireless channel characteristics represent an inherent shared secret between any two wireless devices that can communicate. We exploit the physical layer measurements representing the wireless channel characteristics to establish a secret/symmetric key for private communication at an upper layer between these devices. Secret key establishment using wireless channel variations can provide *information-theoretic security*, i.e., security against adversaries with unlimited computational power, given that the reciprocal channel between the legitimate nodes is statistically independent of the channel between an eavesdropper and a legitimate node [16]. Figure 1.1 depicts the spatio-temporal and symmetric variations of received signal strength measurements between a pair of wireless nodes.

1.2 Efficient Dynamic Spectrum Access Networks

Due to the rapid proliferation of wireless devices in recent times, the available frequency spectrum space is very heavily used. Most of the existing wireless network technologies, e.g., 802.11, operate on a fixed set of channels or frequencies over a given bandwidth. Given this, certain portions of the wireless spectrum are heavily crowded in comparison to others. Thus, next-generation wireless networks have an opportunity to dynamically choose portions of the available spectrum that are under-utilized, or alternatively, avoid those parts of the spectrum that are over-crowded. Our research envisions the building of efficient dynamic spectrum access networks, where multiple nodes compete to utilize a shared frequency spectrum, as shown in Figure 1.2. While such networks have been gaining widespread attention in recent years, building such networks presents numerous challenges in terms of *achieving synchronization*, *minimizing interference* between the transmissions of different nodes and *system computational complexity*.

For efficiency of data transmission, we choose a suitable *physical layer pulse shape* on the basis of the *traffic type* and acceptable/desired level of *system complexity*. Specifically, when applications primarily exchange *best-effort traffic*, where the trans-

missions of different nodes are *not necessarily synchronized*, we have proposed to build an efficient system through the use of special *square–root Nyquist pulse shape* at the physical layer at the expense of only a slight increase in system complexity; this system is referred to as *filterbank multicarrier (FBMC [10])*. FBMC, which promises very low out-of-band energy of each subcarrier signal when compared to OFDM, is evaluated in our work to understand its ability in reliably transmitting *upper layer data packets* at a very high rate. We first analyze the *mutual interference power* across subcarriers used by different transmitters. Next, to understand the impact of FBMC beyond the physical layer, we devise a distributed and adaptive *medium access control protocol* that coordinates data packet traffic among the different nodes in the network in a best-effort manner.

1.3 Challenges

We address the following significant challenges in this dissertation.

1.3.1 Secret Key Extraction

When Alice and Bob collect channel measurements for secret key establishment, their measurements may exhibit minor asymmetries due to noise, interference, hardware limitations, manufacturing variations, vendor-specific differences in implementing automatic gain control, and the inability in sampling the channel simultaneously at both Alice and Bob with time-duplex transceivers. The main challenge is in making Alice and Bob agree upon the same bit sequence while ensuring that during the process of reconciling any potential bit mismatches, only a minimal amount of information is leaked over the insecure public channel.

There may be short-term correlation between subsequent bits when Alice/Bob happen to probe the channel more than once within the coherence time, where coherence time represents an interval over which the measurements remain predictable. However, it is extremely difficult to estimate the coherence time due to the presence of unpredictable movements of different objects in a real environment. Furthermore, the information reconciliation stage reveals a certain fraction of bits to reconcile the potential differences between Alice's and Bob's bitstreams. An adversary can take advantage of the leaked bits, and guess portions of the extracted key. Therefore, the

main challenge is in eliminating redundant/leaked information from the extracted bit sequence so that the output bit sequence has high entropy bits.

Secret key extraction performance may vary depending on the type of environment since the rate at which the channel changes depends on the movement of different objects in environment. So, the key challenge is in finding the type of settings that are well suited for secret key extraction – i.e., those settings which can produce secret bits at a fast rate, while at the same time, not being vulnerable to certain new types of attacks, which we demonstrate in this work.

When two groups of nodes, instead of only a pair of nodes, are used for secret key generation with the goal of improving the secret key generation performance, the asymmetry between the measurements of Alice and Bob increases due to the simultaneous reduction in the channel sampling rate and increase in the time gap between the bidirectional measurements of Alice and Bob. In other words, the use of multiple nodes can negatively impact secret key extraction and the main challenge is to reduce the increase in asymmetry even under decreased sampling rate and increased time gap between the bidirectional measurements.

1.3.2 Dynamic Spectrum Access Networks

When we have multiple nodes, which possibly belong to different administrative entities, compete to utilize the shared frequency spectrum, it is likely that the simultaneous transmissions of different nodes are not synchronized. However, the well-known solution, i.e., OFDM, for sharing the spectrum requires perfect synchronization between different nodes, and a lack of synchronization will cause significant mutual interference. In other words, OFDM, which is designed for exchanging synchronous traffic, will have a poor cross layer performance when the nodes exchange best-effort traffic, as we demonstrate in this work. Thus, the important challenge is to find an alternative physical layer that is capable of sharing the spectrum *without a centralized control* and also to develop a new medium access control protocol that can share the spectrum among various nodes in a distributed manner.

While there are a large number of OFDM systems available commercially, currently, there exists no practical system implementation with the FBMC PHY layer. This poses a significant challenge in evaluating FBMC for dynamic spectrum access.

Thus, we need to develop a new alternative method of evaluation that is capable of extensively modeling both the PHY and MAC layer aspects, at very small time-scales such as micro seconds, for a very realistic evaluation of the cross layer performance of FBMC.

1.4 Contributions

Our key contributions in this dissertation include the following.

1.4.1 Secret Key Extraction

We develop an environment adaptive secret key extraction scheme that works in conjunction with techniques that we have borrowed from quantum cryptography, namely information reconciliation [17] and privacy amplification [18], which can generate high entropy bits at a high rate.

We evaluate secret key extraction in different environments using off-the-shelf 802.11 devices and find the environment that is best suited for secret key extraction. Our experimental results show that (i) in certain environments, due to lack of variations in the wireless channel, the extracted bits have very low entropy, making these bits unsuitable for a secret key; and (ii) in dynamic scenarios where the two devices are mobile, and/or where there is a significant movement in the environment, high entropy bits are obtained fairly quickly.

We demonstrate a new form of attack on secret key extraction known as predictable channel attack where an adversary can cause the key establishing parties to extract a predictable secret bit sequence. The adversary can cause this attack by controlling the movements of intermediate objects in the environment. In other words, the adversary can break the secret key extraction mechanism without spending any computational power.

We explore the use of two groups of TelosB sensor nodes for secret key extraction. We show that when two groups of nodes collaborate in exchanging probe packets for collecting RSS measurements, they can extract stronger secret keys in an efficient manner. We also show that the collaborating nodes can improve the performance further when they exploit both *space* and *frequency diversities*.

We develop an analytical method for effectively removing the effects of shadow fading so that the secret key bits obtained are mainly due to the hard-to-predict effects of fast-fading or small-scale fading. We show that applying a running average filter to the channel measurements effectively removes the effects of shadow fading, which are caused by obstructions in the environment. This establishes that the extracted secret bits are primarily due to the hard-to-predict effects of fast-fading (or small-scale fading), which are caused because of the relative motion between the radios and the different objects in the environment. We use the well-known Gudmundson statistical model for shadow fading signals to show that it is essentially a low pass filter whereas the running average filter is a high pass filter. Our results show that depending on the speed of nodes and the sampling rate, if we appropriately choose the size of the running average window, we can significantly reduce the effects of shadow fading.

1.4.2 Dynamic Spectrum Access Networks

We derive an analytical expression for the interference power at a receiver due to mutual interference across subcarriers used by different transmitters. We express the mutual interference power on a subcarrier at a receiver node as a function of the subcarrier indices in which the desired transmitter and the interferer transmit their signals as well as of the wireless channel between the interferer and the receiver nodes.

We design and evaluate a new medium access control (MAC) layer that promises a fair share of available spectrum to each node. Our distributed and adaptive MAC protocol coordinates data packet traffic among the different nodes in the network in a best-effort manner. It adapts the size of the channel (i.e., number of subcarriers) of each node depending on the packet transmission success rate, and on how the current channel access delay compares with the historic average delay. Our MAC protocol increases or decreases the number of subcarriers in an additive increase and multiplicative decrease (AIMD) manner with the aim of achieving fair use of subcarriers across multiple nodes.

We evaluate the cross layer performance of FBMC and OFDM in static, indoor settings as well as dynamic settings in vehicular networks. Our work highlights the cross layer performance of FBMC in achieving *order-of-magnitude* improvement over OFDM in terms of transmission delay, channel access delay and data rate available

at the MAC layer.

We express our thesis statement in a concise form as follows. We can achieve secrecy and efficiency of data transmission in wireless networks by exploiting new cross layer opportunities.

The rest of this dissertation proposal is structured as follows. Chapter 2 describes our work on secret key extraction for a pair of devices using WiFi received signal strength measurements in real environments. Chapter 3 presents our work on using two groups of wireless sensor nodes for extracting stronger secret keys in an efficient manner. In Chapter 4, we describe the use of filterbank multicarrier communication system in conjunction with our AIMD MAC protocol for efficient use of the available spectrum in dynamic spectrum access networks.

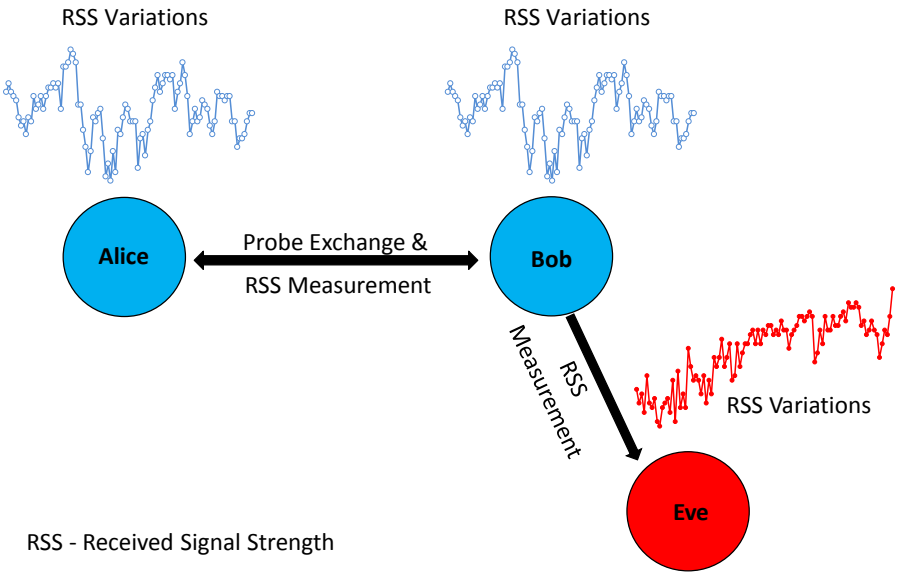

Figure 1.1. Spatio-temporal and symmetric variations of received signal strength measurements.

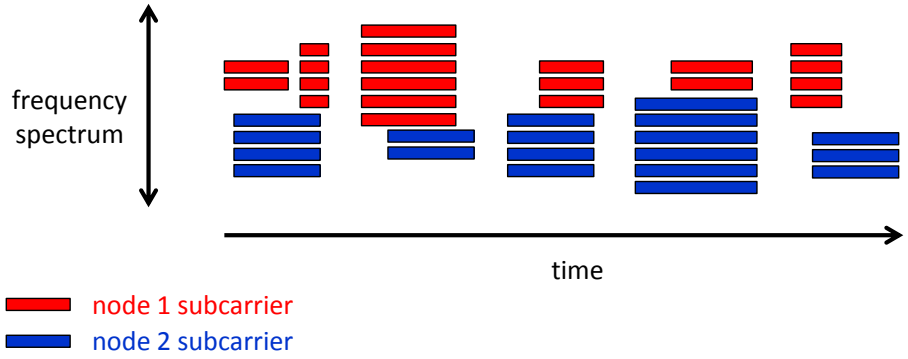

Figure 1.2. Dynamic spectrum access using multicarrier communication system. Number of subcarriers per node can be changed depending on traffic.

CHAPTER 2

SECRET KEY EXTRACTION FROM WIFI RECEIVED SIGNAL STRENGTH MEASUREMENTS IN REAL ENVIRONMENTS

2.1 Overview

Secret key establishment is a fundamental requirement for private communication between two entities. Currently, the most common method for establishing a secret key is by using public key cryptography. However, public key cryptography consumes a significant amount of computing resources and power that might not be available in certain scenarios (e.g., sensor networks). More importantly, concerns about the security of public keys in the future have spawned research on methods that do not use public keys.

Quantum cryptography [11, 12] is a good example of an innovation that does not use public keys. It uses the laws of quantum theory, specifically Heisenberg's uncertainty principle, for sharing a secret between two end points. Although quantum cryptography applications have started to appear recently [13], they are still very rare and expensive.

A less expensive and more flexible solution to the problem of sharing secret keys between wireless nodes (say Alice and Bob) is to extract secret bits from the inherently random *spatial and temporal variations* of the *reciprocal wireless channel* between them [19, 20, 21, 22, 16]. Essentially, we exploit the following properties of the wireless channel for secret key extraction:

- Reciprocity of radio wave propagation: At any point in time, the multipath properties of the radio channel (gains, phase shifts, and delays) are identical

on both directions of a link, because the radio waves from each direction of the link traverse the same set of multipaths and undergo identical radio wave propagation effects (reflection, refraction, scattering, diffraction, etc.).

- Temporal variations in the radio channel: The multipath channel changes over time due to the movement of either end of the link and/or any intermediate objects in the environment.

- Spatial variations: The properties of the radio channel are unique to the locations of the two endpoints of the link. An eavesdropper, who is a few wavelengths away from either endpoint, will measure a different, uncorrelated radio channel [15].

Thus, any two wireless nodes that can communicate with each other inherently share a secret between them, and we use that as the basis for extracting the secret key bits. Secret key extraction using channel variations provides information-theoretic security given that the reciprocal channel between the legitimate nodes is statistically independent of the channel between an eavesdropper and a legitimate node [16]. In other words, our secret key establishment method is capable of producing arbitrarily long secret keys, which when used as one-time pad, can provide security against adversaries with unlimited computational power.

Received signal strength (RSS) is a popular statistic of the radio channel and can be used as the source of secret information shared between a transmitter and receiver. We use RSS as a channel statistic, primarily because of the fact that most of the current off-the-shelf wireless cards, without any modification, can measure it on a per frame basis. The variation over time of the RSS, which is caused by motion and multipath fading, can be quantized and used for generating secret keys. The mean RSS value, a somewhat predictable function of distance, must be filtered out of the measured RSS signal to ensure that an attacker cannot use the knowledge of the distance between key establishing entities to guess some portions of the key. These RSS temporal variations, as measured by Alice and Bob, cannot be measured by an eavesdropper (say Eve) from another location unless she is physically very close to Alice or Bob. However, due to nonideal conditions, including limited capabilities of

the wireless hardware, Alice and Bob are unable to obtain identical measurements of the channel. This asymmetry in measurements brings up the challenge of how to make Alice and Bob agree upon the same bits without giving out too much information on the channel that can be used by Eve to recreate secret bits between Alice and Bob.

Azimi-Sadjadi et al. [19] suggested using two well-known techniques from quantum cryptography, *information reconciliation* and *privacy amplification*, to tackle the challenge caused by RSS measurement asymmetry. Information reconciliation techniques (e.g., Cascade [17]) leak out minimal information to correct those bits that do not match at Alice and Bob. Privacy amplification [18] reduces the amount of information the attacker can have about the derived key. This is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, to transform the reconciled bit stream into a nearly perfect random bit stream.

Most of the previous research work on RSS-based secret key extraction, including that of Azimi-Sadjadi et al. [19], is based on either simulations or theoretical analysis. Other than the recent work by Mathur et al. [20] that was performed in a specific indoor environment, there is very little research on evaluating how effective RSS-based key extraction is in real environments under real settings. We address this important limitation of the existing research in this paper with the help of wide-scale real-life measurements in both static and dynamic environments. In order to perform our measurements and subsequent evaluations, we implement different RSS quantization techniques in conjunction with information reconciliation and privacy amplification.

We first collect measurements under different environments to generically evaluate the effectiveness of secret key generation. We find that under certain environments due to lack of variations in the channel, the extracted key bits have very low entropy, making these bits unsuitable for a secret key. Interestingly, we also find that an adversary can cause predictable key generation in these static environments. However, in scenarios where Alice and Bob are mobile, and/or where there is a significant movement in the environment, we find that high entropy bits are obtained fairly quickly. Next, building on the strengths of the existing schemes, we develop an environment adaptive secret key generation scheme that uses an adaptive lossy

quantizer in conjunction with Cascade-based information reconciliation and privacy amplification. Our measurements show that our scheme performs the best in terms of generating high entropy bits at a high bit rate in comparison to the existing ones that we evaluate. The secret key bit streams generated by our scheme also pass the randomness tests of the NIST test suite [23] that we conduct.

The rest of this chapter is structured as follows. In Section 2.2, we outline our problem setup. We describe the secret key extraction process in Section 2.3 and present our adaptive secret bit generation method in Section 2.4. Section 2.5 describes our implementation. We present the characteristics of real-world received signal strength measurements under a diverse set of environments in Section 2.6. We compare different secret key extraction approaches in Section 2.7. We evaluate our multiple bit extraction method in Section 2.8. We evaluate secret key extraction using handheld devices in Section 2.9. We present the related work in Section 2.10 and summarize our findings in Section 2.11.

2.2 Problem Setup

In our problem setup, there are two wireless nodes, Alice and Bob, that need to establish a shared/secret key. We assume that the adversary, Eve, can listen to all the communication between the wireless nodes representing Alice and Bob. Eve can also measure all the channels between herself and Alice and Bob at the same time when Alice and Bob exchange probes and measure the channel between themselves. Such a passive adversary model has been widely used/adopted in existing measurements-based work on secret key extraction (e.g., [14, 24]). We also assume that Eve knows the key extraction algorithm and the values of the various parameters used in the algorithm. However, we assume that Eve is not very close (less than a few multiples of the wavelength of the radio waves being used; for example, the wavelength of signals in the 2.4 GHz band is approximately 12.5 cm) to either Alice or Bob while they are exchanging probes. This ensures that Eve measures a different, uncorrelated radio channel [15]; [24] experimentally show that there is little mutual information between Eve and Alice/Bob; they also show that the information obtained by Eve is negligible even if she possesses multiple antennas. We assume

that Eve does not know the exact positions of Alice, Bob, and every intermediate object at every possible time instant, and also does not know the electrical/magnetic properties of every possible object in the environment; this will prevent Eve from precharacterizing the wireless channel characteristics and later use sophisticated approaches like ray tracing to deduce the channel variations seen by Alice/Bob when they collect the measurements. We assume that Eve is not interested in disrupting the key establishment, i.e., she neither jams the communication channel between Alice and Bob, nor does she modify any messages exchanged between Alice and Bob. However, Eve is free to move any intermediate object between Alice and Bob and affect the communication channel between them, although we assume that Eve cannot restrict the movement of other objects in the channel and thus will not be able to significantly increase the coherence time of the channel. In our earlier work [14], we showed that in static scenarios, when Eve positions herself strategically on the signal path between Alice and Bob, she can cause the predictable channel/key generation attack. To avoid such attacks, in this work, we only consider dynamic scenarios[1], where Eve is incapable of causing predictable channel variations. We also assume that Eve cannot cause a person-in-the-middle attack, i.e., Alice and Bob are not authenticated. Hence, our proposed scheme works against passive adversaries. The Diffie-Hellman secret key establishment scheme has found widespread use in network security protocols and standards (e.g., for providing perfect forward secrecy, strong password protocols, etc.) even without an authentication mechanism [25]. We believe that our scheme will provide a strong alternative to the Diffie-Hellman scheme in wireless networks. There is a growing amount of literature in authenticating wireless devices based on their physical and radiometric properties (e.g., [26, 27]). These and future authentication mechanisms can be used in conjunction with our efficient high rate secret key establishment scheme.

2.3 Background on Secret Key Extraction Process

Existing secret key extraction approaches convert a set of RSS measurements into a sequence of bits using a quantization stage. However, due to nonideal conditions,

[1]We also show that dynamic scenarios are best for key extraction in [14].

including limited capabilities of the wireless hardware, Alice and Bob are unable to obtain identical measurements of the channel. This asymmetry in measurements brings up the challenge of how to make Alice and Bob agree upon the same bits without giving out too much information on the channel that can be used by Eve to recreate secret bits between Alice and Bob. Azimi-Sadjadi et al. [19] suggested using two well-known techniques from quantum cryptography, *information reconciliation* and *privacy amplification*, to tackle the challenge caused by RSS measurement asymmetry. Information reconciliation techniques (e.g., Cascade [17]) leak out minimal information to correct those bits that do not match at Alice and Bob. Privacy amplification [18] reduces the amount of information the attacker can have about the derived key. This is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, to transform the reconciled bit stream into a nearly perfect random bit stream. In our work, we also use the information reconciliation and privacy amplification stages. Figure 2.1 depicts the process of wireless RSS-based secret key extraction. We describe the purpose of each stage as follows.

2.3.1 Quantization

To extract a secret key based on wireless channel variations, Alice and Bob begin by exchanging probe packets and measuring RSS values of the probes. After collecting enough measurements, each node quantizes its measurements to generate an initial bitstream. The quantization is done using specified thresholds. Figure 2.2 shows a sample quantizer with two thresholds. Alice and Bob perform the following steps in the quantization stage - (i) define two quantization thresholds q^+ and q^- such that $q^+ = \mu + (\alpha \times \sigma)$ and $q^- = \mu - (\alpha \times \sigma)$, where μ and σ represent the running average and standard deviation over a window of RSS measurements, and $\alpha \geq 0$. (ii) Discard those measurements that lie between q^+ and q^- and maintain a list of indices of measurements that are discarded; exchange the indices list and only keep those measurements that both parties decide not to discard. (iii) Generate initial bitstreams by extracting a 1 or a 0 from each RSS measurement depending on whether the measurement lies above q^+ or below q^-. While this kind of single bit quantization

with two thresholds was introduced in [20], many other earlier works (e.g., [22, 16, 19]) also use some form of thresholding for performing single bit quantization.

2.3.2 Information Reconciliation

Once both Alice and Bob extract an initial bitstream by quantization, to agree upon the same secret key, they must correct the bits where their bitstreams differ. Differences arise primarily due to noise and interference, hardware limitations, manufacturing variations, vendor-specific differences in implementing automatic gain control, and the inability in sampling the channel simultaneously at both Alice and Bob with time-duplex transceivers.

Cascade [17] is an iterative, interactive information reconciliation protocol. In Cascade, one party (say Alice) permutes the bitstream randomly, divides it into small blocks, and sends permutation and parity information of each block to the other party (Bob). Bob permutes his bitstream in the same way, divides it into small blocks, computes parities, and checks for parity mismatches. For each mismatch, Bob performs a binary search on the block to find if a few bits can be changed to make the block match the parity. These steps are iterated a number of times to ensure a high probability of success.

Bloch et al. present an alternative reconciliation method that uses multilevel coding and optimized low density parity check codes [28]. However, they conclude that the memory requirements and the complexity of their method may be too high for embedded or low-cost systems. In this work, we only consider Cascade for information reconciliation.

2.3.3 Privacy Amplification

There may be short-term correlation between subsequent bits when the channel probing rate is greater than $\frac{1}{\text{coherence time}}$, where coherence time is defined as the time interval during which the channel measurements remain predictable. However, it is extremely difficult to estimate the coherence time due to the presence of unpredictable movements of different objects in a real environment. When two subsequent channel measurements occur within the coherence time, the bits extracted may exhibit short-term correlations. Further, the information reconciliation stage reveals a certain

fraction of bits to reconcile the differences between Alice's and Bob's bitstreams. An adversary can take advantage of the leaked bits, and guess portions of the extracted key. Therefore, it is necessary to remove those leaked bits.

Privacy amplification addresses these two problems by reducing the size of the output bitstream. It is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, and generate fixed size smaller length output bit streams from longer input bit streams. These methods are generally based on leftover hash lemma, which is a well-known technique for obtaining random bits from imperfect random sources [18]. The need for privacy amplification was recognized by [29].

2.3.4 Metrics for Comparing Different Key Extraction Approaches

We compare different secret key extraction approaches (e.g., Aono et al. [22], Tope et al. [16], Mathur et al. [20], Azimi-Sadjadi et al. [19]) for the quality of the bit streams they generate. This quality is quantified by three performance metrics -

1. **Entropy:** Entropy characterizes the uncertainty associated with a random variable. We estimate the entropy of a bit stream using NIST test suite's *approximate entropy test* [23].

2. **Bit mismatch rate:** We define the bit mismatch rate as the ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted from RSS quantization.

3. **Secret bit rate:** We define the secret bit rate as the average number of secret bits extracted per collected measurement. This rate is measured in terms of final output bits produced after taking care of bit losses due to information reconciliation and privacy amplification.

Note that the bit mismatch rate value we calculate is based on the bits we obtain immediately after the quantization step, and not after the privacy amplification step. In fact, the bit mismatch rate is expected to be zero after the information reconciliation step.

2.4 Adaptive Secret Bit Generation (ASBG)

Our experimental results in Section 2.7 suggest that some quantizers like those of Aono et al. or Tope et al. that aim to achieve high bit rate can output bit streams with low entropy in certain settings, especially in those that have minimal movement. On the other hand, some other quantizers like that of Mathur et al. can output bit streams with reasonably high entropy but sacrifice the bit rate to achieve this or vice versa. In summary, the existing approaches that use RSS measurements do not generate secret bits at a high rate and/or with high entropy. We develop a method, which we call Adaptive Secret Bit Generation (ASBG), that builds on the strengths of the existing approaches. In our method, we use a modified version of Mathur's quantizer [20] in conjunction with two well-known information reconciliation and privacy amplification techniques.

We first describe our quantizer and then identify the differences with Mathur's scheme. Our modified quantizer is described as follows. (i) Alice and Bob consider a block of consecutive measurements of size *block_size* which is a configurable parameter[2]. For each block, they calculate two adaptive thresholds $q+$ and $q-$ independently such that $q+ = mean + \alpha * std\_deviation$ and $q- = mean - \alpha * std\_deviation$, where $\alpha \geq 0$. (ii) Alice and Bob parse their RSS measurements and drop RSS estimates that lie between $q+$ and $q-$ and maintain a list of indices to track the RSS estimates that are dropped. They exchange their list of dropped RSS estimates and only keep the ones that they both decide not to drop. (iii) Alice and Bob generate their bit streams by extracting a 1 or a 0 for each RSS estimate if the estimate lies above $q+$ or below $q-$, respectively.

Our modified quantizer divides the RSS measurements into smaller blocks of size *block_size* and calculates the thresholds for each block separately. The adaptive thresholds allows our quantizer to adapt to slow shifts of RSS. Mathur et al. [20] subtract a running windowed average of RSS measurements before computing thresholds $q+$ and $q-$ to make their scheme adaptive to the slow variations of RSS. We also perform experiments to find the optimal block size. The results of these experiments are shown

[2]The Cascade block size is not related to the *block_size* we use for determining the quantization thresholds.

in Section 2.7. Unlike the Mathur quantizer that preserves only a single bit from m consecutive 1s or 0s and drops the other repeating $m-1$ bits, our modified quantizer extracts a bit out of each measurement that falls above the upper threshold or below the lower threshold but depends on the privacy amplification step to remove the effect of correlated bits.

Various single bit quantization methods drop a large amount of RSS samples that lie in between the upper and lower thresholds. These dropped samples constitute a loss of valuable information that can be used by Alice and Bob to generate secret bits and also result in an inefficient utilization of the wireless medium because more probes must be sent and received. Furthermore, privacy amplification also reduces the secret bit rate while increasing entropy. To increase the secret bit rate, we propose an adaptive scheme for extracting multiple bits from a single RSS measurement. Our multiple bit extraction scheme is described as follows.

Once Alice and Bob collect the RSS measurements, they perform the following steps - (i) determine the *Range* of RSS measurements from the minimum and the maximum measured RSS values; (ii) find N, the number of bits that can be extracted per measurement, where $N \leq \lfloor \log_2 Range \rfloor$; (iii) divide the *Range* into $M = 2^N$ equal sized intervals; (iv) choose an N bit assignment for each of the M intervals (for example, use the Gray code sequence [30]); and (v) for each RSS measurement, extract N bits depending on the interval in which the RSS measurement lies. After completing the above steps, as in the single bit extraction case, Alice and Bob use information reconciliation to correct the mismatching bits, and finally, apply privacy amplification to the reconciled bit stream and extract a high entropy bit stream.

Our results, as presented in Section 2.7, show that our single bit extraction in conjunction with information reconciliation and privacy amplification is able to achieve higher entropy in comparison to existing schemes, and our multiple bit enhancement (evaluated in Section 2.8) allows us to significantly increase the secret bit rate as well.

2.5 Implementation

We implement our key extraction scheme on two laptops (Alice and Bob) equipped with built-in Intel PRO/Wireless 3945ABG wireless network cards, operating in the

802.11g mode. Both laptops run the Ubuntu Linux operating system. In order to establish a secret key, Alice and Bob exchange probe packets periodically and use these probe packets to measure the RSS values.

In our implementation, we use specially crafted 802.11 management frames as probe packets. We prefer to use management frames as a communication mechanism over standard data frames because in the case of data frames, acknowledgement frames are sent by the receiving wireless card. On the other hand, in the case of management frames, no acknowledgement frame is sent by the receiving wireless card. Moreover, management frames are prioritized over data frames and are queued separately. These facts motivate us to design our own acknowledgement scheme using management frames instead of data frames to better control the probing rate. In our implementation, among the different management frames, we choose to use the *beacon* frames for the communication between the initiator and the responder. However, data frames could be used opportunistically as well since, in this case, we can obtain RSS measurements without sending extra traffic in the network.

The sequence number field of beacon packet is used as our protocol's sequence number to handle packet loss and retransmissions. We use raw packet injection in the *monitor* mode to send these specially crafted beacon frames. We utilize *ipwraw* [31], a wireless card driver for Intel 3945 cards, for raw packet injection. We also use the monitor mode to receive the beacon frames. In any other mode (e.g., the AP, or STA mode), the wireless device driver does not forward these frames to any upper layer applications. In our implementation, the endpoints exchange beacon frames at a rate of approximately 20 frames per second, and measure the RSS values on a per-frame basis. The RSS measurements we collect are reported by ipwraw driver in the radio tap header of each received frame [32].

We implement our key extraction scheme in a modular way so that different methods of performing quantization, information reconciliation, or privacy amplification can be put together to build different schemes using the same basic framework. To compare the performance of different quantizers, we implement them as pluggable modules to our key extraction scheme. For information reconciliation, we use the well-known interactive Cascade [17] protocol. For privacy amplification, we use the

2-universal hash family of functions. Alice and Bob use these hash functions to generate the output secret bits. We describe our implementation in greater detail in our work [14].

We also use an Atheros-based card to evaluate the effect of heterogeneous hardware on the key extraction process. We present the results that we obtain using the Atheros card in Section 2.6.5.

2.6 Measurements

In this work, we observe the variations in the wireless channel through measurements of RSS on a per frame basis. An RSS measurement represents the average of the energy arriving during the preamble sequence. The wireless card drivers report the RSS values as integers, and the calculation of RSS is vendor dependent. For example, Atheros devices report RSS values from -35 dB to -95 dB, Symbol devices report RSS values from -50 dB to -100 dB, in 10 dB steps, and Cisco devices report RSS values in the range -10 dB to -113 dB [33]. We use Intel-based wireless cards for all of our experiments except one experiment with heterogeneous devices, in which we also an Atheros-based wireless card. Each of our RSS measurements is quantized to produce one or more bits, depending on the quantization scheme used, and forms the basis for key extraction.

We conduct our experiments in a wide variety of environmental settings and under different scenarios (with and without mobility of endpoints/intermediate objects, etc.). The environments considered include an underground concrete tunnel, a typical office building, and different outdoor environments. The primary goal of these experiments is to find the type of settings that are best suited for secret key extraction and also to evaluate the capability of our secret key extraction approach in producing bit streams with high entropy, minimal number of mismatched bits between Alice and Bob, and at a fast rate.

2.6.1 Stationary Endpoints and Intermediate Objects

2.6.1.1 Experiment A: Underground Concrete Tunnel

We perform our first experiment inside an underground concrete tunnel that runs between two engineering buildings inside the University of Utah campus. The concrete

tunnel provides an environment that is free from most of the external interference sources, and the effects of mobility of any objects in the environment. Therefore, even though this is an atypical environment, it provides us the opportunity to study the amount of channel variation observed in a completely stationary environment. The two laptops are separated by a distance of about 10 feet during the experiment. Figure 2.3 shows the variations in RSS measurements collected by Alice and Bob. As expected, there are not many noticeable variations in the channel - at each instant, the RSS values vary only as much as 2 dB from the mean. We also note that the curves for Alice and Bob do not follow each other, indicating a channel with low reciprocity. This happens because the variations in a static channel are primarily generated by hardware imperfections and thermal effects which are nonreciprocal. RSS measurements in this type of environment contain very low inherent entropy. Therefore, it is not possible to extract secret bits at a fast rate in this type of setting.

2.6.1.2 Experiment B: Gallery in the Engineering Building

Next, we perform RSS measurements in an indoor setting in one of the engineering buildings. This experiment is done on a holiday evening to ensure that the gallery is mostly empty and there is minimal external movement. Note that unlike Experiment A, this setting has normal interference effects caused by other wireless devices operating in the vicinity. This setting allows us to study the channel variations with laptops separated by larger distances ($\sim$ 30 feet), in a relatively calm indoor environment. Figure 2.4 shows the variations in RSS measurements made by Alice and Bob. We find that like our tunnel experiment, Alice's and Bob's measurements are significantly different, indicating a very low channel reciprocity. The nonreciprocity of the channel is primarily due to the large distance between the laptops. When the distance between Alice and Bob becomes large, the channel measurements are dominated by random thermal noise and different interference sources affecting each laptop in a different manner. Like the tunnel scenario, it is not possible to extract secret bits at a fast rate in this type of setting either.

2.6.1.3 Experiment C: Lawn inbetween the Cafeteria and Library

We perform this experiment on a calm, windless day with minimal external movement on a lawn under the trees inbetween the cafeteria and the library. The distance between the laptops is about 10 feet. Figure 2.5 shows the RSS measurement variations as seen by Alice and Bob, respectively. In this figure, due to the stationary settings, we only find infrequent, small-scale variations in the channel measurements. This experiment shows that low-reciprocity is not just a characteristic of the indoor environments; it can occur even in typical stationary outdoor environments. Similar to the first two experiments, this type of setting is also not conducive to fast secret bit extraction.

2.6.2 Mobile Endpoints

2.6.2.1 Experiment D: Walk Inside an Engineering Building

To examine the effect of mobility of nodes in indoor environments, we carry around two laptops at normal walking speed on the third floor of an engineering building and perform RSS measurements. The laptops are carried along the corridors in the third floor in such a way that one trails the other and are separated by a distance of 10-15 feet for the most part[3]. Figure 2.6 depicts the variations in RSS values measured by Alice and Bob. As we can clearly observe, unlike previous experiments, the channel varies often with a wide variation window (-49 dB to -73 dB) and with a high degree of reciprocity. This experiment shows that mobility in indoor settings can help achieve fast secret key extraction from RSS measurements by increasing the inherent entropy of the measurements and by improving the reciprocity of the channel.

2.6.2.2 Experiment E: Walk from an Engineering Building to the Cafeteria

We perform an experiment by carrying two laptops while walking at a normal speed from an engineering building to the cafeteria along two parallel streets. For most of the experiment, the laptops are separated by a distance of about 20-25 feet. The results of this experiment are shown in Figure 2.7. As we can see, the

[3]Except for the very initial phase of our experiments, and/or when there is intervening traffic in our paths during the experiment, the specified distance is maintained.

measurements show a wide range of variation. The channel variation window is from −49 dB to −76 dB. We also note that like the measurements while walking inside the engineering building, the RSS measurements in this experiment also show a high degree of reciprocity. This shows that the outdoor environment combined with mobility causes a significant increase in the variation of the channel and improves its reciprocity. Consequently, we can expect a significant increase in the secret bit rate compared to the stationary experiments.

2.6.2.3 Experiment F: Bike Ride on City Streets

To evaluate the effect of nodes moving faster than normal walking speeds on the channel variation, we perform an outdoor experiment while we go on a bike tour on city streets. With one bike trailing another, a distance of 10 feet or more is maintained for most of the bike ride. As expected, this outdoor experiment exhibits the widest variations (−35 dB to −70 dB) in the channel, as shown in Figure 2.8. The bikes moving at a higher speed compared to walking create an even faster changing channel. As in the previous two cases, this environment also results in a highly reciprocal channel. These two factors together help in achieving a higher secret bit generation rate.

2.6.3 Mobile Intermediate Objects

2.6.3.1 Experiment G: Crowded Cafeteria

As we find in our previous experiments that mobile nodes result in a variable and highly reciprocal channel, we expect to observe similar effects if we have moving intermediate objects in the environment between the nodes instead of the nodes moving themselves. To verify this, we first perform an experiment where we study the effects of randomly moving intermediate objects at low speed. We conduct this experiment during a busy lunch hour in a crowded cafeteria. We keep our laptops stationary on two tables separated by a distance of 10 feet across the main entrance of the cafeteria. In this setting, we see many people frequently walk between these two tables. The channel variations measured by Alice and Bob are shown in Figure 2.9. As expected, even though the laptops are stationary, the random movements of people inbetween causes channel variations comparable to the last three experiments with

mobile endpoints.

2.6.3.2 Experiment H: Across a Busy Road

We perform another experiment to examine the effect of fast moving intermediate objects between two stationary nodes on the RSS measurements. We conduct this experiment across a busy road adjacent to the engineering building. In this experiment, the vehicles on the road move at high speeds ($\sim$ 30-40 mph). Our laptops are stationary and are separated by a distance of about 25 feet across the road. This environment causes the nodes to experience the highest packet loss rate compared to all the previous experiments. We expect the channel variations to be larger than the previous measurement as the intermediate objects are moving at a faster rate in this case. However, Figure 2.10 shows that the channel variation window is smaller (-70 dB to -77 dB) than the cafeteria case (Experiment G). Notice that the channel variation and reciprocity in Experiment H are still high compared to the pure stationary environment with a similar distance between the two laptops (Experiment B) and hence will result in secret key extraction at a faster rate.

2.6.4 Predictable Channel Attack

As mentioned earlier, stationary environments cannot support fast secret key extraction. However, another significant drawback of stationary environments is that an adversary can use planned movements in such environments, causing desired and predictable changes in the channel between the actual sender and receiver nodes.

We conduct two experiments to show that the adversary can, in fact, cause desired changes in the channel between the sender and receiver by controlling the movements of some intermediate object or of the actual radios. The first experiment is conducted in a student lab in one of the engineering buildings with two laptops; the separation between the two laptops is about 10 feet and the intermediate object is moved at about the halfway point inbetween the laptops.

The schematic of the first experiment is shown in Figure 2.11. A person (say X), sitting on a chair and intermittently leaning backward and forward, takes the role of the intermediate object. Sitting on the chair, whenever X leans backward obstructing the line of sight path, the RSS drops, and whenever X leans forward so

that there is no obstruction along the line of sight path, the RSS regains its original value. Figure 2.12 shows the variations of the RSS values and the pattern of variation follows the movements of X. Under these circumstances, when any key extraction scheme is used on such a data set, it produces a predictable pattern of secret bits.

For the RSS values shown in Figure 2.12, our quantization scheme actually generates an alternating sequence of multiple 0s and 1s, e.g., 0000111100001111 .... Alice and Bob could possibly use random subsampling of the bit sequence, as in [20], or use privacy amplification, to ensure that the resulting bit pattern is random. However, if an adversary is able to completely control the bit sequence coming out of the quantization process, then no postprocessing technique will be able to ensure the security of the resulting bit sequence. Consequently, it is important to weigh the relationship between the adversary's ability to control the environment and the block size used in subsampling or privacy amplification.

In the second experiment, we use a laptop (receiver) and a wireless router (sender) such that they are separated by about 5 feet. The wireless router periodically sends beacon packets that are received by the laptop. While resting the hinges of the laptop on a flat table, we move the laptop back and forth so that the leading edge of its base goes up and down. Again, as in the first experiment, the RSS values follow a pattern similar to Figure 2.12.

It is very important to note that we obtain the above results even with coarse movements, without the use of any precision machinery to create the movements. Thus, our experiments demonstrate that it is quite easy for an adversary to launch a "predictable channel" attack in a stationary environment and cause desired changes in the channel between the sender and receiver, making them extract a predictable sequence of secret key bits. One of the possible ways to avoid this attack is to use the RSS measurement-based secret extraction scheme only in places where multiple moving objects are present so that the attacker's movement alone will not be able to change the channel predictably. The effectiveness of the predictable channel attack on key extraction methods using other channel characteristics (e.g., channel impulse response) will be explored in the future.

2.6.5 Heterogeneous Devices

The experiments described so far use identical hardware for both transmitter and receiver. However in reality, different users could have different hardware. To investigate the effects of using heterogeneous devices, we perform an experiment in a setting similar to that of Experiment D (walk inside an engineering building). For this experiment, Alice is equipped with an Intel 3945 ABG card and Bob with an Atheros chipset-based card. Figure 2.13 depicts the variations in RSS values measured by Alice and Bob. We can clearly see that even with heterogeneous endpoints, the channel measurements exhibit a very high degree of reciprocity. Alice's RSS values range from -80 dB to -51 dB while Bob's RSS values range from -70 dB to -46 dB. We find that with heterogeneous hardware, when using our quantization method, the mismatch fraction between Alice's and Bob's bit streams is about 11%. In our implementation, information reconciliation can handle this mismatch rate. Therefore, even though heterogeneous hardware introduces higher bit mismatch rates than using homogeneous ones, we can still perform secret key extraction with reasonable efficiency.

2.6.6 Summary of Measurements

In summary, the environments with stationary endpoints and stationary intermediate objects exhibit small-scale variations in the wireless channel. Comparatively, environments with mobile endpoints exhibit a much wider variation in the channel. The small-scale variations (for example, -55 dB to -57 dB in Experiment A) in static settings are mainly due to variations in the hardware and random noise. On the other hand, the large-scale variations in the mobile settings (for example, -35 dB to -70 dB in Experiment F) are primarily caused by actual changes in the channel. Random noise due to the hardware are also present in the measurements taken in the mobile settings, but its effects are not large enough to affect the reciprocity of the channel. Therefore, stationary environmental settings yield much higher bit mismatch rates compared to mobile settings. Further, due to lack of enough variations, static settings also produce bit streams with very low secret bit rates. In short, mobility improves both secret bit rate and bit mismatch rate and hence, mobile environments

are better suited for the RSS measurement-based key extraction schemes.

An adversary can potentially guess the secret key established between the sender and receiver if the adversary, by some means, can affect the channel in a predictable way. Before applying the key extraction methods based on wireless channel characteristics, care must be taken to ensure that there is enough randomness in the environment so that an adversary cannot cause such attacks. One way to ensure this is to force Alice and/or Bob to move in a somewhat unpredictable manner while extracting secret keys. Environments including outdoor busy streets and crowded cafeterias are characterized by unpredictable relative motion between the sender, receiver, and the objects in the environment. These environments are most suitable for key extraction based on reciprocal and dynamic wireless channels.

2.7 Comparison of Key Extraction Approaches in Different Environments

In this section, we compare the performance of ASBG with other existing schemes in terms of entropy, secret bit rate, and bit mismatch rate. Although ASBG is capable of multiple bit extraction, we evaluate only single bit extraction in this section. We show that ASBG not only outputs a secret bit stream with the highest entropy but also the secret bit rate and bit mismatch fraction of ASBG are comparable, if not better than all the existing methods.

Various key extraction approaches that we compare in this work use one or more configurable parameters. We choose the parameters for all these quantization schemes such that they help strike a balance between the entropy and the secret bit rate. For the results shown in this section, we use the following configurable parameters. In the scheme of Aono et al., the configurable parameter β is chosen such that at most, 15% of the RSS measurements are deleted from the data set. The method of Tope et al. uses two thresholds - γ_l and γ_h. We choose $\gamma_l = avg\_of\_delta\_values + 0.4 * std\_deviation$, and $\gamma_h = avg\_of\_delta\_values + std\_deviation$. In the scheme of Mathur et al., two thresholds $q+$, $q-$ and m, the minimum number of measurements on an excursion above or below the thresholds, are used such that $q+ = mean + \alpha * standard\_deviation$ and $q- = mean - \alpha * standard\_deviation$. In order to remove the affects of slowly moving average signal power, as suggested in [20], we subtract a

windowed average from each RSS measurement. We choose $\alpha = 0.2$ and $m = 2$ to ensure that a large fraction of measurements is considered for bit extraction. We do not implement the random subsampling step because although this step improves the entropy of the extracted bit stream, it negatively impacts the secret bit rate. In the scheme of Azimi et al., a threshold value of 10 is used to determine the deep fades. When extracting one bit per measurement, ASBG uses two thresholds $q+$, $q-$ with $\alpha = 0.8$ and $block\_size = 25$. Figure 2.14 shows the variation of the bit mismatch rate with block size for our ASBG scheme. We observe that the mismatch rate gradually falls and becomes very small after a certain block size threshold and stays small even when the block size is increased beyond the threshold. We pick a block size (= 25) where the mismatch rate is low.

The performance of the different secret key extraction schemes is shown in Figures 2.15, 2.16 and 2.17. The scheme of Aono et al. has the highest secret bit rate. However, their scheme produces bit streams with very low entropy. On the other hand, the scheme of Mathur et al. generates bit streams with relatively high entropy at a moderate rate. Note that when a random sampling step is employed in the scheme of Mathur et al., the secret bit rate will be correspondingly lower than what we report in Figure 2.17. The scheme of Azimi-Sadjadi et al. results in bit streams with highest entropy. However, the bit rate of their scheme is very low. ASBG produces bit streams with highest entropy, like the scheme of Azimi-Sadjadi et al., while still maintaining the bit rate as high as the scheme of Mathur et al. In Figure 2.15, the plots corresponding to the scheme of Azimi-Sadjadi et al. and ASBG are one behind the other.

To ensure the randomness of the bit streams generated by ASBG, we also run randomness tests available in the NIST test suite [23]. There are a total of 16 different statistical tests in the NIST test suite. Of these 16 tests, we run only 8 tests. The bit streams that we obtain from our experiments meet the input size recommendation [23] of the 8 NIST tests only. We find that the ASBG-generated bit streams pass all the 8 tests. The results of these tests are shown in Table 2.1, Table 2.2, and Table 2.3. The remaining 8 tests require a very large input bit stream (specifically, 6 of the 8 remaining tests require $\approx 10^6$ bits).

We briefly describe the purpose of these statistical tests in the NIST test suite [23] as follows. The frequency test determines whether the number of ones and zeros in a sequence are approximately the same. The block frequency test checks whether the frequency of ones in a given M-bit block is approximately $M/2$. Using numeric values -1 and $+1$ in place of bits 0 and 1, the cumulative sums test determines whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of the cumulative sum for random sequences. The runs test verifies whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. The purpose of the longest run of ones (LRO) test is to determine whether the length of the LRO within the tested sequence is consistent with the length of the LRO that would be expected in a random sequence. The FFT test checks for periodic features that would indicate a deviation from the assumption of randomness. The approximate entropy test compares the frequency of overlapping blocks of two consecutive lengths against the expected result for a random sequence. The serial test determines whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.

Each of these statistical tests outputs a P-value; the P-value summarizes the strength of the evidence against the null hypothesis, which corresponds to the fact that the sequence being tested is random. P-value denotes the probability that a perfect random number generator would have produced a sequence less random than the input sequence that is tested. For a P-value ≥ 0.01, the sequence is considered as random with a confidence of 99%. Note that all the P-values shown in Table 2.1, Table 2.2, and Table 2.3 are at least 0.01, which demonstrates that the secret bit streams are in fact random with a very high degree of confidence.

2.8 Multiple Bit Extraction

In this section, we evaluate the performance of extracting multiple bits from a single RSS sample. The goal here is to find whether or not the extraction of multiple bits from a single RSS sample increases the secret bit rate in comparison to single bit extraction.

In Section 2.6, we have shown that the measurements from static settings exhibit a very narrow RSS range (for example, only 2 dB variation in the experiment of Section 2.6.1). Extracting even 2 bits from an RSS sample requires a range of at least 4 dB when RSS is reported in 1 dB steps. Further, in Section 2.7, we have shown that the mismatch rate in the static settings is as high as 50%. Attempting to extract multiple bits will cause the mismatch rate to increase further. Therefore, we apply our multiple bit extraction method only to mobile settings that do not suffer from these problems of narrow range and very high mismatch rates.

Recall from Section 2.4 that N is the number of bits extracted per RSS measurement, and M $(= 2^N)$ is the number of equi-sized intervals into which the RSS range is divided. Figure 2.18 shows the mismatch rates for extracting $N = 2 - 4$ bits, respectively, from each RSS measurement. Observe that the mismatch fraction increases with N, the number of bits extracted per measurement. Further, the way in which the N bits are assigned to each of the M intervals also affects the mismatch fraction. For example, the use of Gray codes results in a substantially lower mismatch fraction compared to the use of a regular binary sequence, as shown in Figure 2.18. Due to nonperfect channel reciprocity, if an RSS measurement of Alice and that of Bob belong to adjacent intervals, use of Gray codes ensures that the N bits extracted by Alice and Bob differ by at most one bit, whereas using a regular binary sequence causes the bits extracted by Alice and Bob to potentially differ in all the N bits. This accounts for a lower mismatch rate and subsequently higher secret bit rate when using a Gray code sequence.

Figure 2.19 shows a comparison of secret bit rates for our single and multiple bit extraction methods under various mobile settings. Notice that for the mobile settings, the secret bit rate for single bit extraction is about 16%, whereas for 2 bits extraction ($N = 2$) using Gray coding, the secret bit rate is about 67%. Notably, the secret bit rate of the multiple bit extraction method is at least four times higher than that of the single bit extraction method even when only 2 bits are extracted from each measurement. This substantial improvement accounts for the fact that the single bit extraction method drops all the RSS measurements that lie within the upper and lower thresholds, while the multiple bit extraction method utilizes most

of the measurements. Furthermore, similar to our single bit extraction method, the extracted bit streams have an entropy value close to 1 due to privacy amplification. To summarize, the multiple bit quantization scheme substantially improves the secret bit rate in environments with mobile devices.

2.9 Secret Key Extraction Using Handheld Devices

Given the widespread prevalence of inexpensive and low-power mobile devices, in this section, we evaluate our secret key extraction using two mobile devices, Google Nexus One smartphones, that are equipped with Broadcom BCM 4329 chipset-based 802.11 wireless network cards. We first perform experiments similar to the ones described in the previous section in two different environments. Although not shown here, we obtain high entropy secret bits fairly quickly when using these smartphones and our secret bit streams also pass the NISTs approximate entropy test, achieving an entropy value close to the ideal value of one. In the rest of this section, we examine the impact of distances between two smartphones, Alice and Bob, on secret key extraction in two different environments while they transmit at a very low power.

2.9.1 Experimental Setup

We conduct a number of experiments in the University of Utah campus under two different environments that are changing with time. In each environment, we perform four *walk-experiments* where the phones representing Alice and Bob are carried at normal walking speeds. The average distance (d) in feet between Alice and Bob is varied with each experiment and $d \in \{25, 50, 75, 100\}$.

This first environment is a hallway on the third floor of the Merrill Engineering Building. In the experiments conducted in this environment, our phones use the lowest transmit power of 4 dBm.

We conduct a second set of experiments in an outdoor environment across varying terrain, with many trees and bushes in the path between Alice and Bob. Because of the terrain and obstructions in this environment, the path losses are higher. Due to greater path loss in this environment, we use a higher transmit power of 8 dBm.

2.9.2 Results

In this subsection, we evaluate secret key extraction as a function of distance between Alice and Bob[4]. Our results show that in the hallway environment, even with the lowest transmit power, Alice and Bob can extract about 0.25 secret bits per probe when they are separated by about 25 feet. Figure 2.20 shows a plot of secret bits per probe as a function of the distance between Alice and Bob. Though we use a lower transmit power in the hallway-environment, in comparison to the trees-environment, the hallway-environment achieves a higher performance due to lower signal attenuation – from our measurements, we find that for a given distance, the average received powers are about $2 - 7$ dB higher in the hallway environment in comparison to the obstructed outdoor environment. As we show in Figure 2.20, secret bits per probe decreases with increase in distance, which is attributed to the following reason: As the distance increases, the signal-to-noise ratio (SNR) decreases, which consequently increases both the bit mismatch rate (Table 2.4) and the packet drop probabilities (Table 2.5); the increase in packet drop further contributes to an increase in the time duration between channel measurements. Nevertheless, on the whole, a comparison of our results in Figure 2.20 and Figure 2.17 shows that secret keys can be established efficiently even with low-powered, mobile devices.

2.10 Related Work

This paper advances the research area [35, 36, 37, 20, 38, 39, 29, 40, 41] of generation of shared secret keys from the observation and processing of radio channel parameters.

Amplitude or channel gain is the most common reciprocal channel feature used for secret generation in the literature [19, 21, 30, 22, 16, 20]. Amplitude can be measured more easily than time delay or phase on most existing hardware, and thus is more readily applicable to common wireless networks. In this paper, we similarly use measurements of amplitude, based on their universal availability in wireless networks.

In [41], several bidirectional UWB measurements are made and used to compute the number of secret bits which could be generated. In [21], an implementation using

[4]We borrowed the code for RSS extraction for Android smartphones from Jessica Croft [34].

the universal software radio peripheral (USRP) and GNU software radio generates and receives the required multicarrier signal and evaluates the secret bit rate of the system. In [22], researchers use a steerable directional antenna in combination with Zigbee radio hardware to generate a secret between two nodes and test what an eavesdropper would have received. In [20], Mathur et al. implement two different systems, one using channel impulse response and another using amplitude measurements, to generate secret keys and test how an eavesdropper's measurements differ from the original measurements. Our work differs from Mathur's in the following significant ways. First, we perform extensive real-world measurements in a variety of environments and settings to determine the effectiveness of RSS-based secret key extraction. Second, we propose an adaptive secret key extraction scheme that instead of dropping mismatched bits, uses information reconciliation to reduce the mismatched bits and also uses privacy amplification. Third, we expose the problem of a predictable channel attack. Last, we further increase the secret bit rate by extracting multiple bits from each RSS measurement.

Bloch et al. [28] and Ye et al. [42] present an alternative multiple bit extraction scheme that is strongly tied to their use of a low-density parity-check (LDPC)-based error correction mechanism, which allows them to exploit the correlation between the bits of each sample for error correction. Our work differs from Bloch et al. [28] and Ye et al. [42] in the following ways. First, Bloch et al. conclude that the memory requirements and the complexity of such LDPC-based schemes may be too high, especially for low-cost systems, while the Cascade [17]-based information reconciliation mechanism in our ASBG scheme has very low memory requirements and is much less complex than the LDPC-based schemes. Second, these LDPC-based schemes rely on *redundant/over-quantized* bits for error correction; they extract M bits from each sample, where M is *at least* $\log_2 K$, and K denotes the number of unique, discrete-valued measurements; in our multiple bit quantization, on the other hand, we extract *at most* $\lfloor \log_2 K \rfloor$ bits from each sample. Hence, in our scheme, we do not extract more bits per sample than what is indicated by the upper bound on the actual information content / entropy present in the measurements, which equals $\log_2 K$. Third, it is possible to calculate the fraction of information that is

leaked with Cascade for a given bit mismatch rate, and our privacy amplification stage appropriately reduces the output secret key size depending on this fraction of information leakage.

2.11 Conclusion

We evaluated the effectiveness of secret key extraction from the received signal strength (RSS) variations in wireless channels using extensive real-world measurements in a variety of environments and settings. Our experimental results showed that bits extracted in static environments are unsuitable for generating a secret key. We also found that an adversary can cause predictable key generation in static environments. However, bits extracted in dynamic environments showed a much higher secret bit rate. We developed an environment adaptive secret key generation scheme and our measurements showed that our scheme performed the best in terms of generating high entropy bits at a high bit rate in comparison to the existing ones that we evaluated. The secret key bit streams generated by our scheme also passed the randomness tests of the NIST test suite that we conducted. We were able to further enhance the rate of secret bit generation of our scheme by extracting multiple bits from each RSS measurement. We have presented these results in two major papers [14, 43]. The conclusions drawn in this work, specifically the predictable channel attack, are primarily for secret key extraction using RSS measurements, and these may not directly apply to key extraction using channel impulse response measurements. We would like to explore this in our future work. In this chapter, we have described our work on secret key extraction between *a pair* of wireless nodes; in the next chapter, we explore the use of *two groups* of wireless sensor nodes for extracting stronger keys in an efficient manner.

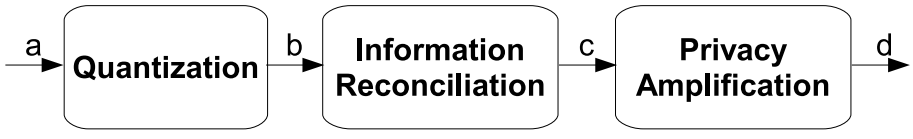

Figure 2.1. Secret key extraction process. a - RSS measurements, b - quantized bits, c - reconciled bits, d - secret bits.

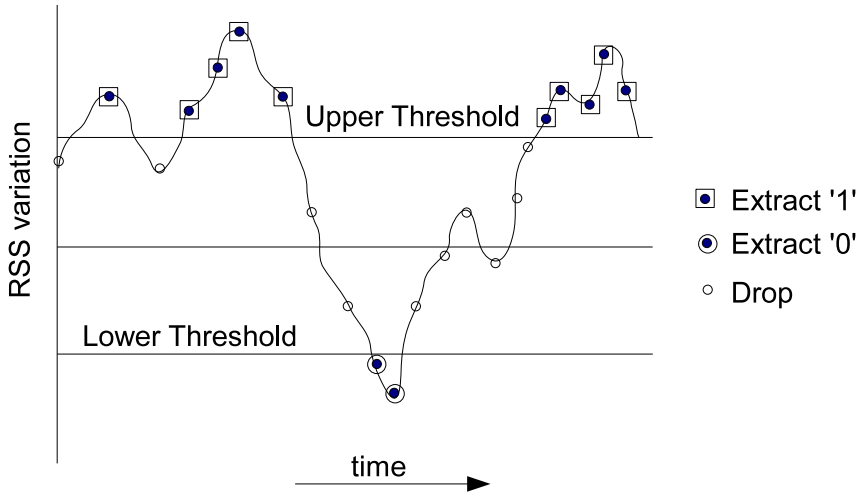

Figure 2.2. A sample quantizer. Measurements above the upper threshold encoded as bit "1"; those below the lower threshold encoded as bit "0"; others are discarded. For this set of measurements, the quantizer outputs 111110011...

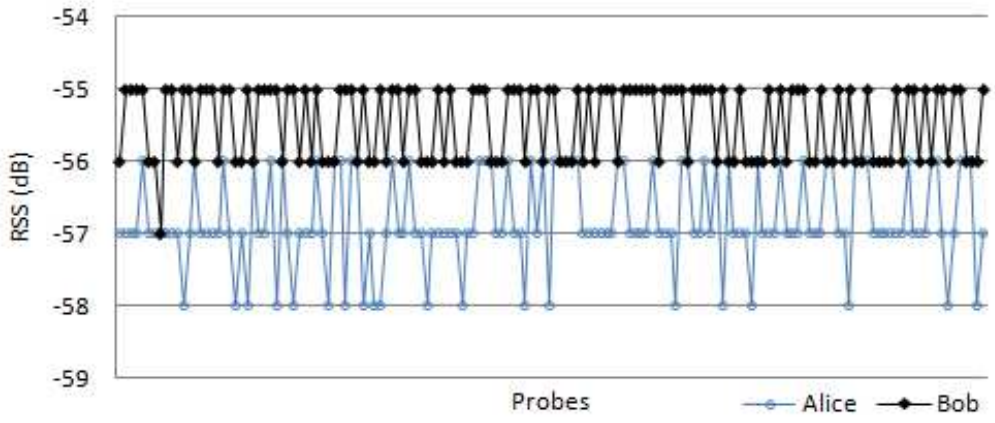

Figure 2.3. Underground concrete tunnel measurements. Note that Alice and Bob exchange about 20 probe packets per second for collecting the measurements.

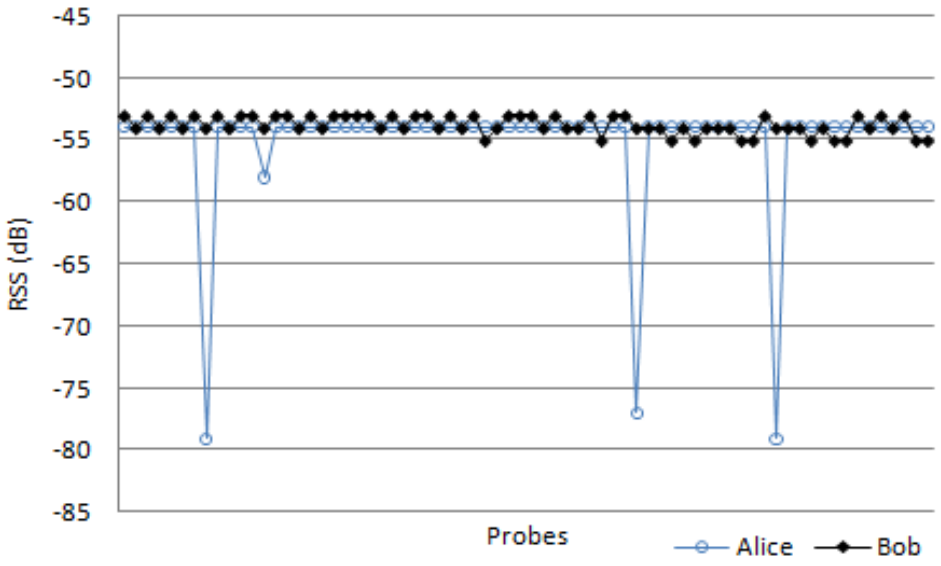

Figure 2.4. Engineering building gallery measurements

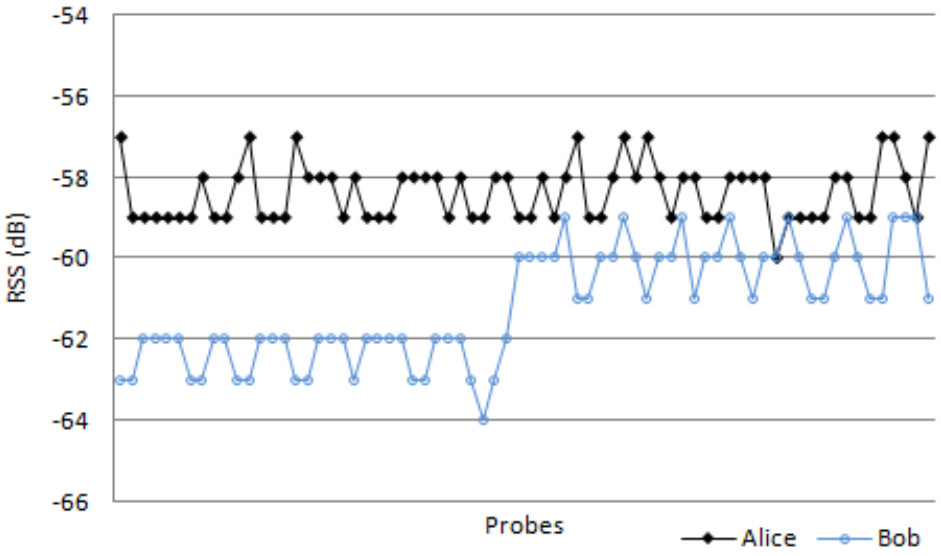

Figure 2.5. Measurements in the lawn between the cafeteria and library

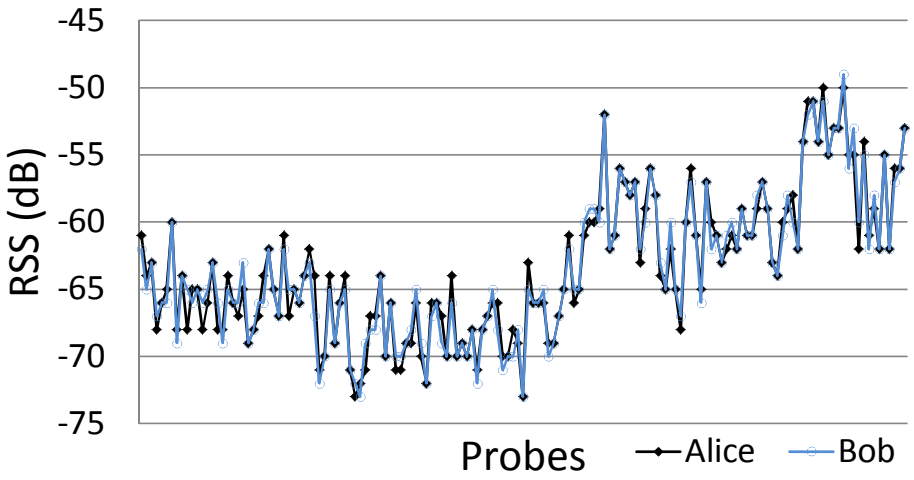

Figure 2.6. Measurements while walking inside an engineering building

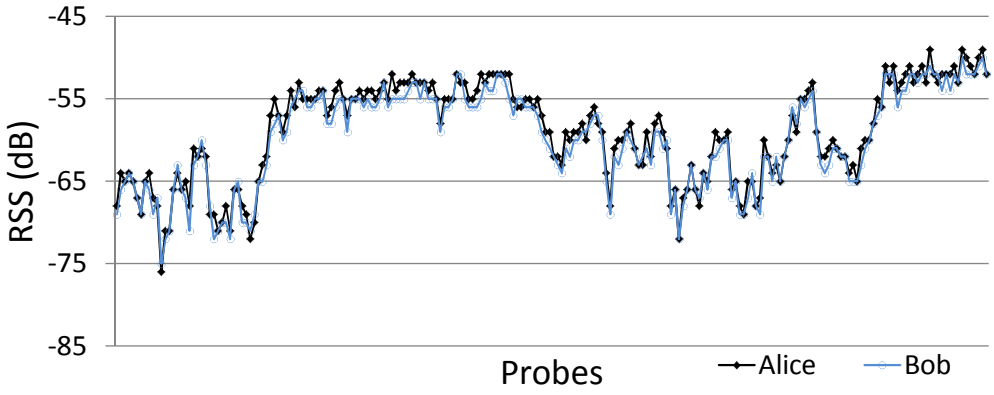

Figure 2.7. Measurements while walking from an engineering building to the cafeteria

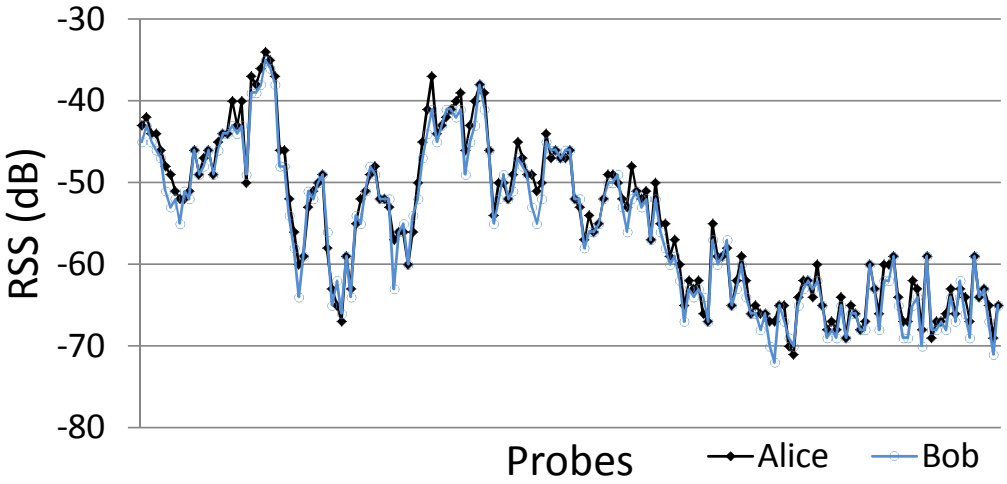

Figure 2.8. Measurements from slow bike ride on city streets

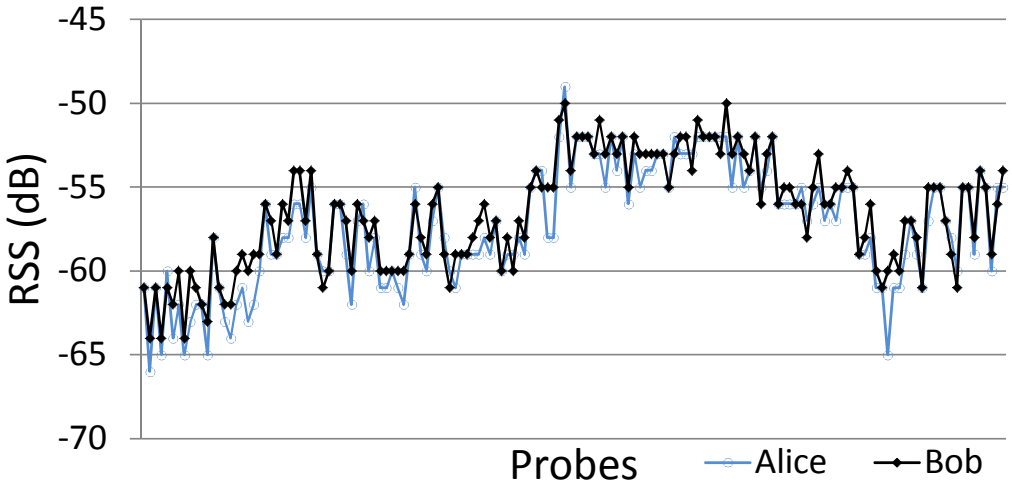

Figure 2.9. Crowded cafeteria measurements

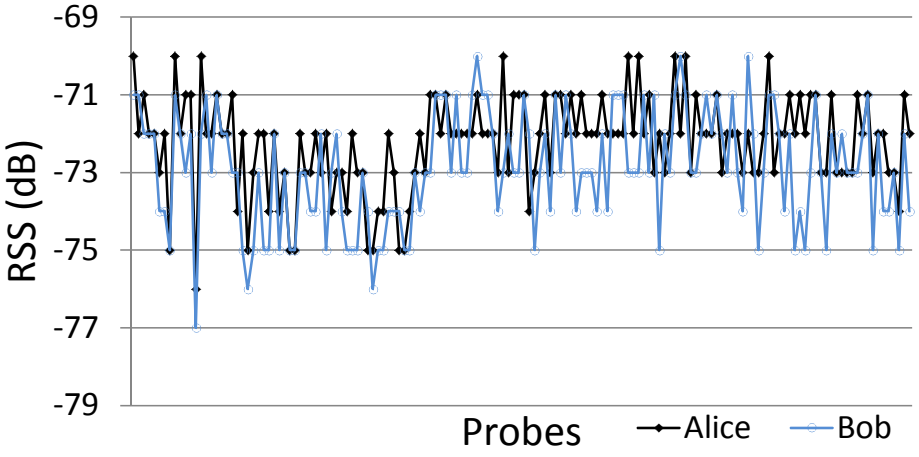

Figure 2.10. Measurements across a busy road

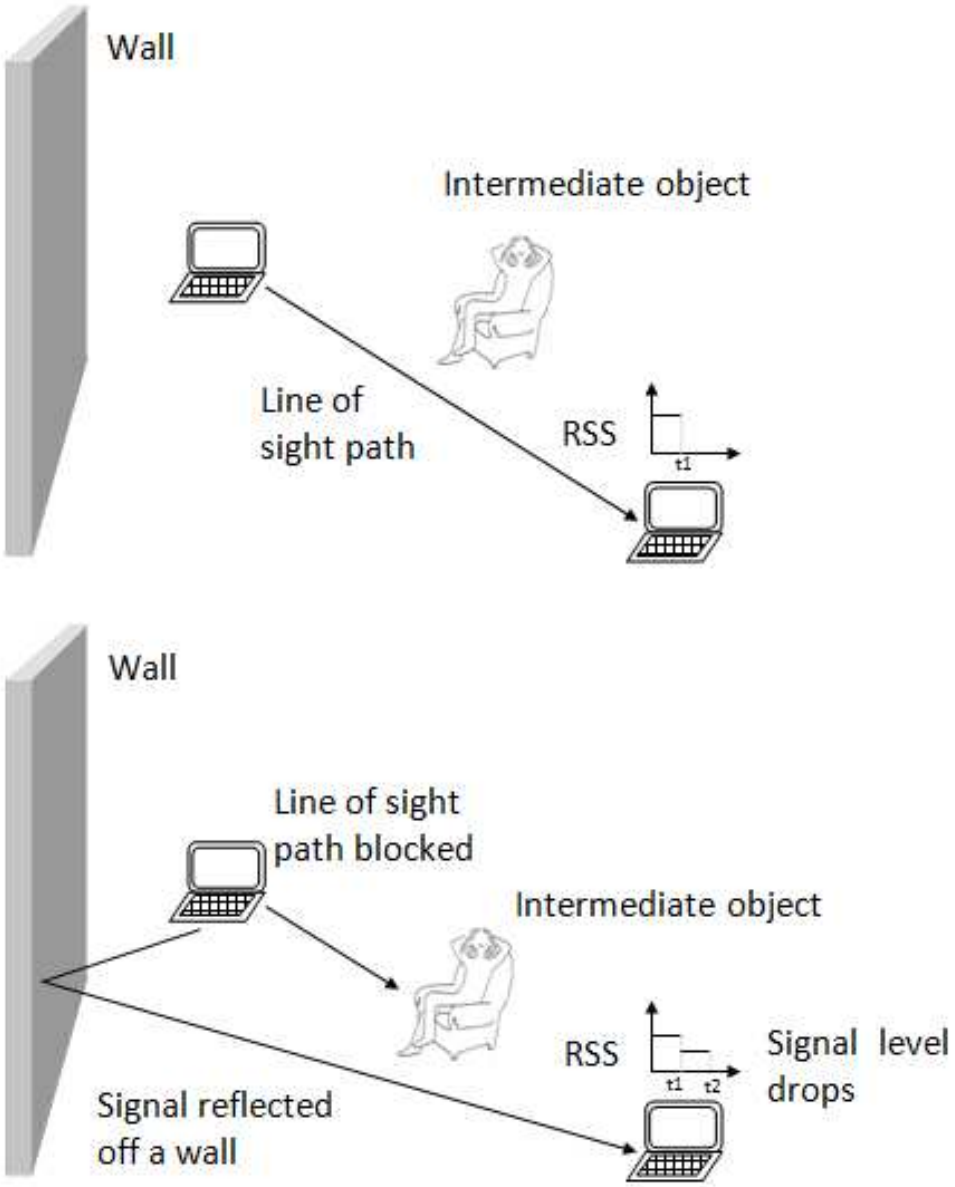

Figure 2.11. Schematic of the attack. In the top portion of this figure, there is a line of sight path. In the bottom portion, the attacker intermittently blocks the line of sight path causing a predictable drop in the RSS values.

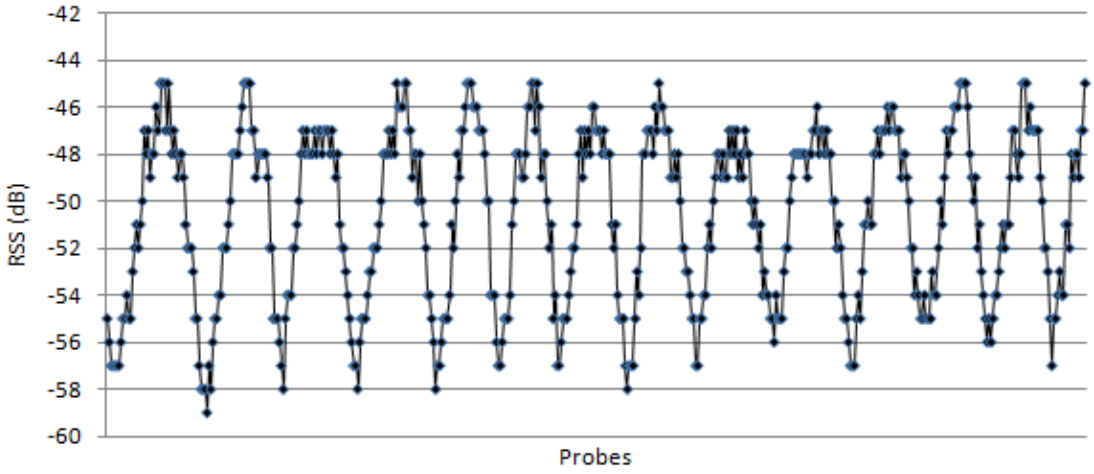

Figure 2.12. Predictable variations of the RSS values when an adversary repeatedly blocks and unblocks the line of sight path using an intermediate object.

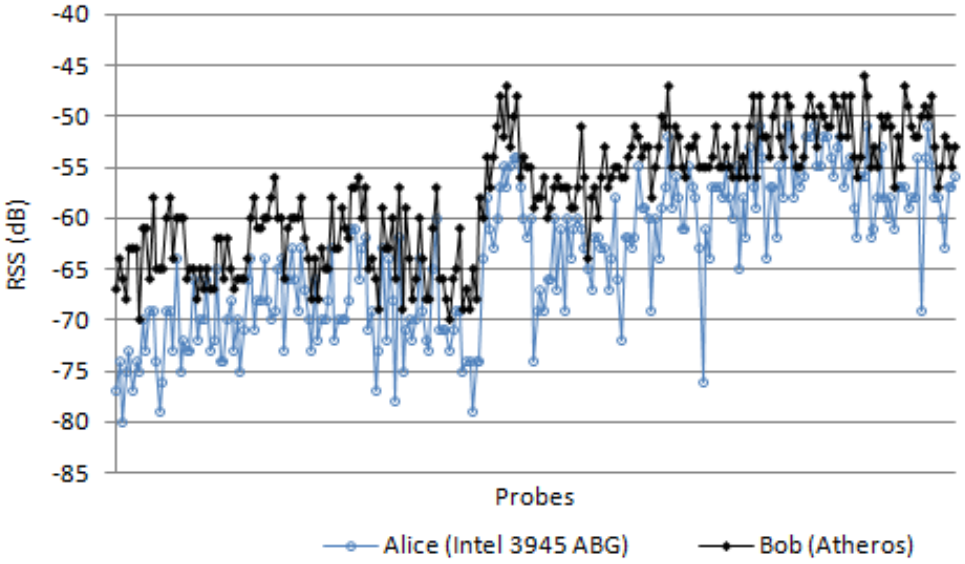

Figure 2.13. Measurements from heterogeneous devices while walking inside an engineering building

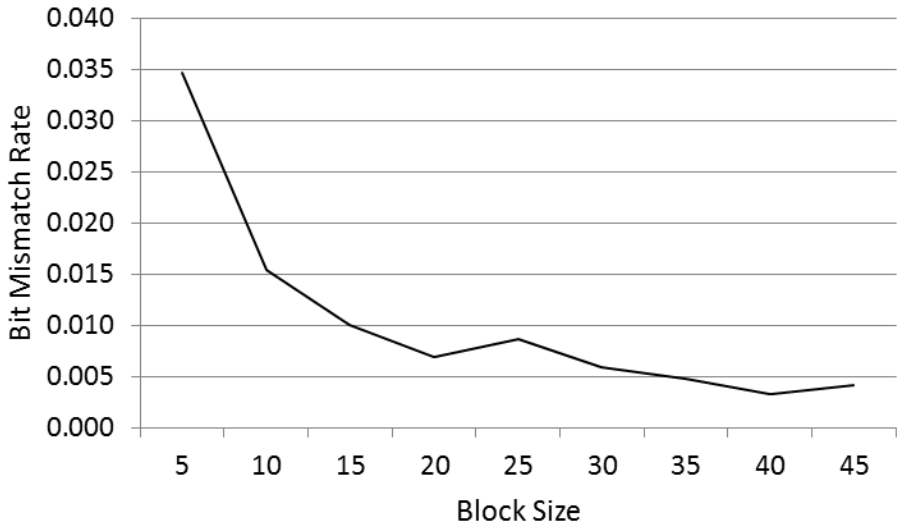

Figure 2.14. Variation of bit mismatch rate against block size for ASBG method.

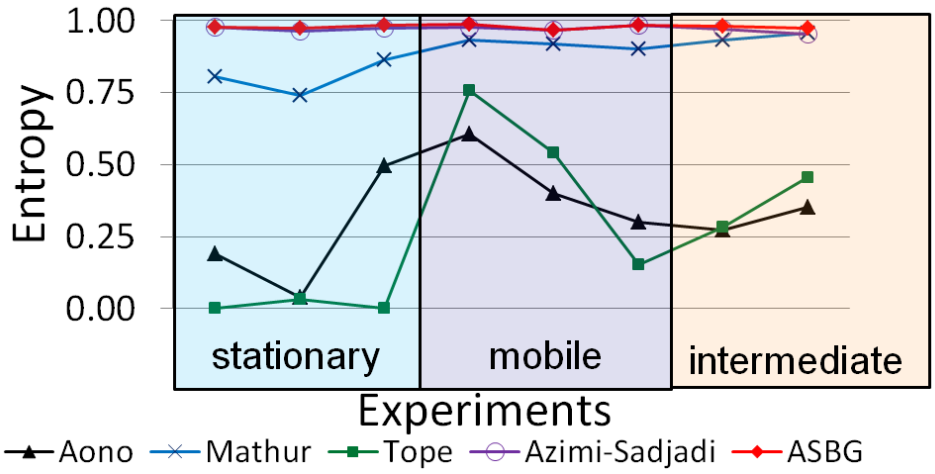

Figure 2.15. Entropy comparison between existing quantization schemes and ASBG under various settings.

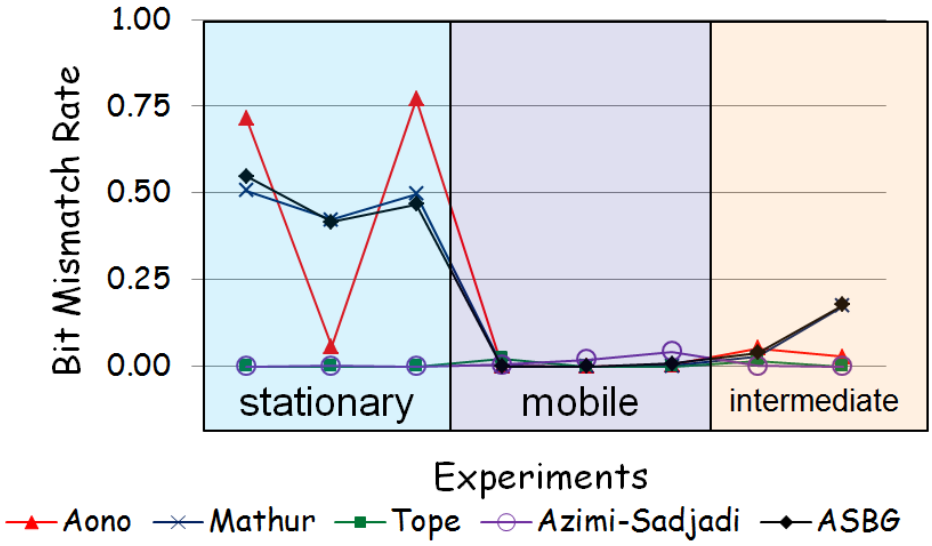

Figure 2.16. Bit mismatch rate comparison

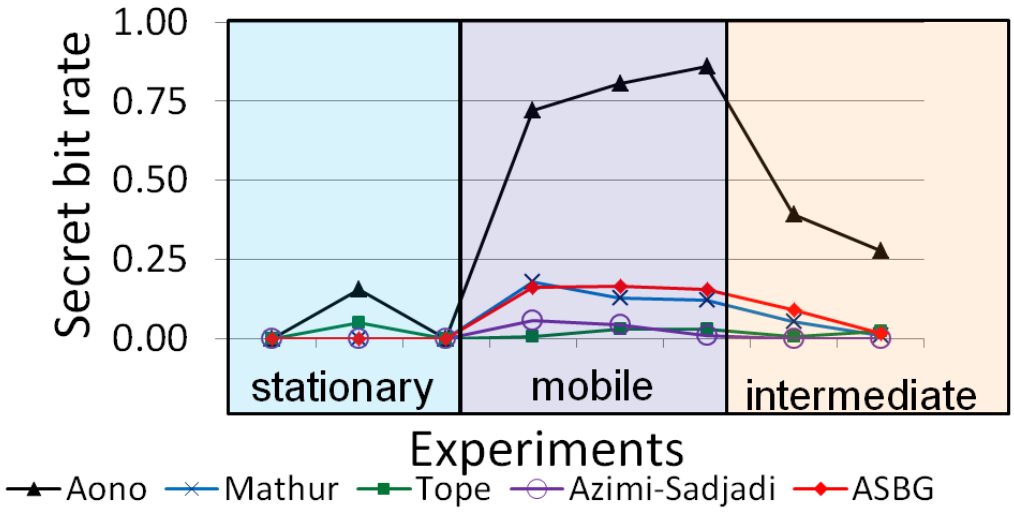

Figure 2.17. Secret bit rate comparison

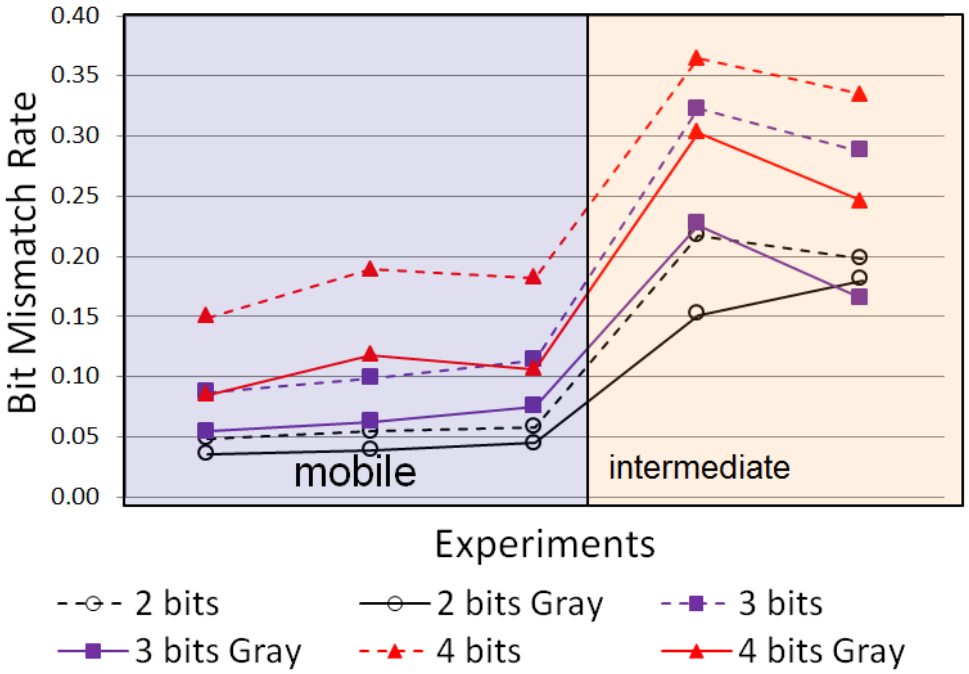

Figure 2.18. Bit mismatch rate comparison

Table 2.1. P-values from NIST statistical test suite results. Experiments $\{A, B, C\} \in$ stationary category.

Test	A	B	C
Frequency	0.35	0.03	0.51
Block Frequency	0.52	0.57	0.82
Cumulative sums(Fwd)	0.46	0.05	0.78
Cumulative sums (Rev)	0.27	0.03	0.46
Runs	0.21	0.54	0.74
longest run of ones	0.08	0.1	0.49
FFT	0.71	0.74	0.28
Approx. Entropy	0.06	0.34	0.56
Serial	0.84, 0.50	0.40, 0.23	0.84, 0.64

Table 2.2. P-values from NIST statistical test suite results. Experiments $\{D, E, F\} \in$ mobile category.

Test	D	E	F
Frequency	0.14	0.51	0.37
Block Frequency	0.66	0.38	0.94
Cumulative sums(Fwd)	0.19	0.34	0.68
Cumulative sums (Rev)	0.09	0.89	0.39
Runs	0.41	0.74	0.38
longest run of ones	0.65	0.76	0.40
FFT	0.59	0.51	0.52
Approx. Entropy	0.67	0.65	0.21
Serial	0.50, 0.59	0.50, 0.64	0.43, 0.59

Table 2.3. P-values from NIST statistical test suite results. Experiments $\{G, H\} \in$ intermediate category.

Test	G	H
Frequency	0.98	0.95
Block Frequency	0.63	0.03
Cumulative sums(Fwd)	0.55	0.18
Cumulative sums (Rev)	0.52	0.21
Runs	0.55	0.07
longest run of ones	0.78	0.96
FFT	0.23	0.65
Approx. Entropy	0.55	0.25
Serial	0.60, 0.36	0.16, 0.50

Table 2.4. Bit mismatch rate as a function of distance.

Distance (feet)	**Bit Mismatch Rate (Hallway)**	**Bit Mismatch Rate (Trees)**
25	0.29%	2.46%
50	1.67%	3.60%
75	1.81%	3.63%
100	1.91%	5.29%

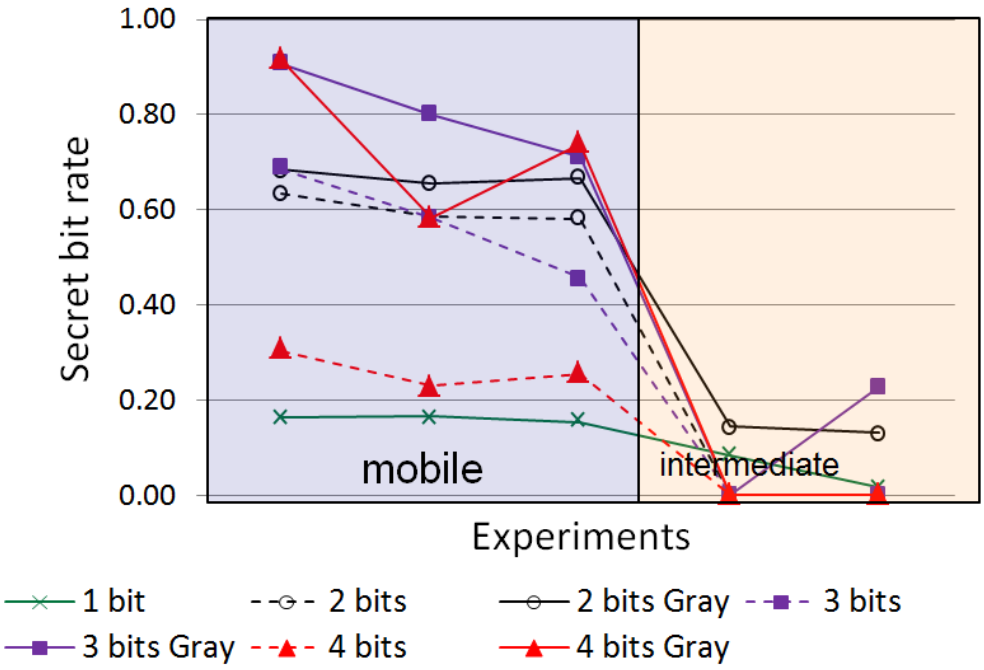

Figure 2.19. Secret bit rate comparison when extracting different number of bits under various settings.

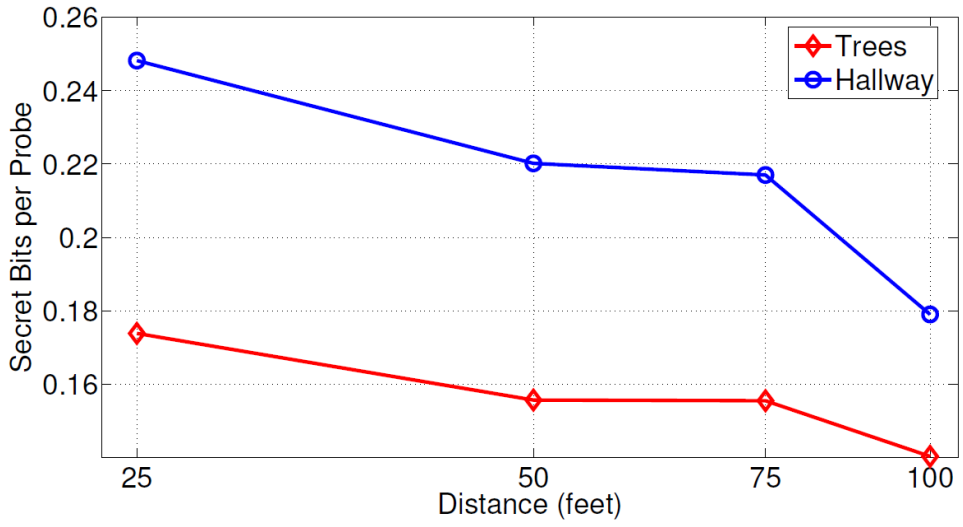

Figure 2.20. Secret bits per probe as a function of distance.

Table 2.5. Packet loss rate as a function of distance.

Distance (feet)	**Packet Loss Rate (Hallway)**	**Packet Loss Rate (Trees)**
25	1%	1%
50	4%	3%
75	4%	18%
100	9%	27%

CHAPTER 3

EFFICIENT HIGH RATE SECRET KEY EXTRACTION IN SENSOR NETWORKS USING COLLABORATION

3.1 Overview

Secret key establishment between a pair of nodes, each of which has a single-input and single-output (SISO) radio, is evaluated extensively in the previous chapter. In this chapter, we investigate secret key extraction under a multiple-input multiple-output (MIMO)-like setup using two groups of wireless sensor nodes. Recently, the use of MIMO has been proposed for enhancing secret key extraction. Wallace et al. [44] present an analytical study and simulation results on the use of multiple antennas for secret key extraction, but they assume that multiple antennas belong to the same node. However, due to size and power limitations, sensor nodes do not typically have multiple antennas. In this part of our work, we propose to obtain the multi-antenna capability through collaboration among sensor nodes such that we leverage the variations in the wireless channels between all possible pairs of nodes among two groups of sensors to extract secret keys at a very high rate and also in an energy efficient manner. Furthermore, unlike MIMO-based schemes, our research does not make any assumptions of phase synchronization of RF signals.

In our new approach, multiple sensors collaborate in exchanging probe packets and collecting channel measurements. Specifically, there are two groups of sensors, with one group representing Alice and another Bob. When one of Alice's nodes transmits a probe packet, all the nodes (say N) belonging to Bob simultaneously receive the packet and measure the received signal strength (RSS) values. Alice's nodes take

turn to transmit their probes. This is followed by probes from Bob's nodes (also N). The simultaneous collection of measurements across different nodes allows us to extract secret bits at a much faster rate and also considerably reduce the energy consumption. Essentially, measurements from multiple channels have a substantially higher differential entropy compared to the measurements from a single channel, thereby resulting in more randomness in the information source for key extraction, and this in turn produces *stronger* secret keys.

We note that the increase in the number of nodes per group, with sensor nodes taking turns to transmit, can reduce the sampling rate of each wireless channel. Furthermore, this can cause a significant time gap between bidirectional measurements between nodes across the two groups. This reduced sampling rate and increased time gap results in a high bit mismatch rate between the nodes performing bidirectional measurements because the rate at which the measurements are taken is lower than the rate of change of the channel. There is a fundamental trade-off between the quadratic increase in the number of measurements of the channels due to multiple nodes per group versus a linear reduction in sampling rate and a linear increase in the time gap. The exploration of this trade-off is an important contribution of this chapter.

To experimentally evaluate collaborative secret key extraction in wireless sensor networks, we first build a simple, yet flexible testbed with multiple TelosB sensor nodes. We employ an interpolation technique [45] to substantially reduce the bit mismatch rate. We perform large-scale experiments with different configurations of collaboration with N ranging from 1 to 5. Specifically, we perform *slow-walk* experiments, and *iRobot* experiments. In the slow-walk experiments, all Bob nodes, fixed on a cardboard, are carried around at a normal walking speed. In the iRobot experiments, Bob nodes are fixed onto the rim of an iRobot rotation platform that is rotated at roughly a constant rate. Alice remains stationary in all experiments. As expected, our experimental results show that, for both type of experiments, there is a substantial improvement in the differential entropy estimate of the information source, almost quadratic in the number of nodes. Our experiments also show that for the iRobot experiments, in comparison to the 1×1 configuration, collaboration (i) increases the rate of extraction of secret key bits per probe up to 332%, and (ii)

increases the rate of extraction of secret key bits per mJ of transmission energy up to 319%, with the maximum achieved when $N = 4$ for both the measures. In the slow-walk experiment, the increase in the corresponding secret bit rates is much lower, but the maximum is achieved at $N = 2$.

We propose and implement a hierarchical collaborative approach, that in addition to using space diversity, also exploits frequency diversity to extract stronger secret keys at an even faster rate. Using measurements from real-world experiments, we show that the hierarchical approach substantially improves the performance and is best suited for key extraction.

The rest of this chapter is organized as follows. In Section 3.2, we describe our simple collaborative secret key extraction approach. Section 3.3 presents our hierarchical collaborative approach. We present the benefits of distributing the different stages of secret key extraction among different nodes in Section 3.4. Section 3.5 shows how a group of nodes can generate group keys. Our experimental setup is discussed in Section 3.6. We present the results for our simple collaboration and hierarchical collaboration approaches respectively in Section 3.7 and Section 3.8. We summarize our main results in Section 3.9. We discuss the related work in Section 3.10 and present our conclusions in Section 3.11.

3.2 Simple Collaboration

In this section, we describe our efficient collaborative secret key extraction approach. There are two groups of sensor nodes, with each group associated to an aggregator / access point. Assume that there are N nodes in each group. There are several ways in which the access points, or the group of nodes, can establish a secret key among them. In the most naive approach, the access points exchange a set of probe packets, collect RSS measurements, and use the secret key extraction process discussed in Section 2.3 to extract the secret key bits. In this approach, in order for a node to record N^2 measurements, the other node needs to transmit N^2 probe packets.

In simple collaboration, a set of N nodes under each access point participates in the exchange of probe packets and collects RSS measurements. Whenever some node $S_{ai}, i \in \{1, \cdots N\}$, belonging to Alice, transmits a probe packet, all the N nodes

$S_{bj}, \forall j \in \{1, \cdots N\}$, belonging to Bob, receive the probe packet, and record an RSS measurement. Thus, with N nodes at each end, and for each transmission from each $S_{ai}, i \in \{1, \cdots N\}$, i.e., for a collective transmission of N probe packets from Alice, Bob can collectively record N^2 measurements. An example of this approach with $N = 3$ nodes is shown in Figure 3.1.

Assume that there is a secure channel already in place between an access point and its group of associated nodes. We recognize the following two alternatives for establishing a secure channel between an access point and its sensor nodes.

First, as described in our previous work [14], two nodes, a sensor node and its access point, can establish an information-theoretically strong secret key using variations in the wireless channel characteristics. Alternatively, we can assume that the AP and its associated sensor nodes are physically very close. Thus, they are able to communicate via other physical layers than the longer-range physical layer used to communicate between APs, for example, via infrared. Because of the short distance, we believe it is reasonable to assume a secure pre-existing channel between AP and its sensors, even though a rigorous demonstration that this is indeed possible is beyond the scope of the present chapter.

Using a secure channel, established through one of the above methods, each sensor node can share its set of measurements with its corresponding access point, in the following three different ways.

- In the first approach, the nodes piggyback the payload of probe packets with measurements from the previous round. These piggybacked measurements are encrypted with a key shared between the node and its access point. The receiving nodes in the other group, their access point, or an eavesdropper can only record the RSS value for the received probe packet, but cannot decrypt the payload. When the associated access point, however, receives the same probe packet, it decrypts the payload to obtain the RSS measurements from the previous round, and stores them for later use.

- In the second approach, each node locally caches all the measurements until the RSS measurements collection phase is over. Then, it encrypts all the

measurements and sends a single packet containing all the measurements to the access point.

- In the third approach, instead of sending all the measurements to the access point, nodes share the quantized bits over the secure channel. We further discuss this approach in Section 3.4.

The simple collaborative method for exchanging probe packets and recording RSS measurements is shown in Algorithm 1. When the nodes in each group share the measurements/quantized bits with their corresponding access point, the access points use the key extraction process that we have outlined in Section 2.3 to establish a secret key between them.

There are two principal advantages with the collaboration approach in comparison to the naive approach. First, since it exploits channel variations across N^2 different bidirectional channels, there is more diversity / variety in the set of measurements Alice (or Bob) collects. We show in Section 3.7 that the differential entropy of a set of measurements that is collected using multiple nodes rises quadratically in the number of nodes in each group. So, a set of measurements from multiple channels is a better source of information for key extraction, and hence, it is likely to produce stronger keys in comparison to the naive approach. Second, since there is a reduction in the number of transmissions by a factor of N, we can expect to extract bits at a faster rate, and in an energy efficient manner. Through 10 different experiments, we show in Section 3.7 that we can achieve substantial improvements in terms of the number of secret bits extracted - per second, - per probe transmission, and - per mJ of transmission energy.

The quadratic increase in the number of channel measurements, however, comes with a linear reduction in the sampling rate and linear increase in the time gap between bidirectional measurements between nodes across the two groups (Section 3.7.3). The increase in time gap reduces the mutual information between a pair of measurements in each direction, resulting in high rate of bit mismatches. So, the number of useful secret bits extracted in the process becomes limited when N becomes sufficiently high (Section 3.7.6).

3.3 Hierarchical Collaboration

While there are several inherent advantages with simple collaboration, we will see in Section 3.7 that this method achieves peak performance for relatively small values of N, and hence is not very scalable. Further, the simple collaboration approach exploits only space diversity - i.e., different nodes are spread over an area operating in the same frequency channel. Typically, the sensor nodes are designed to operate over a number of channels with different carrier frequencies. For example, in the 802.15.4 standard, there are 16 channels in the 2.4 GHz band, each 2 MHz wide, and with a spacing of 5 MHz between the center frequencies of adjacent channels. Such nonoverlapping channels allow for nodes operating in different channels to transmit simultaneously without causing any collisions. Therefore, if the sensor nodes can exploit both space diversity and frequency diversity simultaneously, it is possible to extract secret keys in an even more efficient manner.

In hierarchical collaboration, the N nodes in each group are further subdivided into smaller subgroups, with equal number of nodes, N_s in each subgroup. Each one of these $\frac{N}{N_s}$ subgroups is assigned to operate over a different frequency channel for the duration of the key extraction process. The nodes in each pair of subgroups assigned to the same frequency channel collaborate (using simple collaboration) in collecting the RSS measurements, where one of the subgroups in the pair belongs to Alice and the other subgroup in the pair belongs to Bob. The hierarchical collaboration method is shown in Algorithm 2. Since different subgroup pairs are assigned to different frequency channels, it enables all such pairs to operate in parallel. As a result, we can expect to extract keys at a faster rate. Figure 3.2 shows an example with $N = 4$ nodes, with the first 2 nodes in each group assigned to frequency channel 1, while the remaining 2 nodes in each group are assigned to a different frequency channel 2.

While a static frequency assignment to the different subgroups, as we have described above, allows for faster key extraction, this approach exploits the variations of only $N_s^2 \times \frac{N}{N_s} = N_s \times N$ different channels. Recall that the simple collaboration approach, however, exploits variations in all the N^2 channels. So, in order to obtain measurements from all the N^2 channels in the hierarchical approach as well, we propose that different subgroups of either Bob or Alice periodically switch

between the different available frequency channels, for example, after every n_p packet transmissions. The frequency switch algorithm is shown in Algorithm 3.

As a consequence of hierarchical subgrouping, the $\frac{N}{N_s}$ subgroups and the corresponding access point operate at different frequencies. As a result, nodes cannot share the measurements with the corresponding access points through the piggybacking approach that we have described earlier. To overcome this, the nodes have to cache the measurements and either share them with the access point after the measurements collection phase, or quantize the collected measurements and exchange only the quantized bits, which is addressed next in Section 3.4.

3.4 Distributed Key Extraction Stages

In the simple and hierarchical collaborative key extraction approaches we have described thus far, all the 4 stages, namely, interpolation, quantization, information reconciliation and privacy amplification, are assumed to be carried out at the access point, and the sensor nodes in each group / subgroup take part only in collecting the measurements. Alternatively, it is possible to distribute the various stages among different nodes and extract keys in an efficient manner. In the distributed key extraction approach, interpolation and quantization are carried out at each node, while information reconciliation and privacy amplification are carried out at the access point. Doing so allows each node to send only the quantized bits, and avoid exchanging a lot of measurements with the access point. Therefore, the distributed approach speeds up key extraction in the following ways - (i) by significantly reducing the amount of information that needs to be exchanged between the nodes and the access point, and (ii) by distributing the computational tasks (of interpolation and quantization) across different nodes.

Let K denote the expectation of the number of quantized bits contributed by each node to the key extraction process. Assuming that the measurements follow a Gaussian distribution, it can be easily shown that to obtain K quantized bits, each node needs to record $M = \frac{K}{2 \times Q(\alpha)}$ measurements, where $Q(y)$ denotes the complementary cumulative distribution function of a Gaussian random variable Y, with zero mean and unit variance. In the *undistributed approach*, to contribute K

quantized bits to the key extraction process, each node needs to send M measurements to the access point, where each measurement represents 1 byte of information. In the *distributed key extraction approach*, each node needs to send just K bits to the access point. For example, to contribute $K = 32$ quantized bits, and when $\alpha = 0.4$, for instance, the amount of information exchanged between a node and its access point is just 32 bits, or 4 bytes for the distributed case, and ≈ 46 bytes for the undistributed case.

3.5 Group Key Generation

Nitinawarat et al. [46] consider secret key generation for a pairwise independent network (PIN) model. The objective is to generate a secret key shared by a given subset A of terminals in $M = \{1, \cdots m\}$ using the cooperation of the other remaining terminals. The PIN model is motivated by the practical aspects of wireless communication - that the reciprocal wireless channel between two terminals decorrelate with time (of the order of coherence interval) and distance (of the order of a few wavelengths). In an earlier work, [47] show that the largest rate at which terminals in A can generate secrecy, with the help of remaining terminals, is obtained by subtracting from the total joint entropy the smallest rate of communication which enables each terminal in A to reconstruct all the m components of the multiple source. Nitinawarat et al. [46] express the secret key capacity for the PIN model in terms of a linear combination of mutual information terms that involve only mutually independent pairs of reciprocal random variables.

Based on the maximal packing of Steiner trees in a multigraph, Nitinawarat et al. [46] propose an algorithm for propagating pairwise secret keys for a pair of terminals in M to form a group-wide secret key for the terminals in A. Specifically, for edges (i, j) and (i, j'), vertex i in the Steiner tree broadcasts to vertices j, j', the binary sum of two independent secret key bits - one with j and the other with j'. This enables i, j, j' to share any of these two bits with the attribute that the shared bit is independent of the binary sum; this propagation also enables all the vertices in A, which are connected in the Steiner tree, to share one bit among them.

In the following, we argue on how we can achieve the secret key capacity bounds

of Nitinawarat et al. when one group of nodes collaborates in sharing a group secret key.

(i) It has been shown using real-world measurements by Mathur et al. [20] and Zeng et al. [24] that Alice and Bob share enough mutual information to extract one or two bits from each measurement in the quantization stage. Consistent with these observations from earlier work, we also extract at most one bit from each measurement depending upon whether the measurement lies within or beyond the upper and lower thresholds.

(ii) We have shown that measurements from different channels have very low correlation (Section 3.6.3). Under Gaussian assumption, lack of correlation between different components of a random vector also implies mutual independence between those components. Therefore, our RSS measurements from different channels can be considered to be mutually independent.

(iii) For group key extraction problem, Nitinawarat et al. expressed secret key capacity as a linear combination of mutual information terms that involve mutually independent pairs of reciprocal random variables.

If the nodes in our setup exchange the secret bits that they have extracted with the other nodes using the Steiner graph approach of Nitinawarat et al. in order to share a group-wide secret key, then it follows from (i) and (ii) that we have met the conditions to satisfy the secret key capacity bounds on (iii).

3.6 Experimental Setup

We use Crossbow TelosB wireless sensors for our experiments. TelosB mote is a low power wireless sensor module equipped with an IEEE 802.15.4-compliant RF transceiver (the TI CC2420), built-in antenna, and a microcontroller. The motes are programmed to exchange probe packets and collect RSS measurements, as described by Wilson et al. [48]. This sensor network platform allows us to readily explore the impact of using multiple sensors on secret key extraction.

In this work, we use $2N$, $N \in \{1, \ldots, 5\}$ sensors operating on batteries, divided into two groups of N each representing Alice and Bob. Another sensor connected to a laptop acts as a base station that collects data from all the other $2N$ battery powered sensors; i.e., in our experimental setup, the access points are colocated, and

the measurements are exchanged over an insecure channel for evaluation purposes. Nodes representing Alice are numbered 0 to $(N-1)$ and those representing Bob are numbered N to $(2N-1)$. Node numbered $(i+1) \mod 2N$ transmits a probe packet after it hears a probe packet sent from node numbered i, where $0 \leq i \leq (2N-1)$. With this setup, one cycle of probe exchanges between Alice and Bob is shown in Figure 3.3.

We choose to perform our experiments in a dynamic environment, where one set of nodes is constantly moving and/or rotating and which has several intermediate objects. In such an environment, the presence or the absence of line of sight changes unpredictably over time. Hence, the extracted keys are expected to be unpredictable. In our earlier work [14], we showed that in static scenarios, when Eve positions herself strategically on the signal path between Alice and Bob, she can cause the predictable channel/key generation attack. Thus, our choice of a dynamic environment for the experiments makes it harder for Eve to cause such attacks.

For each $N \in \{1, \ldots, 5\}$, we perform two kinds of experiments - slow-walk and iRobot rotation experiments. Our experimental setup is shown in Figure 3.4. In all our experiments, all the N nodes of both Alice and Bob are arranged in a circular pattern (radius ≈ 15 cm), as shown in Figure 3.4, and Alice remains stationary in all the experiments. In the slow-walk experiments, all the nodes representing Bob are fixed on a board and are carried around in a student lab at normal walking speed while maintaining a distance of about $1-5$ m between the nodes of Alice and Bob. In the iRobot rotation experiments, all the nodes representing Bob are fixed onto the rim of a remote controlled iRobot Create rotation platform. The iRobot is rotated at a roughly constant rate of about 22 rotations per minute, and a distance of about 3 m is maintained between the nodes of Alice and Bob.

Note that in our iRobot experiments, in order for two rotating, neighboring nodes (in the circular arrangement of TelosB sensors) to record similar RSS variations, they have to rotate at a very high speed and also rotate exactly into one another's positions. Neither of these two things happens in our experiments. In our rotation experiments, the iRobot completes about 22 rotations per minute, which is a very low speed and does not even come close to the rate at which the channel changes.

Secondly, note that while the TelosB sensors are fixed onto the rim of the iRobot platform, the iRobot itself is not fixed on the ground in order to allow for its rotation. However, while rotating, the iRobot also moves a very small distance sideways[1]. Therefore, it is highly unlikely for any two nodes to rotate into one another's positions. Consequently, the measurements from our experiments are not duplicated due to rotation.

In our evaluations, we assign all the nodes to use channel 11 (2.405 GHz) in the case of simple collaboration (Section 6.1). For evaluating hierarchical collaboration, we divide the nodes of Alice/Bob into two subgroups and assign channels 11 (2.405 GHz) and 21 (2.455 GHz), respectively, for these two different subgroups (Section 6.2). We set the transmission power on each TelosB sensor so that it does not change with the environment. This transmission power (0 dBm) is the highest available on the TelosB mote.

Table 3.1 shows the total number of probe packets exchanged between the nodes of Alice and Bob in each of our experiments. The Spin program [48] detects packet losses using timeouts. When node numbered $(i+1) \mod 2N$ fails to receive a probe packet from node numbered i within the timeout period, where $0 \leq i \leq (2N-1)$, node numbered $(i+1) \mod 2N$ goes ahead with its packet transmission. In our experiments, we use a timeout period of 25 ms. In our work, we utilize a cycle of probe packets, only if there are no packet losses in that cycle. Table 3.2 shows the fraction of probe packets that we utilize for key extraction in each of our experiments. It turns out that on average, 98.52% of the probe packets are utilized in the key extraction process.

3.6.1 Interpolation Stage

A prequantization stage called interpolation is used in the work of Patwari et al. [45] to reduce the probability of bit mismatch between the bits of Alice and Bob. In this work, we also use interpolation to considerably reduce the probability of bit mismatch so that the information reconciliation stage reveals a lesser fraction of information.

[1]This distance is very small in comparison to the distance between Alice and Bob.

When Alice and Bob use time-duplex transceivers, there will be a short delay between the time instants when Alice and Bob measure the channel, which introduces asymmetry inbetween their measurements. Interpolation addresses this asymmetry by estimating the measurements of Alice and Bob at common time instants. Let T_R denote the time delay between two subsequent measurements of Alice (or Bob). Let $\tau_a(i)$ and $\tau_b(i)$ denote the time instants at which Alice and Bob record the i^{th} measurement, respectively. The fractional sampling offset, μ[2] is defined as follows:

$$\mu = \frac{1}{2}\left[\frac{\tau_b(i) - \tau_a(i)}{T_R}\right]; \text{where } \tau_a(i) < \tau_b(i) \tag{3.1}$$

If Alice delayed its i^{th} measurement by $(1+\mu)T_R$, and Bob delayed its measurements by $(1-\mu)T_R$, we would have simultaneous estimates for the i^{th} measurement. We use the cubic Farrow filter implementation [45] to compute these estimates, which become the input to the quantization stage.

The complete process of wireless RSS-based secret key extraction that we use in wireless sensor networks, unless we specify otherwise, is shown in Figure 3.5.

The fractional sampling offset, μ, as defined above is applicable only when there is a single channel between Alice and Bob. When there are multiple channels, the fractional sampling offset depends on a specific node-pair, because over a duration of one sampling period, the measurements on each channel are made at different time instants, as shown in Figure 3.3.

Let a and b be two sensors belonging to Alice and Bob, respectively. Assume that $a \in \{0, \ldots, N-1\}$ and $b \in \{N, \ldots, 2N-1\}$. Then, the fractional sampling offset to estimate a pair of channel measurements at nodes a and b is given by,

$$\mu_{ab} = \frac{1}{2}\left[\frac{b-a}{2N}\right] \tag{3.2}$$

As an example, Figure 3.3 shows the computation of μ_{05}, and the subsequent delays, which node 5 and node 0 must use to estimate the measurements on channel "$0, 5$" and channel "$5, 0$" at the same time instant.

[2]The fractional sampling offset μ used in interpolation is different from μ in $\mu \pm (\alpha \times \sigma)$, which represent the quantization thresholds.

3.6.2 Reducing the Effects of Shadow Fading

In this section, we show that applying a running average filter to the channel measurements effectively removes the effects of shadow fading, which are caused by obstructions in the environment. Thus, the secret key bits that are obtained ultimately are primarily due to the hard-to-predict effects of fast-fading (or small-scale fading), which are caused because of the relative motion between the radios and the different objects in the environment.

We use the [49] statistical model, which provides an exponentially-decaying correlation for the shadow fading signal. The autocorrelation function of the Gudmundson shadow fading signal is represented as $R_A(k) = \sigma^2 a^{|k|}$, where $a = \epsilon_D^{vT/D}$. Here σ^2, ϵ_D, v and T represent the variance, correlation between two points separated by distance D, speed, and sampling period, respectively.

Since the autocorrelation function, $R_A(k)$, and the power spectral density, $S_A(\phi)$, are discrete Fourier transform pairs according to the discrete-time Wiener-Khintchine theorem [50], it follows that

$$S_A(\phi) = \sigma^2 \left[\frac{1 - a^2}{1 + a^2 - 2a\cos(2\pi\phi)} \right]. \tag{3.3}$$

The impulse response of the running average filter is,

$$h_n = \begin{cases} -\frac{1}{(2M+1)} + \delta_{n-M} & \text{if } n \in \{0, 1, 2, \ldots, 2M\} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where δ_n represents the Kronecker delta function, which equals 1 for $n = 0$ and 0 otherwise.

The magnitude-square of the transfer function of the running average filter, $|H(\phi)|^2$ is expressed as follows,

$$\begin{aligned} |H(\phi)|^2 = 1 - \frac{2}{(2M+1)} \left[\frac{\cos(2\pi\phi M) - \cos(2\pi\phi(M+1))}{1 - \cos(2\pi\phi)} \right] \\ + \frac{1}{(2M+1)^2} \left[\frac{1 - \cos(2\pi\phi(2M+1))}{1 - \cos(2\pi\phi)} \right] \end{aligned} \tag{3.5}$$

We find the power spectral density of the shadow fading signal using the following parameter values: $\sigma^2 = 0.40$, $D = 1$ m and $\epsilon_D = 0.16$, which were obtained as a result of extensive measurements by [51]. For slow-walk experiments, $v = 0.8889$ m/s, which corresponds to a speed of approximately 2 miles per hour. For rotation

experiments, we determine the linear speed using the relationship: $v = r\omega$, when a node is rotating on a circle of radius r m at a rotational rate of ω radian per second. In our rotation experiments, $r = 0.15$ m and $\omega = 22$ rotations per minute $= (44/60)\pi$ radian/s; correspondingly, $v = 0.3456$ m/s. The average sampling period, T, for the different setups is shown in Table 3.3.

Plotting the power spectral density of the Gudmundson shadow fading signal, we find that it is essentially low pass, where as the running average filter is a high pass filter. The 3 dB bandwidth of the shadow fading signal depends on the speed at which the nodes move as well as the channel sampling rate. The 3 dB low cutoff frequency of the running average filter depends on the size of the window $(2M+1)$ over which the running average of the measurements is calculated. If the parameter M is carefully chosen depending on the speed and the sampling rate, it can be ensured that the low cutoff frequency of the running average filter is greater than the bandwidth of the shadow fading signal, thereby significantly removing the effects of shadow fading.

We determine the running average filter parameter M for the different setups. For example, Figure 3.6 shows the power spectral density of the shadow fading signal for the 3×3 and 4×4 cases and the magnitude-square of the transfer function of the running average filter for $M = 24$ and $M = 18$. When $M = 24$ for the 3×3 slow-walk configuration ($M = 18$ for 4×4 slow-walk configuration), the 3 dB bandwidth of the shadow fading signal is smaller than the lower cutoff frequency of the running average filter. Table 3.4 summarizes the M values for different setups. We use these M values to eliminate the significant components of the shadow fading signal from our secret key extraction process.

3.6.3 Correlation Coefficient between the Measurements of Different Channels

We find that there is very low correlation between the measurements on different channels. To show this, we calculate the correlation coefficient, $\rho_{M_X M_Y} = \frac{cov(M_X, M_Y)}{\sigma_{M_X} \times \sigma_{M_Y}}, \forall (X \neq Y)$, where the random variables M_X, M_Y represent a pair of measurements recorded at the same group but on two different channels X, Y; $cov(M_X, M_Y)$ denotes the covariance between the random variables M_X and M_Y; σ_{M_X} and σ_{M_Y} denote the standard deviations of the random variables M_X and M_Y, respectively.

Note that both X and Y represent one of the N^2 channels $\in \{1, \ldots, N^2\}$, where N denotes the number of nodes in each group.

Table 3.5 shows the average correlation coefficient, $\overline{\rho_{M_X M_Y}}$ for the different $N \times N$ cases, where $\overline{\rho_{M_X M_Y}} = \frac{\sum_{X \neq Y} \rho_{M_X M_Y}}{\binom{N^2}{2}}$. The average correlation coefficient is very low - around 0.2 or less for the slow-walk experiments and around 0.01 or less for the rotation experiments. These results conclusively show that the knowledge of measurements on one channel does not help considerably in predicting measurements on another channel.

For an example, Figure 3.7 shows the complete correlation coefficient matrix, C for the 3×3 case in rotation configuration, where element C_{XY} of the matrix C equals $\rho_{M_X M_Y}$. As we can see in the figure, the correlation coefficient values for all the different channel pairs are almost close to zero, demonstrating that it will be difficult to predict measurements on some channel if we know the measurements on some other channel.

Important note: Consider a scenario where one of the nodes in the group is representing the adversary, Eve. In such a case, the above results tell us that even if Eve can measure some channel, it will not help her considerably in predicting the measurements on some other channel measured by a legitimate node!

3.6.4 Secret Bit Sequences from Closely-Located Nodes

We first calculate the distances between the closely-located nodes. Consider the 5 TelosB nodes that are arranged in a circular pattern as shown in Figure 3.8. Let p and q denote the closest and farthest distances between these sensors. We determine p and q as follows: Since the sum of the interior angles of a triangle equals 180°, $\theta_s = (180^\circ - \theta_p)/2$ and $\theta_t = (180^\circ - \theta_q)/2$. Since there are 5 sensors placed around the circle with equal angular separation, $\theta_p = (360/5)^\circ = 72^\circ$ and $\theta_q = 2 \times \theta_p = 144^\circ$. Note that the radius of the circle, $r \approx 15cm$. Then, applying the law of sines, $p = r \times \frac{sin\theta_p}{sin\theta_s} \approx 1.4\lambda$ and $q = r \times \frac{sin\theta_q}{sin\theta_t} \approx 2.3\lambda$; where $\lambda \approx 12.5cm$ is the wavelength for signals in the 2.4 GHz band.

Consider the five channels that we have labeled as c_1 to c_5 in Figure 3.9 for the 5×5 case in rotation configuration. Note that these channels are formed using one

node of Alice and five closely-located nodes of Bob. We apply the process shown in Section 2.3 for the measurements from each of these channels.

Assume that there are two parties. Each party tosses a fair coin K times and encodes 'head' as bit 0 and 'tail' as bit 1; i.e., each party generates a sequence of bits independently. Then, in expectation, $K/2$ (i.e., 50%) of the bits generated by these parties will match. Table 3.6 shows the percentage of bits that match between the secret bit sequences of different pairs of channels that we have shown in Figure 3.9. For example, channel pairs (c_1, c_2) produce a 47.92% match. Notice that all the nondiagonal elements in this table are close to 50%. If the secret bit extraction process is considered as a fair-coin-tossing process, the values in Table 3.6 imply that the secret bit sequences from different channels are almost mutually independent.

Let the random variables B_X, B_Y ($B_X \in \{0, 1\}$, and $B_Y \in \{0, 1\}$) represent a pair of secret bits that are extracted from two channels X, Y, respectively. Let $P_{B_X}(b_x)$ and $P_{B_Y}(b_y)$ denote marginal probability distributions of B_X and B_Y, respectively. Let $P_{B_X,B_Y}(b_x, b_y)$ denote the joint probability distribution of B_X and B_Y. Then, the mutual information ($I(B_X; B_Y)$) between B_X and B_Y is expressed as follows:

$$I(B_X; B_Y) = \sum_{b_x \in B_X} \sum_{b_y \in B_Y} P_{B_X,B_Y}(b_x, b_y) \times \log_2 \left(\frac{P_{B_X,B_Y}(b_x, b_y)}{P_{B_X}(b_x) \times P_{B_Y}(b_y)} \right) \tag{3.6}$$

Table 3.7 shows the mutual information between the secret bits that are extracted from different channels that we have shown in Figure 3.9. For example, the secret bits from channel pairs (c_1, c_2) have a mutual information of 1.20×10^{-3}. Notice that all the nondiagonal elements in this table are negligible, which implies that the secret bits from different channels are almost mutually independent.

All these observations (percentage of matching bits and mutual information) are further supported by our findings in Section 3.6.3, in which we have shown that there is very little correlation between the measurements of different channel pairs.

More importantly, even if Eve manages to plant malicious nodes very close to the other nodes of Alice/Bob with the hope of obtaining bits similar to those that are extracted at those nodes of Alice/Bob, it will be a futile attempt because the secret bits extracted by the nodes of Alice/Bob will be mutually independent from the bits obtained by Eve.

3.7 Performance of Simple Collaboration

We show how collaboration benefits key extraction in improving the differential entropy of RSS measurements in Section 3.7.1. Section 3.7.2 presents the results of per-bit entropy of secret bits that we calculate using the NIST test suite. The effect of collaboration on the rate of bit mismatch is discussed in Section 3.7.3. Sections 3.7.4 and 3.7.5 show the improvement in the rates at which the secret bits are extracted when there are a different number of nodes in each group. Section 3.7.6 presents the fundamental limits on the performance achievable with simple collaboration.

3.7.1 Approximate Differential Entropy of RSS Measurements

We estimate the quality of the information source (i.e., set of RSS measurements) for all $N \times N$ cases ($N \in \{1, \ldots, 5\}$) using approximate differential entropy. Higher entropy implies more randomness in the information source, and vice-versa. Assuming that the measurements in a wireless channel follow a Gaussian distribution, there are $K = N^2$ correlated, Gaussian random variables corresponding to each channel in an $N \times N$ configuration. The differential entropy of a multivariate Gaussian random vector, $X = [X_1, X_2, ..., X_K]'$, is given by,

$$h(X) = \log_2 \sqrt{(2\pi e)^K |\Sigma|}$$

where $|\Sigma|$ is the determinant of the covariance matrix.

Our differential entropy estimation is an approximation in that the data is assumed to be multivariate Gaussian. More accurate differential entropy approximations use higher order cumulants, but they are based on adjustments to the formula for Gaussian data [52]. Further note that we use approximate differential entropy only to show the increase in entropy of channel measurements as a function of N, and not for the estimation of entropy per bit of the output secret bitstream, for which we use the more accurate NIST test suite.

The covariance matrix for each $N \times N$ configuration is computed using the measurements collected from our experiments. Figure 3.10 shows the approximate differential entropy of RSS measurements for the iRobot rotation experiments; similar results are obtained for the slow-walk experiments. The degree of uncertainty, i.e., the differential entropy, increases almost quadratically with N. This shows that a

set of measurements obtained from multiple wireless channels has more randomness associated with it in comparison to a set of measurements obtained from just a single wireless channel. An input source with more randomness is likely to yield an output secret key bit stream that can be considered more random, and is hence likely to be stronger. Therefore, whenever there is an opportunity, it is best to use measurements from multiple wireless channels to generate a secret key.

While the number of channel measurements increases quadratically with N, there is a linear reduction in the sampling rate and linear increase in the time gap between bidirectional measurements between nodes across the two groups (Section 3.7.3). The increase in time gap reduces the mutual information between a pair of measurements in each direction, resulting in a high rate of bit mismatches. Therefore, the number of useful secret bits extracted in the process becomes limited when N becomes sufficiently high (Section 3.7.6).

3.7.2 Entropy of the Output Secret Bits

We estimate the entropy of the output secret key bit streams using NIST test suite's *approximate entropy test* [23]. The approximate entropy of order m, $m \geq 1$ is defined as,

$$ApEn(m) = \Phi(m) - \Phi(m+1)$$

where $-\Phi(m)$ is the entropy of the empirical distribution arising on the set of all 2^m possible patterns of length m; the NIST test suite recommends appropriate values of m depending on the length of the bitstream that we test. Notice from Table 3.8 that the output secret bit streams have an *approximate entropy* value that is close to the ideal value of 1, indicating that there is an almost 1 bit of uncertainty associated with each bit of the extracted secret key.

The privacy amplification stage uses universal hash functions chosen at random from a publicly known set of such functions, and generates fixed size smaller length output bit streams from longer input bit streams. These methods are generally based on leftover hash lemma, which is a well-known technique for obtaining random bits, i.e., bits with high entropy, from imperfect random sources [18]. The values in Table 3.8 that we have obtained using the NIST *approximate entropy test* represent only

an estimate of the entropy; even if the entropy achieved with privacy amplification was 1.0, the estimate of the entropy might not be 1.0.

We test for the randomness of the output secret bit sequences using a number of different statistical tests, in addition to the *approximate entropy test*, available in the NIST test suite [23]. Our secret bit sequences meet the input size recommendation of only the eight tests we have shown in Table 3.9; most of the other tests available in the NIST test suite require very large sequences (on the order of 10^6 bits). Each test outputs a P-value; the P-value summarizes the strength of the evidence against the null hypothesis, which corresponds to the fact that the sequence being tested is random. To pass a test, the P-value for that test must be greater than 0.01, in which case, the sequence is considered as random with a confidence of 99%. Note that all the P-values shown in Table 3.9 are at least 0.01, which demonstrates that the secret bit streams are in fact random with a very high degree of confidence.

3.7.3 Bit Mismatch Rate

Bit mismatch rate is defined as the ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted from RSS quantization. Our *bit mismatch rate* metric differs from the *bit error rate* metric used in data communications, which is meant for quantifying the percentage of transmitted bits that are received in error. Note that for secret key extraction, Alice and Bob do not actually communicate the quantized bits to one another. Rather they exchange probe packets, measure the RSS values on these probe packets, and quantize *their measurements* to derive *their bits*. Our bit mismatch rate metric captures the degree of mismatch between two bit sequences that are not actually exchanged over-the-air.

Note that the bit mismatch rate values that we calculate are based on the bits we obtain immediately after the quantization stage, and not after the final stage (privacy amplification) of the secret key extraction process. Even if there is a single mismatch between the quantized bits of Alice and Bob, the entire bit sequence becomes useless as a secret key. Therefore, we use Cascade, a well-known information reconciliation protocol (Section 2.3.2), to correct the potential bit mismatches. Cascade leaks a certain fraction of information for correcting the bit mismatches. The amount of information leakage depends on the bit mismatch rate, and it leaks all information

only when the bit mismatch rate is around 22%.

Figures 3.11 and 3.12 show the bit mismatch rates as a function of α and N, for the slow-walk and iRobot rotation experiments, respectively. Recall that α is the quantization threshold parameter, that defines the quantization thresholds - $\mu \pm (\alpha \times \sigma)$, and which also controls the amount of measurements censored in the bit extraction process.

Bit mismatch rate decreases as α increases. When we consider the *support* of two correlated random variables, each representing a measurement from Alice and Bob, it should be clear that the area, where these random variables (when quantized) disagree, becomes large as the area corresponding to the censored region is reduced. While it may appear that smaller values of α will yield more secret bits, this is not always true. The number of bits that we can extract does not continue to increase when we reduce α beyond an optimal point, as the amount of information leaked with growing bit mismatch rate offsets the gain in the number of bits that can be extracted. In fact, as we can notice from Figure 3.11, for slow-walk experiments, at $\alpha = 0.1$ and with $N = 4$ or 5, the bit mismatch rate approaches or exceeds 0.22 - the rate at which all bits are essentially leaked in the information reconciliation process.

In general, the bit mismatch rate increases with N. There are two contributing factors for high bit mismatch rates. First, the average time gap between bidirectional measurement pairs increases with N. For an $N \times N$ configuration, this time gap varies between Δ and $(2N-1) \times \Delta$ (Figure 3.3 shows this variation for the 3×3 case), where Δ is the time interval between transmissions of node i and node $(i+1)$; $0 \leq i \leq 2(N-1)$. The increase in the time gap reduces the mutual information between a pair of measurements in each direction, resulting in high bit mismatch rates. In other words, such large time gaps are likely to introduce more noise, and hence, high bit mismatch rate as we increase the number of nodes.

Second, the sampling period increases with N. For an $N \times N$ configuration, the sampling period is equal to $2N \times \Delta$ (Figure 3.3 shows the sampling period for $N = 3$). When N becomes sufficiently large, the channel probing rate falls below the rate of change of the wireless channel, causing an undersampling effect, leading to high bit mismatch rates.

Rotation experiments produce bit streams with lower mismatch rates compared to the slow-walk experiments. Although both types of experiments are done in similar kind of setups, there is a notable difference between the rotation and slow-walk experiments. In the slow-walk experiments, the distance between Alice and Bob changes considerably over time as Bob continues to move along its trajectory. Secondly, for a large fraction of the time, this distance is also larger than that with the rotation experiments. Large distance reduces the signal-to-noise-ratio (SNR), and hence, the measurements from slow-walk experiments produce bits with higher mismatch rate.

3.7.4 Secret Bits per Probe

Secret bits per probe is defined as the average number of secret bits extracted per probe transmission. This rate is measured in terms of final output bits produced after taking care of bit losses due to information reconciliation and privacy amplification.

Figures 3.13 and 3.14 show the secret bits per probe as a function of bit mismatch rate and N, for the slow-walk and iRobot rotation experiments, respectively. Comparing the peaks in the case of slow-walk experiments, the 2×2 case shows about 77% increase in the number of secret bits extracted per probe in comparison to the 1×1 case. A further increase in N does not improve the performance. Comparing the peaks in the case of rotation experiments, the 2×2, 3×3, and 4×4 cases respectively show about 160%, 246%, and 332% increase in the number of secret bits extracted per probe in comparison to the 1×1 case. However, the increasing trend does not continue, as the 5×5 case shows an increase of 300%, which is less than the gain achieved with the 4×4 case.

3.7.5 Secret Bits per Joule of Transmission Energy

Secret bits per Joule of transmission energy is defined as the average number of secret bits extracted per Joule of energy consumed in the process of transmitting a probe packet. This rate is measured in terms of final output bits produced after taking care of bit losses due to information reconciliation and privacy amplification. This metric forms the basis for comparing the energy efficiency.

There are three distinct phases involved in the transmission of a probe packet - (1) copying the packet data from memory to the FIFO buffer on the radio, (2) backoff

transmission for a random delay when the medium is sensed to be busy, and (3) actual transmission of the packet. Figure 3.15 depicts these three phases for one sensor in a 2X2 configuration. In Figure 3.15, the data copying phase extends roughly from -7 ms to -3 ms; the actual packet transmission phase spans from around 0.15 ms to 1.2 ms, and the node backs off from transmission inbetween these two phases. Note that copying data consumes slightly more power than the actual packet transmission, which is also observed in an existing work [53]. Further note that our measurements closely match with the specifications [54] and existing work [53]. The minor difference of a few mW between the measurements and the specifications comes from the fact that the MSP430 microprocessor, which still consumes this small amount of power, is not turned off during our measurements. Consequently, our measurements are more realistic and better reflect the operating scenario of the TelosB sensors exchanging the secret key. Our power measurement experiments also enable us to determine the *durations* of the different phases of a packet transmission, which are needed in the calculation of the total energy consumption, as we describe in this section.

In the Spin program [48], the packet length equals $(2N + 6)$ bytes, where $2N$ is the total number of nodes in the network. Correspondingly, the durations of the data copying and actual packet transmission phases increase with N, which we can clearly observe from Figure 3.16 and Figure 3.17. For all the $N \times N$ configurations, the amount of power consumed is roughly the same, but as we increase N, the same amount of power is consumed for a slightly longer period due to increase in the packet length. Thus, increasing N slightly increases the energy consumption in the process of transmitting a probe packet. However, as we shall see in the following, the reduced number of packet transmissions due to collaborative key extraction actually contributes to significant reduction in the overall consumption of transmission energy.

Secret bits per Joule of transmission energy is calculated as follows. Let $P(t)$ denote the power consumed by the sensor at an arbitrary time instant t. Let E denote the total amount of energy consumed across the aforementioned phases in transmitting a probe packet. Let t_0 denote the time instant at which the data copying phase (phase 1) is initiated. Let t_1, t_2, and t_3 denote the time instants at which the phases 1, 2, and 3 get completed, respectively. The duration of phase 2 (random

backoff) is nondeterministic; i.e., $(t_2 - t_1)$ could vary with each packet transmission. To ensure that our results are not heavily influenced by this nondeterministic component, and therefore to provide fair comparison across different number of nodes, we use the same backoff period in our energy calculations irrespective of the number of nodes. Correspondingly, the energy consumption over phase 2, E_{bp}, is treated as a constant, where E_{bp} equals 60.42 μJ. Then,

$$E = \int_{t_0}^{t_1} P(t)\, dt + E_{bp} + \int_{t_2}^{t_3} P(t)\, dt$$

Let S denote the number of secret bits extracted and k, the number of probe packets transmitted. Thus, the number of secret bits extracted per Joule of transmission energy, $S_J = \frac{S}{k \times E}$ bits per Joule.

In the slow-walk experiments, the 2×2 configuration achieves about 75% increase in secret bits per mJ of Tx energy in comparison to the 1×1 case. A further increase in N does not improve the performance. In Figure 3.18, we show the secret bits per mJ of transmission energy as a function of bit mismatch rate and $N \times N$, for the iRobot rotation experiments. Comparing the peaks, the 2×2, 3×3 and 4×4 cases respectively show about 158%, 239%, and 319% increase in the number of secret bits extracted per mJ of transmission energy in comparison to the 1×1 case. However, the increasing trend does not continue to the 5×5 case. Nevertheless, from Figure 3.18, we can clearly see that collaboration is significantly energy efficient in comparison to the 1×1 case.

Note: The reduced energy consumption is further split among N different nodes, implying that the battery on each sensor will last substantially longer because of collaboration.

3.7.5.1 Energy Efficiency when Probe Packet Size Is Constant

We have noted earlier that in the SPIN program [48], the packet length and therefore its transmission delay increases with N. However, it is possible to construct probe packets whose size is independent of N. We evaluate the energy efficiency in using probe packets that are two bytes long. Note that sixteen bits are long enough to encode the node id of the sender, and the piggybacked, encrypted quantized bits,

for all values of N we use in this work. Figure 3.19 shows our estimates of the peak secret bits per mJ of Tx energy vs N. A smaller, constant sized packet improves energy efficiency only marginally. It shows that changing the size of probe packets that a TelosB sensor transmits has only a minor impact on the energy consumption.

3.7.6 Trade-off Discussion

Since differential entropy increases with N, collaboration enables the extraction of stronger secret keys by increasing the number of nodes. However, this increasing trend is not reflected in the case of secret bit rates beyond specific values of N. Notice from Figures 3.13, 3.14, and 3.18 that, for each N, the secret bit rate (per probe, or per Joule of Tx energy) starts increasing until some point and then starts declining. We find that these secret bit rates peak at an optimal α value, which lies in the range of $0.3 - 0.7$, and then starts diminishing on either side of this optimal point. For each N, the secret bit rates do not continue to increase when we reduce α beyond this optimal point, as the amount of information leaked with growing bit mismatch rate offsets the gain in the number of bits that we can extract.

When we consider the peak secret bit rates across different values of N, we find remarkable improvements only up to some value of N, i.e., the secret bit rates peak at a point (at $N = 4$ and $N = 2$ for rotation and slow-walk experiments, respectively), and then start declining. We identify this declining behavior to be a result of high bit mismatch rates which offset the gain in the number of secret bits that we can extract.

Thus, there is a clear trade-off in using simple collaboration for extracting secret keys. We capture this trade-off in Figure 3.20. While we obtain stronger secret keys with increasing N, the secret bit rates peak at relatively small values of N.

3.7.7 Performance under Constant Sampling Rate

We have seen the performance of simple collaboration as a function of N, for the case where the channel sampling rate decreases linearly with N, from Section 3.7.1 to Section 3.7.6. In this subsection, we show the performance for the case when the sampling rate is the same for all values of N. We find that in such a setup, there is higher bit mismatch rate in comparison to the setups we have considered thus

far. To solve this problem, we introduce a *distillation* stage[3] in our key extraction methodology comprising the quantization, information reconciliation, and privacy amplification stages. The distillation stage, introduced between the quantization and the information reconciliation stages, iteratively improves the output from the quantizer by eliminating measurements that are likely to cause mismatching bits at Alice and Bob. This stage ensures that the percentage of mismatching bits is low enough to be handled by information reconciliation without compromising security. In fact, without the distillation stage, the information reconciliation stage by itself is unable to reconcile the bit mismatch.

3.7.7.1 Experimental Setup

We conduct a few experiments in a student lab to evaluate the effectiveness of distillation. Nodes representing Alice remain stationary in one corner of the lab while the other set of nodes (Bob) is carried around at normal walking speed. The distance between Alice and Bob is maintained between 2 m - 8 m. Nodes of Alice and Bob are arranged in two parallel rows, with each sensor separated from its neighbor by a distance of about 12 cm, which is greater than the de-correlation distance of 6.25 cm for signals transmitted in the 2.4 GHz band. This ensures that the measurements collected at neighboring nodes are mutually uncorrelated. Therefore, we use the secret bit extraction process (shown in Figure 3.21) separately for each one of the N^2 channels, where N represents the number of nodes at Alice/Bob. We extract two bits from each RSS measurement that we collect in this setup.

3.7.7.2 Prohibitively High Bit Mismatch

When using multiple sensors, we find that the bit mismatch rate is significantly higher in comparison to our earlier experiments that use 802.11 single antenna systems. Note that for a mismatch rate of about 22%, the information reconciliation protocol essentially reveals all the bits. Therefore, the collected measurements that exhibit very high bit mismatch are not useful in establishing a secret key.

---

[3]The distillation stage as described in this work does not involve any exchange of parity information, and is different from the advantage distillation in quantum cryptography.

We identify the following reasons for such high mismatch rates. First, when multiple nodes take turn in exchanging probe packets, it increases the average time-gap between any pair of measurements taken in each direction of a channel, and also reduces the probing rate on each channel. Both these factors contribute in increasing the bit mismatch rate. This is also verified in a plot of bit mismatch rate vs channel distance, where channel distance is the absolute difference between the node ids (as defined by the token ring order) of the transmitting and receiving sensors. Figure 3.22 clearly shows the general increase in mismatch rate with channel distance. Time-gap between each unidirectional measurement pair is proportional to the channel distance. So, mismatch rate increases with channel distance/multiple antennas.

Second, channels in 802.15.4 are much narrower in comparison to 802.11. A nonreciprocal deep fade (perhaps due to strong interference only at Alice) occurring on a narrow channel significantly reduces the average RSS computed at Alice while not affecting much at Bob. This results in a greater likelihood of asymmetry in measurements, and therefore higher bit mismatch when using narrow channel measurements.

3.7.7.3 Distillation

To address the problem of very high bit mismatch rates, we augment the secret key extraction process with the distillation stage. Distillation ensures that the percentage of mismatching bits is low enough for information reconciliation to correct the differences without revealing all the extracted bits. Figure 3.21 shows the distillation stage in relation to the other stages of the key extraction process.

Plotting the measurements from channels with large channel distances, we find that a large fraction of consecutive measurements exhibit abrupt transitions from one quantization level to another, resulting in asymmetry. The distillation stage seeks to iteratively eliminate such measurements causing abrupt transitions. If the mismatch is still too high even after one round of eliminations, it is necessary to eliminate further; in which case, the next best elimination candidates are those that follow the previously eliminated measurements. When this process is iterated over a number of times, it is likely to improve the bit mismatch rate. Note that the number of iterations required depends on the *current expected mismatch rate* of the channel, which can

be determined based on the *history of mismatch rate of the channel*. Algorithm 4 succinctly expresses the steps taken in each iteration. Essentially, in a given block of at least i quantization labels, which are identical (e.g., consecutive a's), iteration number i removes the prefix of length i from that block; in case the block length is less than i, it removes the entire block.

Algorithm 4 assumes that the quantizer outputs the labels (e.g., a, b, c, d) of each quantization interval instead of the actual bit pattern assigned to each interval. *exclude_label* is a special label indicating an eliminated measurement. In each iteration, the distiller processes the input as shown in Algorithm 4. For the first iteration, the distiller gets its input from the quantizer, and for the successive iterations, the distiller's output becomes the input for the next iteration. In the last iteration, the distiller outputs the bit patterns corresponding to each quantization interval. The following example shows two iterations of distillation; the _ symbol represents the *exclude_label*.

Distiller Input: $aaaaabbaaaabbbbbaaaa\cdots$

Iteration 1 output: $\_aaaa\_b\_aaa\_bbbb\_aaa\cdots$

Iteration 2 output: $\_\_aaa\_\_\_\_aa\_\_bbb\_\_aa\cdots$

Figure 3.23 shows the improvement in bit mismatch rate with each iteration for the 5×5 configuration. Without distillation, the average mismatch rate is about 23%, in which case information reconciliation leaks out all the bits. However, two iterations of distillation reduces the mismatch rate to a sufficiently small value ($< 5\%$) for efficient information reconciliation. Thus, despite the simplicity of the distillation approach, these results show that it can reduce the bit mismatch rate very effectively.

3.7.7.4 Gain in Secret Bit Rate

Figure 3.24 shows a plot of the secret bit rate as a function of number of nodes at Alice/Bob. It can be clearly seen that the secret bit rate increases almost linearly with the number of nodes when the sampling rate is the same for all the $N \times N$ cases. We also measure the randomness of the extracted bit streams using NIST's approximate entropy test. We find that the entropy values for the extracted secret bit streams from all the $N \times N$ configurations ($1 \leq N \leq 5$) are close to 1, the ideal value.

3.8 Performance of Hierarchical Collaboration

In our evaluation of hierarchical collaboration, we use 4 nodes each at Alice and Bob, respectively, in the slow-walk configuration. Following our discussion in Section 3.7.6, we limit the size of each subgroup to 2 nodes. We assign channels 11 (2.405 GHz) and 21 (2.455 GHz), respectively, for the different subgroups of Alice and Bob. The two subgroups of Bob switch between these frequencies for every $n_p = 100$ packet transmissions. For collecting the measurements from all the group members, in our evaluation, we have used two TeloB nodes that are connected to the laptop. One of the nodes collects measurements from all the group members operating on channel 11, while the other node collects measurements from all the group members operating on channel 21. The peak performance comparison of simple and hierarchical collaboration is shown in Table 3.10.

Since there are two collaborating groups operating in parallel, it results in substantially higher secret bits per second (an increase of about 160%) using hierarchical collaboration in comparison to simple collaboration. The bit mismatch rate is substantially lower with the hierarchical approach, 5.5%, vs 9.5% for simple collaboration. Although more measurements are recorded per probe packet in the simple approach compared to the hierarchical approach, a large fraction of the quantized bits is lost due to the information reconciliation stage to reconcile the differences in the bit sequences with such high bit mismatch rates. Consequently, the hierarchical approach outperforms the simple collaborative approach in terms of secret bits per probe and secret bits per mJ of transmission energy as well, with increases of about 33% and 35%, respectively.

The per-bit entropy of the output secret bits for both the approaches have comparable values that is close to the ideal value of one. However, the hierarchical approach produces measurements with higher differential entropy in comparison to the simple collaboration approach, because the hierarchical approach exploits both space diversity and frequency diversity, and therefore, the collected measurements are also affected by frequency selective fading. Thus, we can clearly see that a combination of space and frequency diversity with frequency switching offers a very good performance.

3.9 Summary of Collaborative Secret Key Extraction

We summarize the results of our collaborative key extraction approach as follows.

1. Collaboration enables the extraction of stronger secret keys from more random information sources because it significantly improves the differential entropy of the collected measurements that is quadratic in the number of nodes.

2. Collaboration among sensor nodes allows for faster key extraction and in an energy efficient manner.

3. Simple collaboration, exploiting space diversity, improves performance only with a small number of nodes due to the fundamental trade-off between the quadratic increase in the number of channel measurements due to multiple nodes per group versus a linear reduction in sampling rate and a linear increase in the time gap - both contributing to high bit mismatch rates, and which ultimately reduce the rate of secret key extraction.

4. Hierarchical collaboration - (i) overcomes the high mismatch rate problem by limiting collaboration among smaller subsets of nodes, (ii) achieves faster key extraction by exploiting frequency diversity in addition to space diversity, and (iii) further improves the differential entropy of collected measurements through frequency switching.

We do not claim that our secret key extraction will work under all possible real-world scenarios[4]. Nor do we claim that the scenarios we use in our experiments represent all of the real-world scenarios. However, we do believe that our technique advances the state-of-the-art of secret key extraction from wireless channel characteristics and we show that it works well under our experimental set up. We expect our scheme to also work under similar real-world scenarios, especially when the spatio-temporal characteristics of the wireless channels cannot be predicted.

---

[4]In our earlier work [14] for a 1×1 link, we demonstrated that secret key extraction works well only under nonstatic scenarios.

3.10 Related Work

While there is an extensive amount of research on using radio channel properties for secret key extraction (e.g., [14, 20, 38, 41]), very few existing works have developed these ideas in the context of sensor networks. Aono et al. [22] used a steerable directional antenna in combination with Zigbee radio hardware to generate a secret between two nodes and to test what an eavesdropper would have received. More recently, [45] introduced high rate uncorrelated bit extraction (HRUBE), a framework for interpolating, transforming for de-correlation, and encoding channel measurements using a multibit adaptive quantization scheme. Patwari et al. analyzed the probability of bit disagreement using a Gaussian assumption, and presented experimental results for bit extraction using TelosB sensors. Our research significantly enhances these existing works on secret key extraction in sensor networks by using collaboration among sensors to obtain high entropy bits at a much higher bit rate.

The use of multiple antennas is not a new idea. [44] presented a theoretical study on the use of MIMO radio channels for enhancing secret key extraction. Even though we achieve multiple antenna capability through the use of collaborating sensor nodes, our system differs from a typical MIMO transceiver because in our setup, the signals transmitted from different nodes are not synchronized - nodes need to take turns in transmitting their packets. Therefore, we cannot directly apply the methods of Wallace et al. for our setup. Instead, a multiple node setup introduces new challenges in terms of higher bit mismatch, which we address using interpolation, information reconciliation, etc.

Our work differs from the work of Wallace et al. in the following significant ways. First, we obtain the multi-antenna capability using multiple sensor nodes instead of MIMO. Second, unlike a MIMO-based scheme, our research does not assume any phase synchronization of RF signals. Third, we use real-world measurements from TelosB sensors placed in the slow-walk or iRobot configurations for our evaluations. Fourth, we extract secret bits using a 4-stage key extraction process that combines the interpolation stage from [45] with the other stages from our earlier work [14]. Last, we also address the energy efficiency in our research.

3.11 Conclusion

We proposed and experimentally evaluated a collaborative secret key extraction scheme for wireless sensor networks. Our experimental results showed that there is a significant increase in secret bit rate per second and per probe as well as per mJ of transmission energy, due to collaboration. We also evaluated the fundamental performance trade-off due to the increased number of measurements versus the increased bit mismatch rate. While it may appear that collaboration requires many nodes, leveraging measurements from many different sensors enables extraction of stronger secret keys at a faster rate and in an energy efficient manner. While the hierarchical approach uses more bandwidth (i.e., more than one channel), it correspondingly reduces the duration over which the channels are occupied. We have presented these results in three papers [55, 43, 56]. In the future, we will evaluate our collaborative secret key extraction scheme with more experiments in different kinds of environments. In this chapter and the previous chapter, we have seen how wireless nodes can exploit the *physical layer measurements* representing the wireless channel characteristics for establishing a secure communication channel at *upper layers* between these nodes. In the next chapter, we show how the choice of *physical layer pulse shape* can impact the efficiency of data transmission for *best-effort data traffic*.

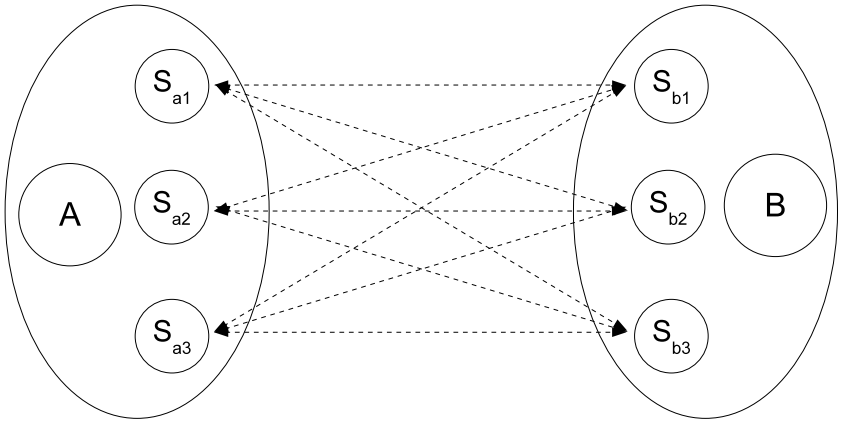

Figure 3.1. Simple collaboration exploits variations across N^2 (here, $N = 3$) channels. Sensors S_{ai}, and S_{bi} belong to access points A and B, respectively.

Algorithm 1 Simple Collaboration Method

procedure SCM(A, B, S_a, S_b, N)
▷ A, B - access points; S_a, S_b - sets of sensors in groups A, B; N - number of nodes in each group

repeat
for $i \leftarrow 1, N$ **do** ▷ group A's transmissions
node S_{ai} transmits a probe packet
for $j \leftarrow 1, N$ **do**
node S_{bj} records an RSS measurement
end for
end for
for $i \leftarrow 1, N$ **do** ▷ group B's transmissions
node S_{bi} transmits a probe packet
for $j \leftarrow 1, N$ **do**
node S_{aj} records an RSS measurement
end for
end for
until enough measurements to establish key
for $i \leftarrow 1, N$ **do**
node S_{ai} shares measurements / quantized bits with A over secure channel sc_a
node S_{bi} shares measurements / quantized bits with B over secure channel sc_b
end for
end procedure

Algorithm 2 Hierarchical Collaboration Method

procedure HCM(A, B, S_a, S_b, N, N_s)
▷ A, B - access points; S_a, S_b - set of sensors in groups A, B; N - number of nodes in each group; N_s - number of nodes in each subgroup

$num\_sub\_groups \leftarrow \frac{N}{N_s}$
for $i \leftarrow 1, num\_sub\_groups$ **do**
assign frequency c_i to nodes $\in$ subgroup S_{ai}
assign frequency c_i to nodes $\in$ subgroup S_{bi}
end for
for $i \leftarrow 1, num\_sub\_groups$ **do**
run $SCM(A, B, S_{ai}, S_{bi}, N_s)$ ▷ parallel execution
end for
end procedure

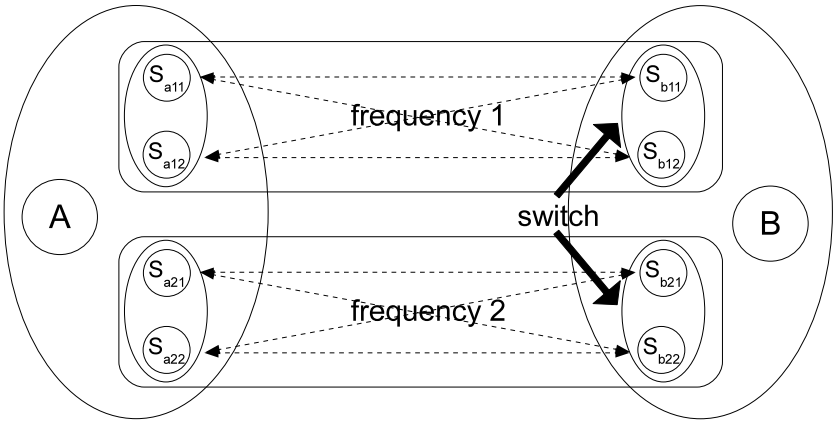

Figure 3.2. Hierarchical collaboration. Subgroups S_{a1}, S_{a2} assigned to frequencies 1 and 2, respectively. Subgroups S_{b1}, S_{b2} switch between frequencies 1 and 2 periodically to produce N^2 channels (in this figure, $N = 4$).

Algorithm 3 Frequency Switch Algorithm

procedure FREQUENCY SWITCH PROCEDURE(c_i, n_c, n_p)
▷ c_i - channel initially assigned to calling node; n_c - number of available channels; switch frequency after every n_p transmissions; available channels = $[0, 1, ...(n_c - 1)]$

curr_channel $\leftarrow c_i$
loop
for $i \leftarrow 1, n_p$ **do**
transmit probe packet on *curr_channel*
end for
curr_channel $\leftarrow$ (*curr_channel* + 1) mod n_c
end loop
end procedure

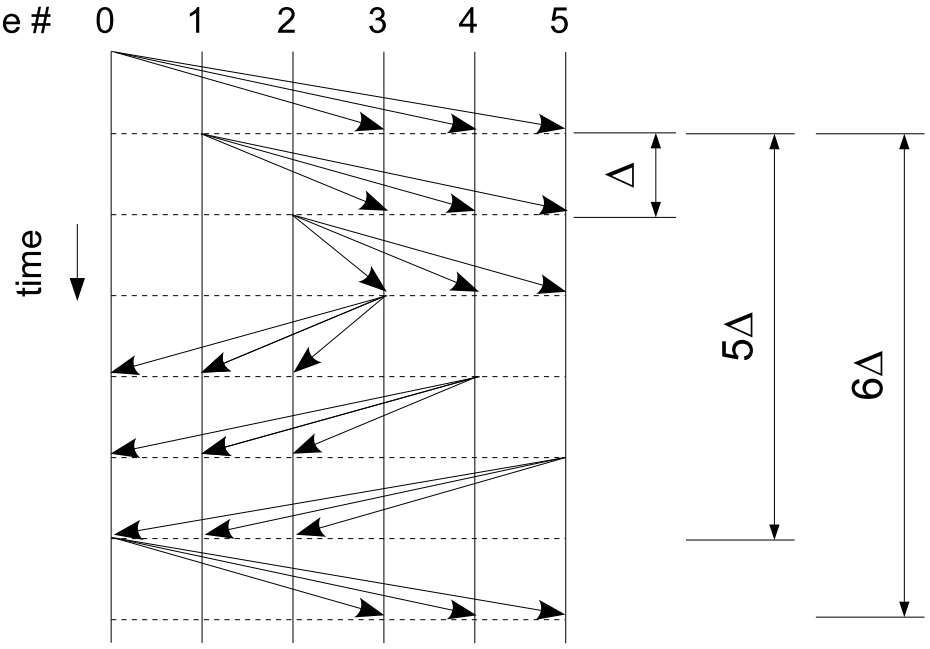

Figure 3.3. Timing diagram for 3×3 setup. $\{0, 1, 2\} \in$ Alice; $\{3, 4, 5\} \in$ Bob. Sampling period, $T_R = 6\Delta$. Fractional sampling offset, $\mu_{05} = \frac{1}{2}\left[\frac{5\Delta}{6\Delta}\right] = \frac{5}{12}$. Measurements on channel "$0, 5$" delayed by $(1 + \mu_{05})T_R = 8.5\Delta$. Measurements on channel "$5, 0$" delayed by $(1 - \mu_{05})T_R = 3.5\Delta$.

Table 3.1. Total number of probe packets exchanged between the nodes of Alice and Bob

Configuration	**Slow-walk experiment**	**iRobot rotation experiment**
1×1	34446	30486
2×2	43251	30977
3×3	57885	28739
4×4	68781	30406
5×5	63332	32064

Table 3.2. Fraction of probe packets utilized in the key extraction process

Configuration	**Slow-walk experiment**	**iRobot rotation experiment**
1×1	0.9987	0.9677
2×2	0.9960	0.9920
3×3	0.9554	0.9938
4×4	0.9935	0.9906
5×5	0.9888	0.9830

Table 3.3. Average sampling period, T for different setups

$N \times N$	T **second (slow-walk)**	T **second (rotation)**
1×1	0.0202	0.0214
2×2	0.0415	0.0415
3×3	0.0672	0.0641
4×4	0.0878	0.0879
5×5	0.1139	0.1142

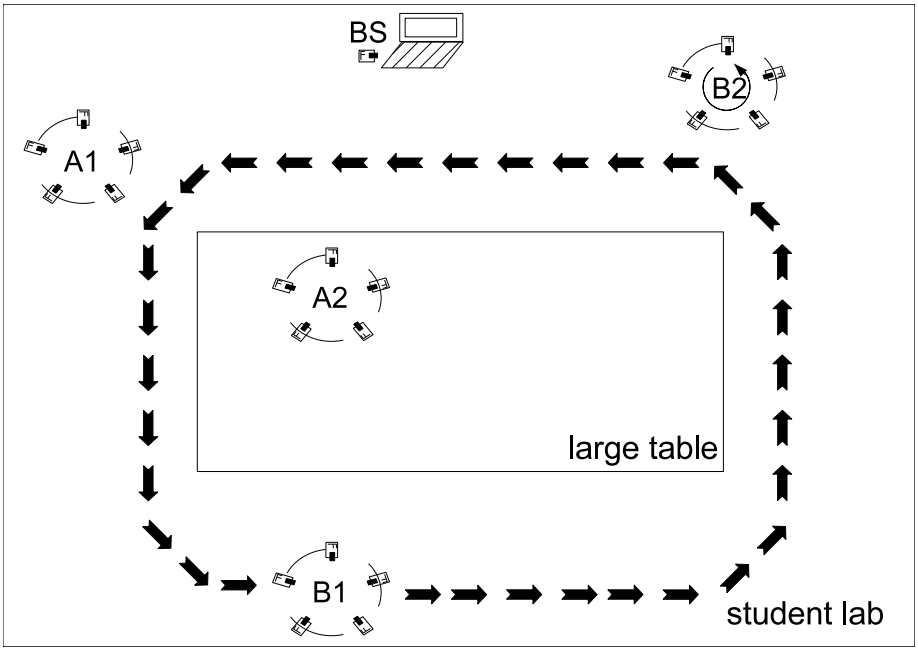

Figure 3.4. Experimental setup. $\{A1, B1\} \in$ slow-walk experiments. $\{A2, B2\} \in$ rotation experiments. BS - base station. A1, A2 - stationary. B1 - moves along the trajectory indicated by the arrows. B2 - rotates in place.

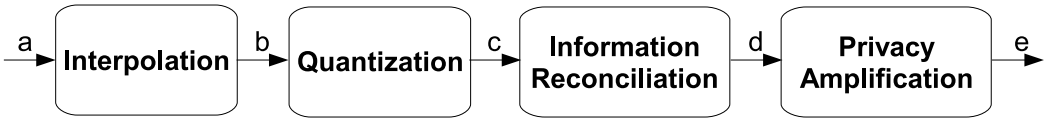

Figure 3.5. Secret key extraction process. a - RSS measurements, b - interpolated measurements, c - quantized bits, d - reconciled bits, e - secret bits.

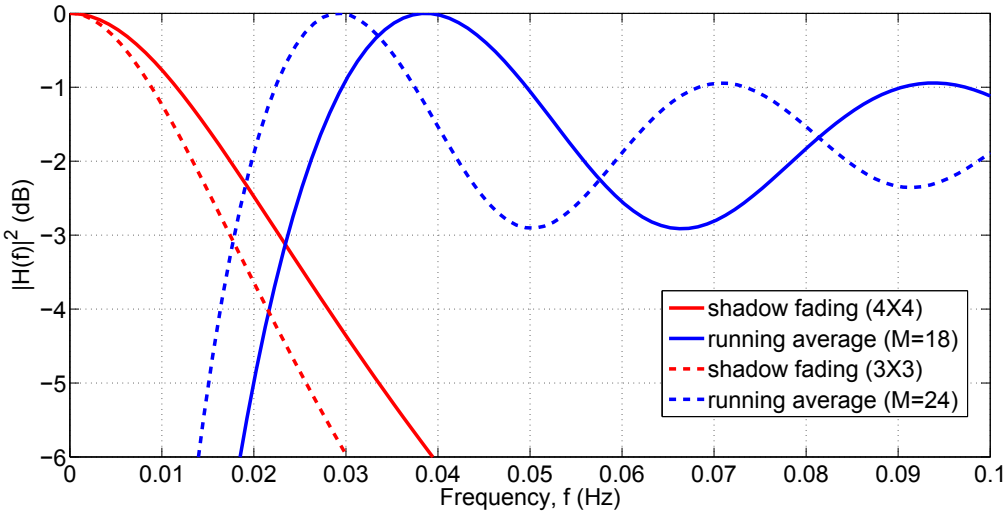

Figure 3.6. Power spectral density of the shadow fading signal for the 3×3 and 4×4 cases and the magnitude-square of the transfer function of the running average filter for $M = 24$ and $M = 18$.

Table 3.4. Running average filter parameter M that reduces the effects of shadow fading

$N \times N$	M **(slow-walk)**	M **(rotation)**
1×1	50	50
2×2	39	50
3×3	24	50
4×4	18	48
5×5	14	37

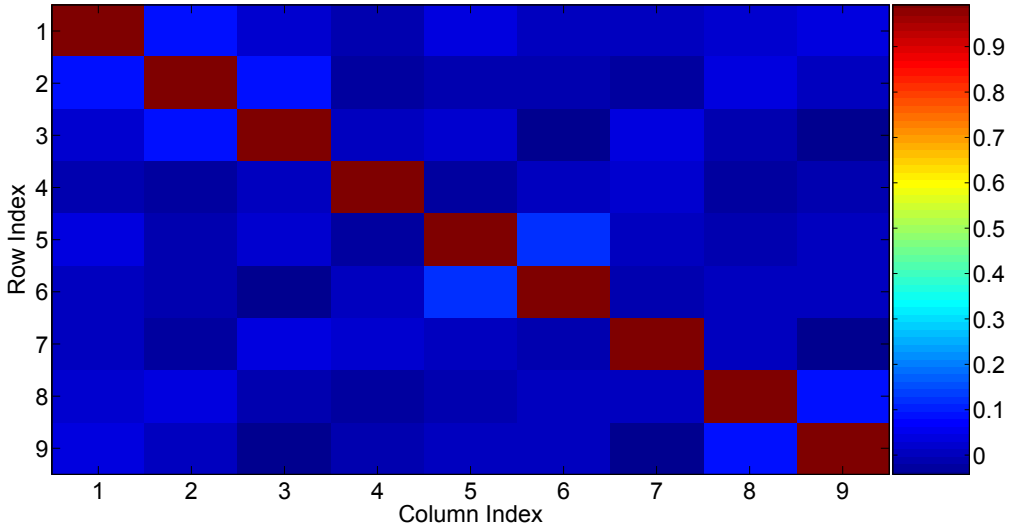

Figure 3.7. Correlation coefficient matrix, C for the 3×3 case in rotation configuration, where element C_{XY} of the matrix C equals $\rho_{M_X M_Y}$. All $C_{XY}, \forall(X \neq Y)$ are almost close to zero.

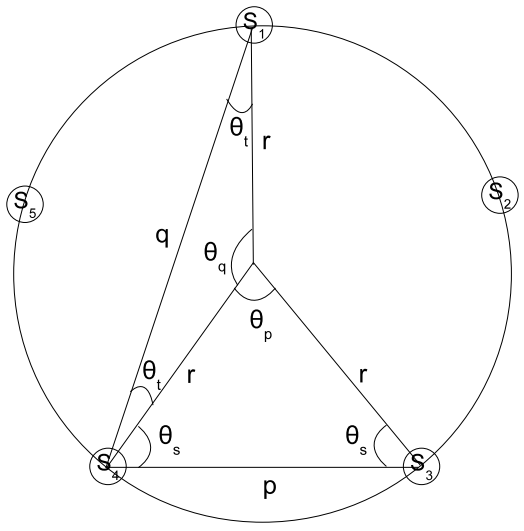

Figure 3.8. p and q denote the distances between the closest and the farthest sensors, respectively, in a circular configuration of 5 sensors. r is the radius of the circle, which is approximately equal to $15cm$.

Table 3.5. Average correlation coefficient between measurements on different channels

$N \times N$	$\overline{\rho_{M_X M_Y}}$ **(slow-walk)**	$\overline{\rho_{M_X M_Y}}$ **(rotation)**
2×2	0.1876	0.0031
3×3	0.1841	0.0119
4×4	0.2291	0.0155
5×5	0.1900	0.0039

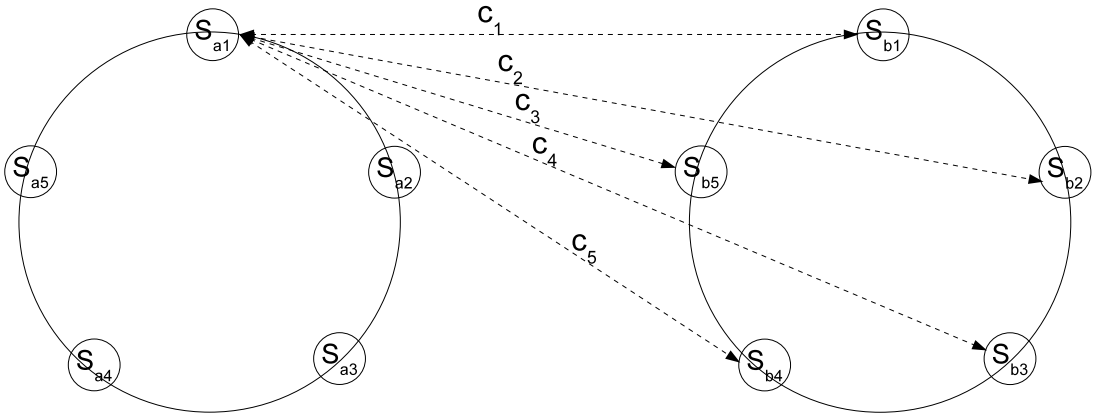

Figure 3.9. Channels between one node of Alice and 5 nodes of Bob.

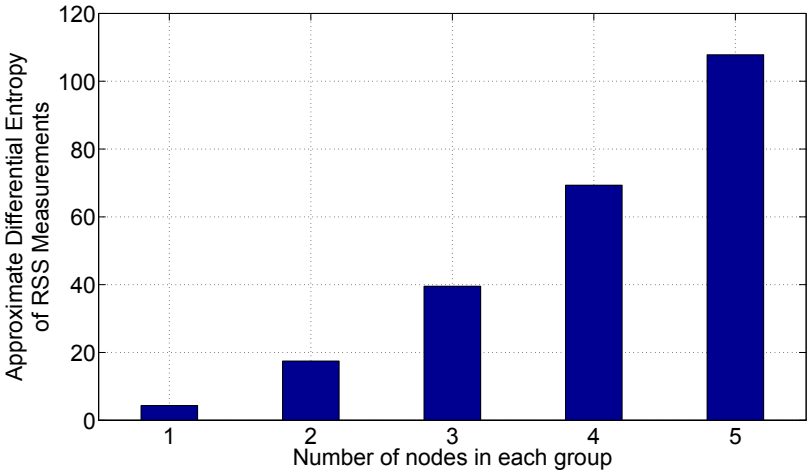

Figure 3.10. Approx. differential entropy vs N for rotation experiments.

Table 3.6. Percentage of bits that match between the secret bit sequences of different channel pairs

	c_1	c_2	c_3	c_4	c_5
c_1	100.00%	47.92%	49.22%	51.69%	48.96%
c_2	47.92%	100.00%	50.00%	50.65%	47.40%
c_3	49.22%	50.00%	100.00%	49.87%	53.39%
c_4	51.69%	50.65%	49.87%	100.00%	51.43%
c_5	48.96%	47.40%	53.39%	51.43%	100.00%

Table 3.7. Mutual information between secret bits that are extracted from different channels

	c_1	c_2	c_3	c_4	c_5
c_1	1.00	1.20×10^{-3}	1.84×10^{-4}	8.31×10^{-4}	3.20×10^{-4}
c_2	1.20×10^{-3}	0.9994	1.16×10^{-6}	1.17×10^{-4}	1.90×10^{-3}
c_3	1.84×10^{-4}	1.16×10^{-6}	0.9986	3.32×10^{-6}	3.20×10^{-3}
c_4	8.31×10^{-4}	1.17×10^{-4}	3.32×10^{-6}	0.9999	6.05×10^{-4}
c_5	3.20×10^{-4}	1.90×10^{-3}	3.20×10^{-3}	6.05×10^{-4}	0.9994

Table 3.8. NIST - approximate entropy test results

Configuration	Entropy (slow-walk)	Entropy (rotation)
1×1	0.9837	0.9882
2×2	0.9836	0.9819
3×3	0.9787	0.9858
4×4	0.9879	0.9791
5×5	0.9823	0.9832

Table 3.9. NIST statistical test suite results. The P-value from each test is listed below. To pass a test, the P-value for that test must be greater than 0.01.

NIST test	P-value (slow-walk)	P-value (rotation)
Frequency	0.258744	0.217602
Block frequency	0.359077	0.350225
Cumulative sums (Fwd)	0.327781	0.127687
Cumulative sums (Rev)	0.378058	0.401227
Runs	0.317043	0.417624
Longest run of ones	0.157396	0.695291
FFT	0.915226	0.555973
Approximate entropy	0.269068	0.84317
Serial	0.040004, 0.051236	0.494123, 0.134310

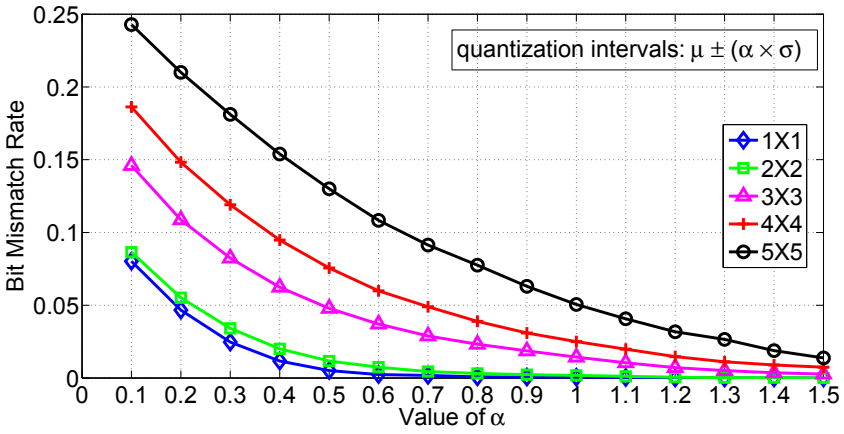

Figure 3.11. Mismatch rate as a function of α and N for slow-walk experiments.

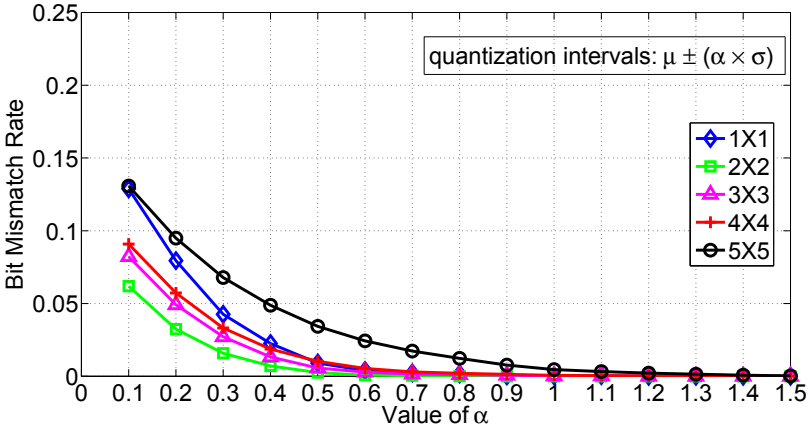

Figure 3.12. Mismatch rate as a function of α and N for rotation experiments.

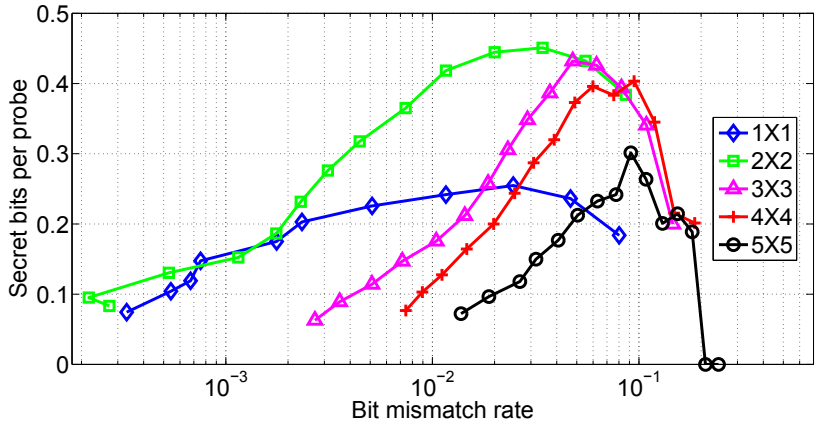

Figure 3.13. Secret bits/probe as a function of mismatch rate and N (slow-walk)

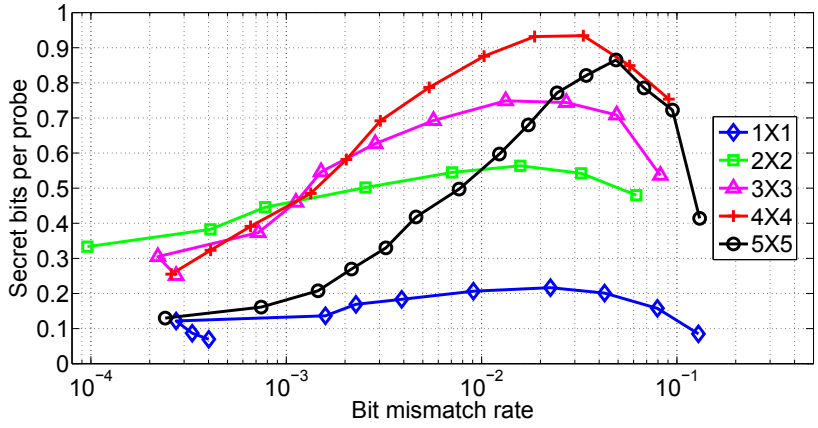

Figure 3.14. Secret bits/probe as a function of mismatch rate and N (rotation)

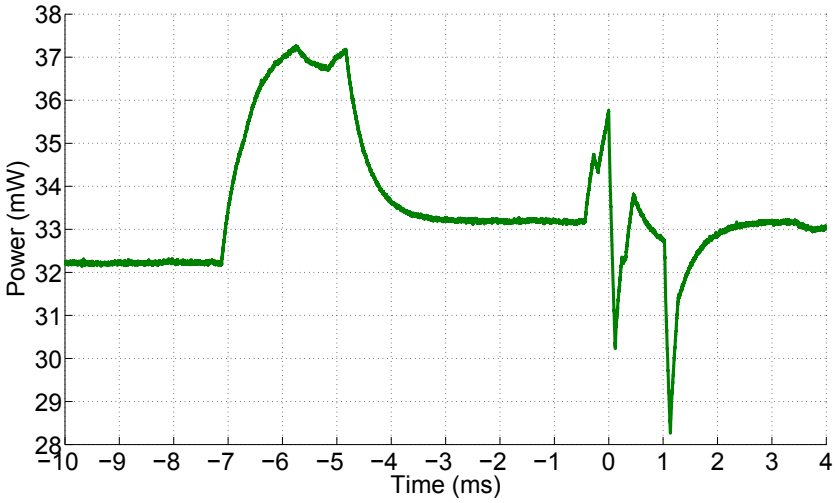

Figure 3.15. Power consumption of a TelosB sensor in the process of transmitting a probe packet.

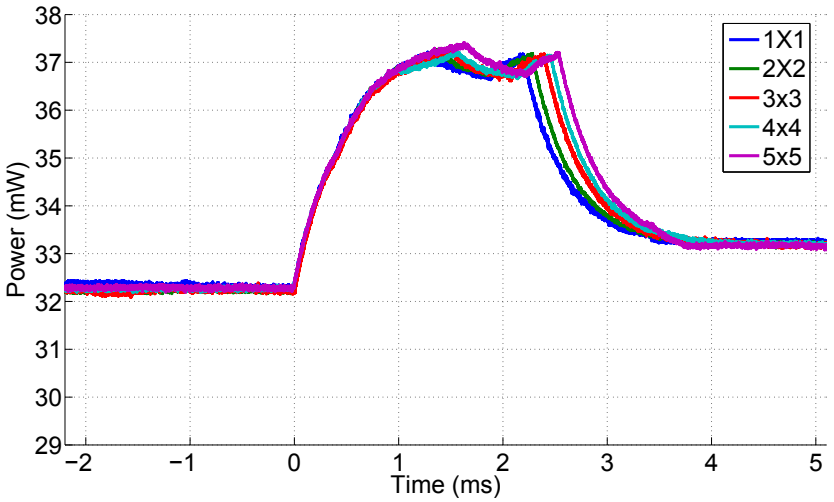

Figure 3.16. Power consumption of a TelosB sensor in copying data from memory to the FIFO buffer on the radio.

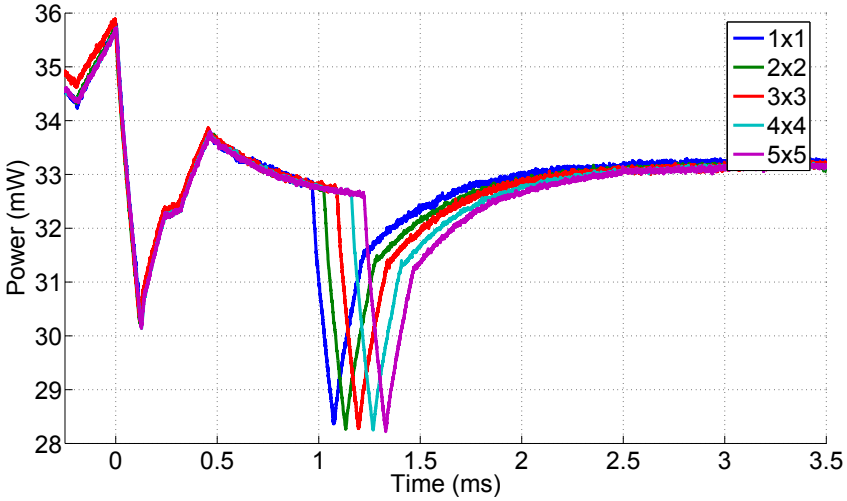

Figure 3.17. Power consumption of a TelosB sensor in actually transmitting a packet.

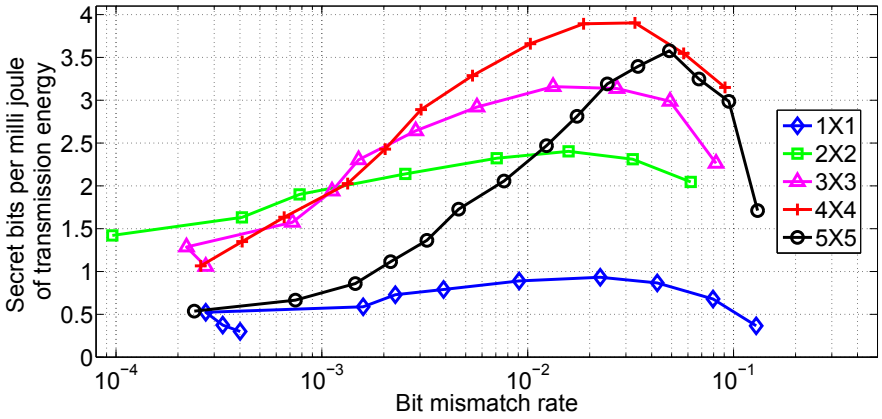

Figure 3.18. Secret bits/mJ of Tx energy vs mismatch rate and N (rotation)

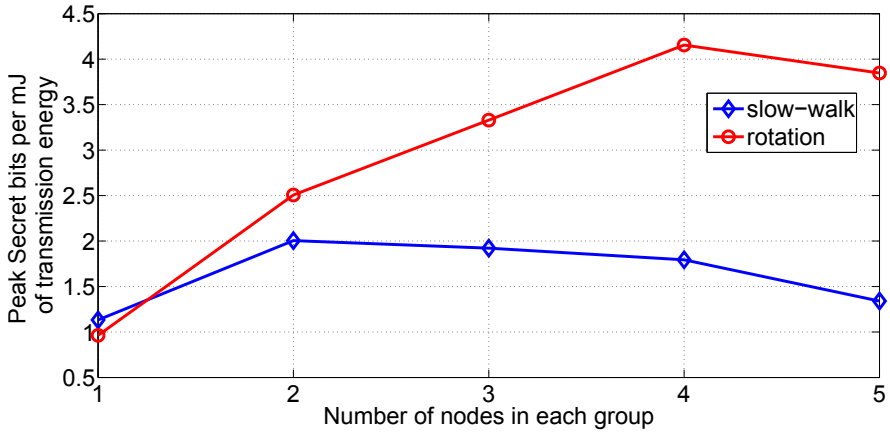

Figure 3.19. Peak secret bits/mJ of Tx energy vs N with 2 byte probe pkts

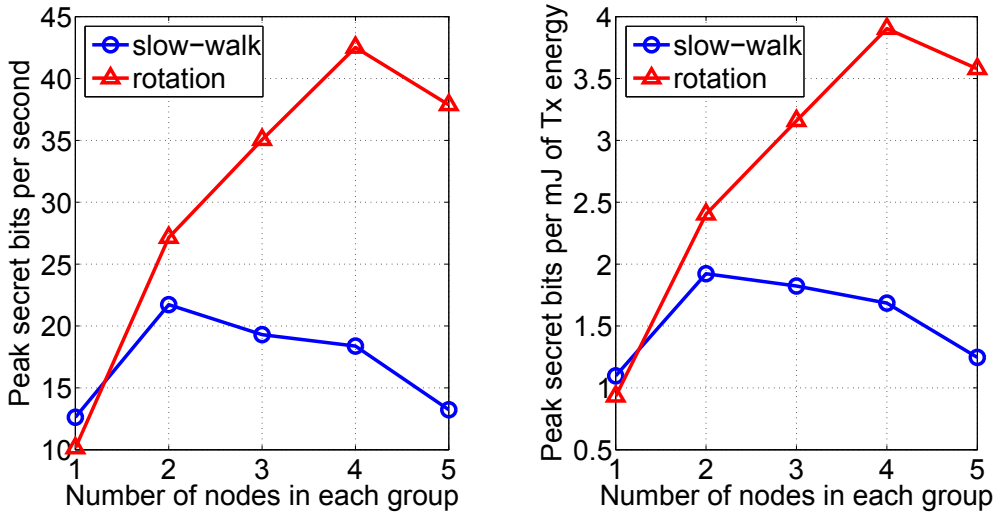

Figure 3.20. Peak secret bit rate as a function of number of nodes in each group.

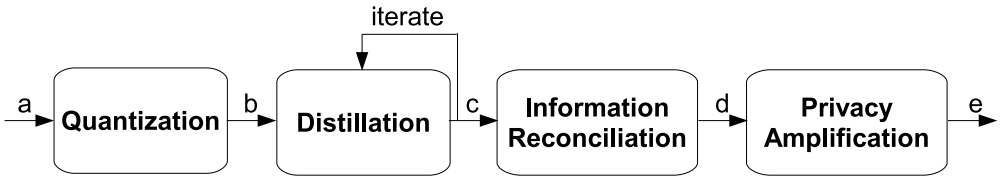

Figure 3.21. Secret Bit Extraction Process. a - RSS measurements, b - quantization interval labels, c - distilled bits, d - reconciled bits, e - secret bits.

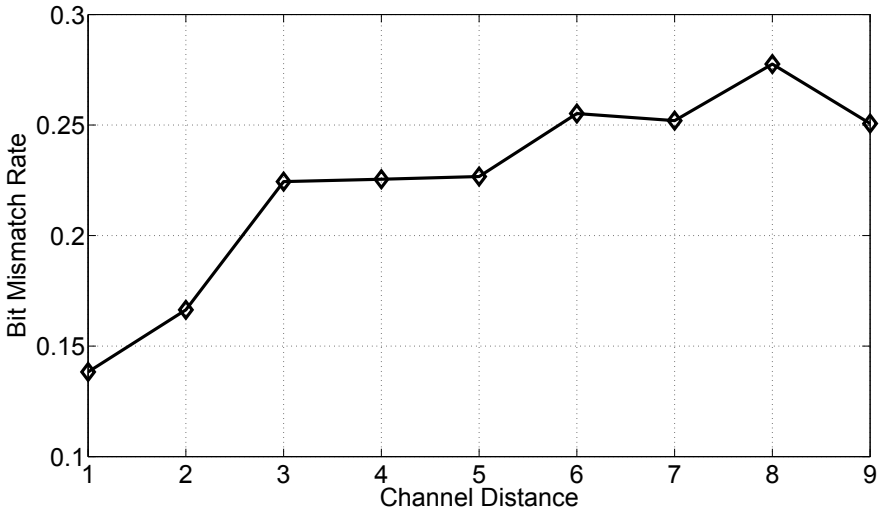

Figure 3.22. Bit mismatch rate vs channel distance

Algorithm 4 Distill Input

```
while there is input do
    if current_label = previous_label then
        Output current_label
    else
        Output exclude_label
        previous_label ← current_label
    end if
end while
```

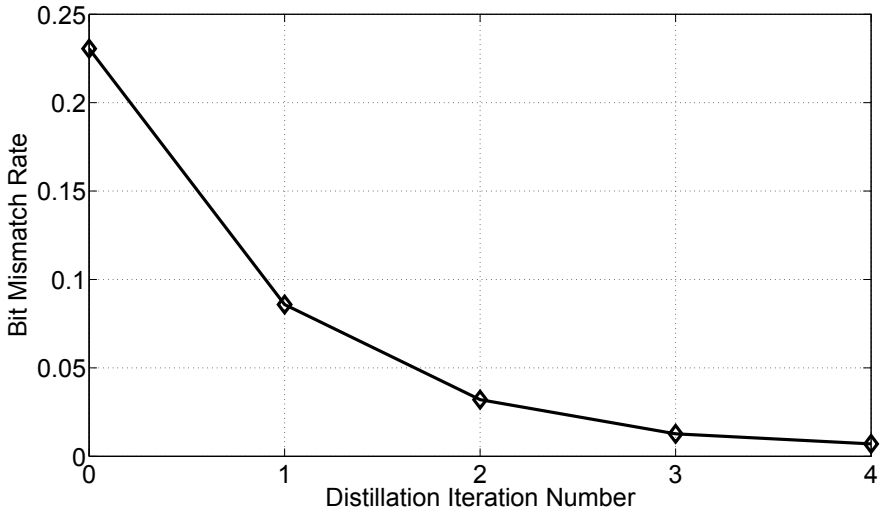

Figure 3.23. Effectiveness of distillation in drastically reducing the bit mismatch rate

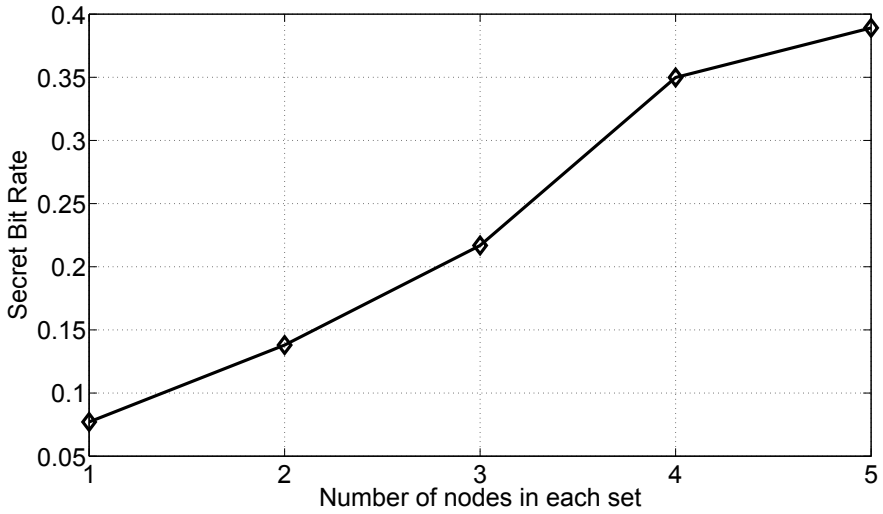

Figure 3.24. Secret bit rate vs number of nodes

Table 3.10. Comparison of hierarchical and simple collaboration ($N = 4$)

Measure	**Hierarchical**	**Simple**
Secret bits per second	47.70	18.37
Secret bits per probe	0.5354	0.4033
Secret bits per mJ of Tx energy	2.283	1.685
Bit mismatch rate	5.475%	9.477%
Approx. differential entropy of RSS measurements	77.10	71.45
Entropy of output secret bits	0.9785	0.9879

CHAPTER 4

BEYOND OFDM: BEST-EFFORT DYNAMIC SPECTRUM ACCESS USING MULTICARRIER FILTERBANK

4.1 Overview

Dynamic spectrum access networks, where multiple nodes compete to utilize a shared frequency spectrum, have been gaining widespread attention in the recent years. However, there are significant challenges in actually building such networks, including limiting the amount of mutual interference among the transmissions from different nodes.

Orthogonal frequency division multiplexing (OFDM), which uses an orthogonal set of subcarriers, has been proposed for the purpose of sharing the different subsets of these subcarriers among nodes [7, 8] interested in dynamic spectrum access. However, OFDM imposes tight timing and frequency synchronization requirements among different nodes, which are very likely to be difficult to achieve in practice, especially when the nodes belong to different administrative entities. Any lack of synchronization can result in significant mutual interference among the signals of different transmitters. A somewhat lesser known and understood multicarrier communication system, filterbank multicarrier (FBMC), first proposed by Saltzberg [10], can overcome the above limitations of OFDM through the use of special transmitter and receiver pulse shaping filters, namely, the square root Nyquist filters. Through the use of these filters, FBMC, in comparison to OFDM, promises a more efficient spectrum utilization by minimizing interference across subcarriers [9]. Essentially,

FBMC reduces the mutual interference between different transmitters to insignificant levels.

While FBMC has been studied at the physical (PHY) layer [57], there is no existing work on understanding its impact on reliable transmission of data packets/frames and also on the data transmission rates. Furthermore, there is no cross layer research on understanding the impact of FBMC beyond the PHY layer, e.g., at the MAC layer. In order to fully understand and evaluate the *promise* of FBMC, in this chapter, we first examine the use of special pulse shaping filters of the FBMC PHY layer in reliably transmitting data packets at a very high rate. For this purpose, we analyze the mutual interference power across subcarriers used by different transmitters. Next, to understand the impact of FBMC beyond the PHY layer, we devise a distributed and adaptive medium access control (MAC) protocol that coordinates data packet traffic among the different nodes in the network in a best-effort manner. In our protocol, when a node senses the current channel to be free, it seeks to increase or reduce the size of the channel (i.e., number of subcarriers) depending on the packet transmission success rate, and on how the current channel access delay compares with the historic average delay. When the current channel is sensed to be busy, the node actively attempts to transmit under smaller nonbusy channels, with fewer subcarriers, to reduce the channel access delay. Our MAC protocol adds or drops subcarriers in an additive increase and multiplicative decrease (AIMD) manner with the aim of achieving fair use of subcarriers across multiple nodes.

We build a discrete event simulator to evaluate FBMC and compare its performance with OFDM. We conduct extensive simulations of both FBMC and OFDM systems that operate in static indoor environments with frequency selective fading channels, and under different data traffic rates with varying number of nodes. Our results show that FBMC consistently achieves *at least an order of magnitude performance improvement* in several aspects including packet transmission delays, channel access delays, and effective data transmission rate available to each node. These improvements can be understood based on our findings that, in comparison to OFDM, FBMC can (a) achieve substantially higher SINR, (b) reliably support modulation schemes with very high data rates for a significant portion of time, and (c) achieve

very low packet error rates.

We also examine the use of FBMC for dynamic spectrum access in a vehicular network setup. Vehicular network communication is gaining importance in recent times as it enables the sharing of emergency information, traffic/road conditions, and weather-related data among different vehicles in real time, and also for delivering advertisements on nearby gas stations, restaurants, etc. [58, 59]. We perform extensive vehicular network simulations over larger, mobile, outdoor settings. For our vehicular network simulations, we generate realistic vehicular mobility traces over the streets of a typical city using VanetMobiSim [60] and the US Census Bureau TIGER GIS database [61]. We find that FBMC achieves *an order of magnitude performance improvement over large distances* in the vehicular networks scenario as well. Further, in the case of multihop vehicular networks, FBMC can achieve about 20× smaller end-to-end data packet delivery delays, and relatively low packet drop probabilities.

In summary, our research shows that a *FBMC-based communication system offers a much higher performing alternative to OFDM* for networks that dynamically share the spectrum among multiple nodes. We expect our research to have a profound impact on the design of future dynamic spectrum access networks.

The rest of this chapter is structured as follows. In Section 4.2, we describe our problem setup. Section 4.3 describes the FBMC PHY layer. In Section 4.4, we compare the PHY layer characteristics of OFDM and FBMC. In Section 4.5, we describe our MAC layer, and discuss how we adapt TCP congestion control principles and apply them to the problem of spectrum sharing. We present our results for the static indoor environment in Section 4.6 and the results from our vehicular network setup in Section 4.7. We discuss the related work in Section 4.8 and summarize our findings in Section 4.9.

4.2 Problem Setup

We assume that there are multiple nodes, possibly belonging to different administrative entities, that compete to utilize the shared frequency spectrum. Nodes belonging to different entities may be associated to different base stations, while still sharing the same spectrum. The spectrum is divided into N subchannels or

subcarriers. In this work, we assume that nodes contend for free spaces in a 20 MHz channel, which is divided into $N = 64$ subcarriers, out of which 53 subcarriers at the center of the channel are used, while the rest of the subcarriers on either end form guard bands, as it is done typically with 802.11a channels. Unlike the 802.11 OFDM model, our dynamic spectrum access system model allows different nodes to interleave their packet transmissions and achieve fair sharing of resources across their heterogeneous traffic demands. Furthermore, within an 802.11 channel, the channel gains for each specific node could be significantly different over different subcarriers due to frequency selective fading. Thus, when each node contends only for a portion of the spectrum that is favorable to that node, it may lead to a more efficient spectral usage (e.g., FARA [62], distributed OFDMA [63]). However, we note that in the case of an *802.11-like* model, OFDM may be preferable due to its lower computational complexity. We assume that a node can simultaneously receive packets from multiple senders when the sets of subcarriers used by those senders are all disjoint. We also assume that each node has multiple radios, in order to allow for full duplex communication (as in standards like 3GPP LTE [64]) and to accurately sense the spectrum for free space. Note that when only one radio is used, a node will incur a short delay of about 5 μs to switch from receive/sense mode to transmit mode before it can transmit a packet [65]. During this short intervening delay, if another node starts to transmit a packet on an overlapping set of subcarriers, it will cause collisions. Use of multiple radios allows nodes to avoid such mode switching delays, and minimizes collisions due to inaccurate channel sensing. Finally, we would like to note that while dynamic spectrum access can induce interference between the transmissions of different nodes at a receiver, interference can also be caused by other sources, for example, microwaves, which have not been considered in our model. In other words, our work is mainly focused on assessing the effect of mutual interference in a dynamic spectrum access scenario, independent of other factors.

4.3 Background on Filterbank Multicarrier Communication

Multicarrier communication systems like FBMC and OFDM simultaneously transmit signals across several subcarriers. In each subcarrier, the information bits are

encoded and transmitted as a series of pulses of different amplitudes and phases. To separate the signals on different subcarriers from one another, a pair of transmit and receive filters corresponding to each subcarrier is used. The pulse shape used by the transmit filter characterizes the power spectral density (PSD) of the signals transmitted on each subcarrier, where the PSD describes the distribution of relative signal power at different points in the spectrum.

FBMC uses special transmitter and receiver filters that are based on the square root Nyquist (SR Nyquist [9]) pulse shapes. These filters can be designed from the transformation of a square root raised cosine pulse to minimize a cost function that strikes a balance between stopband attenuation, the residual intersymbol interference, robust sensitivity to timing jitter, and/or reduced peak-to-average power ratio. Note that by design, SR Nyquist pulse avoids intersymbol interference. Figure 4.1 shows the SR Nyquist pulse shape that is used in FBMC, and as a reference, the rectangular pulse shape of OFDM.

In the frequency domain, both the rectangular and SR Nyquist pulse shapes exhibit spectral components at each point in the spectrum. However, for rectangular pulses, these spectral components have very large magnitudes even very far away from the subcarrier in which the signals are actually transmitted. In contrast, with SR Nyquist pulse, these components have negligible magnitudes beyond just two subcarriers from the subcarrier in which signals are transmitted. In other words, an OFDM signal that is transmitted on an arbitrary subcarrier leaks out a significant amount of power over all the other subcarriers in the channel, whereas an FBMC signal does not leak significant power into the other subcarriers in the channel. OFDM's very high leakage power results in significant mutual interference among the signals of different transmitters, while the relatively very low leakage power in FBMC causes little mutual interference. We provide a mathematical analysis of interference power in Section 4.4.

4.3.1 Complexity

In multicarrier communication systems, signal processing at the physical layer typically involves operations such as discrete Fourier transform, circular convolution, etc. These operations, in turn, involve the use of complex numbers and the computational

complexity is expressed in terms of number of complex number multiplications [66]. In general, the basic structure of an FBMC system is perceived to be more complex than OFDM. OFDM offers lower complexity in comparison to FBMC, in the case of single user communication [1], for example [67]. However, the use of polyphase structures lead to efficient implementations of FBMC systems [68]. For typical choices of systems parameters, FBMC is 20% to 40% more complex than OFDM [57]. Moreover, when OFDM is applied to multiple access applications, it either performs poorly, as we demonstrate in this paper, or one has to add significant complexity to the system to achieve perfect synchronization among different nodes or to apply computationally expensive multiple access interference (MAI) cancellation techniques to improve on its performance [69]. Sourck et al. [66] observe that MAI cancellation-enabled multiple access OFDM systems are generally over an order of magnitude more complex than their FBMC counterparts. Moreover, despite their much higher complexity, such OFDM systems are not able to perfectly remove MAI, while FBMC suppresses MAI almost perfectly.

4.4 FBMC vs OFDM - PHY Layer Characteristics

In this section, we present and compare the PHY layer characteristics of FBMC and OFDM. We compare them in two different aspects - (i) power spectral density (Section 4.4.1), and (ii) interference power at a receiver due to mutual interference across subcarriers used by different transmitters (Section 4.4.2). This comparison provides a PHY layer basis for our ultimate goal of understanding the impact of FBMC on reliable transmission of data packets and on the data transmission rates.

4.4.1 Power Spectral Density

OFDM uses rectangular pulse shapes (Figure 4.1). It is widely used in practice, including in the 802.11 WiFi systems. Figure 4.2 shows the power spectral density of an OFDM signal which is transmitted on subcarrier number 0. The first side lobes are just 13 dB below the main peak, and the side lobes near either end of the channel (subcarrier numbers ± 26) are about 40 dB below the main peak.

[1] We only consider the scenario of multi-user communication in this work.

In 802.11a and other standards, it has often been proposed to replace the rectangular pulse of OFDM (see Figure 4.1) by a raised-cosine pulse. This modified OFDM is called Filtered-OFDM (fOFDM) [7]. The fOFDM exhibits lower side lobes when compared with OFDM. However, for the suggested roll-off factor in 802.11a (2.5%), this improvement is not significant. Figure 4.2 compares the power spectral densities (PSDs) of OFDM and fOFDM. The differences between the two PSDs are exhibited only at far away frequencies - the side lobes near either end of the channel are about 45 dB below the main peak for fOFDM, an improvement of about 5 dB only over OFDM.

The very large side lobes of OFDM/fOFDM can result in significant mutual interference among signals of different subcarriers that are transmitted by different nodes if those signals are not perfectly synchronized both in time and frequency. This synchronization is difficult to achieve in practice, especially if the various nodes belong to different entities.

Recall that FBMC uses SR Nyquist pulse shapes (Figure 4.1). Figure 4.3 shows the power spectral density of an FBMC signal which is transmitted on subcarrier number 0. In a stark/sharp contrast to the spectral components of an OFDM signal (Figure 4.2), all the spectral components outside the main lobe of an FBMC signal are at least 80 dB below the main peak (Figure 4.3). Comparison of the power spectral densities of the OFDM and FBMC signals reveals that FBMC will more effectively contain the signals of different transmitters in their respective subcarrier bands, and thereby minimize the mutual interference among the signals of different transmitters.

4.4.2 Analysis of Interference Power

In this section, we derive an analytical expression for the interference power at a receiver due to mutual interference across subcarriers used by different transmitters. In this analysis, we assume that the wireless channel is a linear time invariant (LTI) filter. Therefore, our analysis in this section applies to static channels, which can be described as LTI filters.

Let $R_n(f)$ denote the frequency response of the receiver filter corresponding to the n^{th} subcarrier, where $-\frac{N}{2} \leq n < \frac{N}{2}$ and N is the number of subcarriers. For OFDM, $R_0(f)$ is the Fourier transform of the *rectangular* function, i.e., the *sinc* function.

For FBMC, $R_0(f)$ is obtained by taking the fast Fourier transform of the SR Nyquist pulse shown in Figure 4.1. $R_n(f)$ is obtained from $R_0(f)$ by appropriately shifting the samples depending on the value of n.

Let $H_{x,y}(f)$ denote the frequency response of the wireless channel between the transmitter (x) and receiver (y). Then, the combined frequency response of the wireless channel filter and the receiver filter,

$$C_{x,y,n}(f) = H_{x,y}(f) \times R_n(f). \tag{4.1}$$

$C_{x,y,n}(f)$ is LTI as both $R_n(f)$ and $H_{x,y}(f)$ are LTI filters.

Let $\varphi_{x\rightarrow *,m\rightarrow *}(f)$ denote the power spectral density of the signal that is transmitted by node x on the m^{th} subcarrier, and let $\varphi_{y\leftarrow x,n\leftarrow m}(f)$ denote the power spectral density of the signal that is received by node y on the n^{th} receiver filter, when the signal is transmitted by node x on the m^{th} subcarrier. Since $C_{x,y,n}(f)$ is LTI, it follows that,

$$\varphi_{y\leftarrow x,n\leftarrow m}(f) = |C_{x,y,n}(f)|^2 \times \varphi_{x\rightarrow *,m\rightarrow *}(f). \tag{4.2}$$

Let $I_{y\leftarrow x}(n,m)$ denote the interference power on filter n of receiver y, when the undesired transmitter x is transmitting on subcarrier number m. Then,

$$I_{y\leftarrow x}(n,m) = \int_{-0.5}^{+0.5} \varphi_{y\leftarrow x,n\leftarrow m}(f)\, df. \tag{4.3}$$

Figure 4.4 shows a plot of $I_{y\leftarrow x}(0,m)$ as a function of m, i.e., interference power on receiver filter 0 as a function of the subcarrier number on which the interferer is transmitting for both fOFDM and FBMC. To better understand the differences between the performance of FBMC and fOFDM, consider the following example cases. In all the cases, let the desired sender transmit its signals on subcarrier number 0.

1. The interferer transmits its signals on subcarrier number 0. The channel losses between the (sender, receiver) and (interferer, receiver) node pairs are equal. Since the interference power equals the signal power (0 dB), the signal-to-interference ratio, $SIR = 0$ dB for both fOFDM and FBMC.

2. The interferer transmits on subcarrier number -5 (or $+5$), and as in the previous case, the channel losses between the (sender, receiver) and (interferer, receiver)

node pairs are equal. In this case, the interference power is about 24 dB below the signal power (SIR = 24 dB) with fOFDM. However, with FBMC, the interference power drops drastically - to about 110 dB below the signal power (SIR = 110 dB).

3. The interferer transmits on subcarrier number -5 (or $+5$). However, assume that the interferer is now closer to the receiver in such a way that the interference power is 21 dB more than that in case 2. In this new scenario, the SIR will be around 3 dB for fOFDM, and around 89 dB for FBMC.

4. The interferer uses two subcarriers -5 and $+5$, and is also closer to the receiver as in the previous case. In this case, the SIR ends up with around 0 dB for fOFDM, while FBMC still has a very high SIR of about 86 dB.

These example cases clearly show that FBMC will be very effective in reducing interference when multiple nodes share the spectrum.

4.4.2.1 SINR on a Subcarrier

Let $S_{y\leftarrow x}(m)$ denote the desired signal power on receiver filter m, when x is the desired transmitter for the receiver y. Then,

$$S_{y\leftarrow x}(m) = \int_{-0.5}^{+0.5} \varphi_{y\leftarrow x, m\leftarrow m}(f)\, df. \tag{4.4}$$

Let $I_y^*(n)$ represent the total interference power at receiver y on the receiver filter n. Then,

$$I_y^*(n) = \sum_u \sum_k I_{y\leftarrow u}(n,k) \tag{4.5}$$

where u belongs to the set of all undesired transmitters for receiver y, and for each u, k belongs to the set of all subcarriers on which u transmits its signals.

Let $SINR_{y\leftarrow x}(n)$ denote the signal-to-interference plus noise ratio on subcarrier n at receiver y, when x is the desired transmitter. Let N_{power} denote the noise power. Then,

$$SINR_{y\leftarrow x}(n) = \frac{S_{y\leftarrow x}(n)}{I_y^*(n) + N_{power}}. \tag{4.6}$$

We use (4.6) for calculating the signal-to-interference and noise ratio in our simulations. The plots shown in Figure 4.4 assume only one interferer which is transmitting

on a single subcarrier. When there are more interfering signals, the total interference power ($I_y^*(n)$) will become larger and this may become very significant for OFDM/fOFDM. In comparison, FBMC will be relatively unaffected by interference from other nodes. Consequently, FBMC will achieve very high SINR values consistently, as we show in our evaluation (Section 4.6). Moreover, FBMC achieves very high SINR even though it uses the same amount of signal power as OFDM/fOFDM.

Such improvements at the PHY layer translate into phenomenal performance gains at the MAC layer, as we show in Section 4.6. Considering that OFDM is very widely used currently and is also proposed for spectrum sharing, our research brings forth the broader impact and the benefits that the next-generation dynamic spectrum access networks will have if they use FBMC instead of OFDM.

4.5 AIMD MAC Protocol

In order to understand the impact of FBMC beyond the PHY layer, we devise a medium access control (MAC) protocol that achieves the following goals. Each node in the network - (i) avails a fair share of the available spectrum, (ii) adapts its own traffic according to the overall traffic in the network, (iii) makes decisions independently without any coordination from a centralized node on the use of different subcarriers. Our MAC protocol is designed/intended for conventional applications that exchange best-effort traffic, where the nodes contend for the subcarriers and backoff if necessary.

AIMD TCP congestion control [70, 71] promises a fair share of the link bandwidth to the different TCP connections sharing a common intermediate link. While the TCP-AIMD and our MAC protocol share similar design goals, the differences lie in (i) the type of resource that is being shared, and (ii) how each node adapts the amount of resource that it uses. In TCP, link bandwidth is the shared resource, whereas in our system, the shared resource is a set of subcarriers available in the spectrum. While the TCP sources adapt their sending rate depending on the level of congestion in the network, the nodes in our system model adapt their channel size (number of subcarriers) using a combination of two different measures, namely, channel access delay and packet error rate, which together better reflect the overall traffic/contention in the network and the error performance of the channel.

We explain the principles behind our AIMD-MAC protocol as follows.

4.5.1 Channel Selection

Prior to its first transmission, a node senses the spectrum to identify the set of subcarriers that are not used for transmissions by other nodes. Then, from this set, it considers the largest block of contiguous subcarriers since it promises for a greater expansion in the future; then, it selects a certain number (α) of subcarriers at the center of this block for its current transmission.

4.5.2 Identifying a *Promising* Channel

We consider a channel to be *promising*, when the following conditions are met – (a) the node perceives a low degree of contention / congestion and (b) packet error rate is below a configurable threshold. Otherwise, the channel is considered as unpromising.

4.5.3 Adapting the Channel Size

When a node determines that a channel (set of subcarriers) is *promising*, it attempts to expand it by additively increasing the number of subcarriers by a total of α subcarriers, where $\alpha/2$ subcarriers are added to either side of the current channel (note that α is a constant number). When a channel is *unpromising*, the node multiplicatively decreases the number of subcarriers, retaining only a β fraction of subcarriers at the center of its current channel.

If a channel is *promising*, but expansion is not possible due to the transmissions of other nodes in nearby subcarriers, the node will use the existing channel. Second, if a channel is *unpromising*, but multiplicative decrease is not possible because the number of subcarriers equals the minimum value, our MAC protocol chooses to either use the existing channel or look for opportunities elsewhere in the spectrum with equal probability.

4.5.4 Detecting and Handling Link-layer Congestion

We use the channel access delay for measuring the level of link layer contention. Channel access delay represents the amount of time that a node has waited to acquire the channel and transmit a given packet. A large channel access delay indicates heavy contention for the wireless channel and vice-versa. Therefore, channel access

delay is a direct measure of link-layer congestion in a wireless network. If there is heavy contention, then it is desirable to reduce the channel size as it will improve the probability of finding an unoccupied channel. Essentially, when multiple nodes compete for a certain number of subcarriers, the likelihood of finding a large number of subcarriers to be unoccupied at the same time is comparatively less than the likelihood of finding a smaller number of subcarriers to be unoccupied.

In our evaluation, each node maintains a running average of the channel access delay per packet which is calculated over the set of packets it has transmitted in the past. This running average characterizes the history. Then, a node perceives heavy contention for the channel when its current channel access delay is greater than the historic average delay and vice-versa. When heavy contention is perceived (i.e., channel is unpromising), it multiplicatively reduces the channel size.

4.5.5 Dealing with High Packet Error Rate

When a number of nodes share the spectrum and transmit their information bits on subcarriers that are closely located in the frequency space, bit errors can occur due to strong mutual interference between their signals. If the transmitters that are involved in causing interference to one another's signals mutually reduce the number of subcarriers in such a way that their subcarriers become less close in the frequency space than they currently are (i.e., their respective subcarrier bands become separated further apart), it will help in reducing interference and thereby reduce packet errors.

To make it concrete, a node multiplicatively reduces its channel size when the packet error rate on the current channel is above a configurable threshold (e.g., 20%). We assume that the receiver feedbacks the transmitter with either the actual packet error rates, or raw SINR values which the transmitter can use to obtain an estimate of the packet error rate; the latter case is similar to the SINR feedbacks in FARA [62].

4.5.6 Backoff for Existing Channel versus Transmit Using a Smaller Channel

Suppose that the current channel is busy, but that there exists a smaller channel that is not busy, and whose set of subcarriers is a proper subset of the set of subcarriers

in the current channel. Consider the following cases. *Case 1:* The sender's waiting time has become inordinately large such that it is greater than the transmission delay of the packet, where waiting time equals the current channel access delay + possible random backoff delay, in the immediate future if the packet will not be transmitted at this instant. *Case 2:* A node has waited for a very long time to transmit a packet because the channel has been found to be busy for a number of attempts, with the channel access delay greater than its historic average. In either case, it is more desirable to transmit immediately using the smaller nonbusy channel, rather than backoff the transmission. In all other cases, it is not unreasonable to backoff, i.e., wait further for the current channel to become free.

4.5.7 Contiguous versus Noncontiguous Access

A transmitter node is required to leave a few guard subcarriers on either side of each block of subcarriers it intends to use for transmissions in order to avoid significant interference from the transmissions of other nodes in the nearby subcarriers. In our simulations, we use two guard subcarriers inbetween any two adjacent blocks of subcarriers that are used by different transmitter nodes. Guard subcarriers represent an overhead - wasted spectrum space that could have otherwise been utilized for useful transmissions. If a node chooses to simultaneously use several noncontiguous blocks of subcarriers, then it increases the ratio of the number of guard subcarriers to the number of subcarriers used for transmission; i.e., it reduces the spectrum efficiency. Therefore, in this work, we allow each node to access only one contiguous set of subcarriers for its transmissions. Contiguous access is also favored in a number of existing works (e.g., [72]).

4.5.8 Isolating the Transmissions of Different Nodes

The transmitter can signal the receiver node about its selection of subcarriers through a dedicated, common control channel, which could be a small subset of the N available subcarriers. However, the use of a dedicated control channel may become a significant overhead when the network traffic grows.

Alternatively, a receiver can infer any new packet transmission on its own, without depending on a control channel, as follows. Since any new transmission produces

a distinguishable *spike* in the power levels over a contiguous group of subcarriers, a receiver can readily identify the starting and ending subcarrier indices that a transmitter uses by observing the first order derivative of the power spectral density measurements [63]. Using these indices, the receiver can isolate the transmissions of different nodes, demodulate the signals, and decode the packets. Then, using the link layer header information (e.g., destination address) in the decoded packet, the receiver can choose to pass a packet to an upper layer depending on whether the packet is intended for the receiver.

4.5.9 How Should AIMD-MAC Behave when there Is a Large Number of Subcarriers?

In our simulations, we consider a channel that is divided into 64 subcarriers, which is a relatively small number. However, in some systems, the number of subcarriers could be very large (e.g., up to 2048 subcarriers in a WiMax channel). In such cases, it will be inefficient to additively increment the number of subcarriers by a small α value at each instant. Therefore, it is more appropriate to either have correspondingly larger α values, or increase the number of subcarriers exponentially until a threshold is reached when using smaller α values, akin to the *Slow-Start* component of the TCP congestion control mechanism.

4.6 Performance in Static, Indoor Settings

In this section, we first describe our simulation environment and the various models that we use in building our discrete event simulator (Section 4.6.1). We evaluate the performance of FBMC and OFDM in indoor, static environments and present our extensive simulation results from Section 4.6.2 to Section 4.6.6.

4.6.1 Components of Our Indoor Network Simulator

While there are a large number of OFDM systems available commercially, currently, there exists no practical system implementation with the FBMC PHY layer. Therefore, in order to compare the performance of FBMC and OFDM, we build a discrete event simulator that enables the execution of various tasks – packet transmission, random MAC backoff, idle wait, etc. – among the various nodes in a chronological order.

When there are a large number of nodes, computing the interference power over several subcarriers and at different points in time over each packet, using Equation 4.3 is extremely time consuming. Hence, to run our simulations more efficiently, whenever we compute interference power using Equation 4.3, we cache it in a table for possible reuse later.

4.6.1.1 Simulation Environment

We use a total of N_{t} nodes in our simulations, where $N_{\mathrm{t}} \in \{10, 20, 30, 40, 50\}$ and there are $N_{\mathrm{t}}/10$ base stations, $N_{\mathrm{t}}/10$ mutually disjoint sets of clients (i.e., 9 clients in each set), where each such set of clients is associated to a different base station. For each PHY layer and for each N_{t}, the nodes exchange a total of $100,000$ packets. All the N_{t} nodes are placed in a square shaped area with a diagonal of 100 feet. The position of each node is uniformly distributed inside the square region. Various nodes contend for the 53 subcarriers over a 20 MHz channel in the range 5000 MHz - 5020 MHz.

4.6.1.2 Channel Model

We model the multipath, frequency selective fading wireless channel for an indoor static environment following Hashemi et al. [73]. We determine path loss for our environment using the Keenan-Motley partition loss model [74, 75, 76]. The noise power is calculated as, $N_{power} = NF \times k \times T \times B$, where NF is the noise figure, k is the Boltzmann's constant, T is the ambient temperature ($= 290K$), and B is the bandwidth ($= 20\ MHz$). As defined in the IEEE 802.11a standard, we choose noise figure, $NF = 10$ dB and transmit power of 200 mW [74].

4.6.1.3 Packet Error Rate Model

Awoniyi et al. [74] express packet error rate as a function of the channel gain, SNR, data rate, and packet size. In their model, they assume a single user case, where the entire channel is occupied by one user at a time. We generalize Awoniyi's method to apply it to the scenario where many nodes simultaneously occupy different subsets of subcarriers.

Notice that the interference power on each subcarrier at a receiver could vary

with time, within the course of a single packet transmission, since other nodes may arbitrarily start (or finish) new (or existing) packet transmissions at any instant. Further, the received signal power could also vary with each subcarrier due to frequency selective fading. Thus, the SINR, and hence bit error probability (b.e.p.), could vary with both time and frequency. We appropriately average these probabilities to obtain a more accurate estimate of the overall b.e.p. over the entire packet, using which we compute the packet error rate as outlined in Awoniyi et al. [74].

4.6.1.4 Rate Adaptation

In our model, similar to existing schemes (e.g., [62]), we assume that for each received packet, the receiver feedbacks the transmitter with an SINR value, which represents an average across all the subcarriers used for transmission over its life time. Using this feedback, the transmitter selects the appropriate modulation scheme to use for the subsequent packet transmission.

4.6.1.5 Packet Traffic Modeling

We model packet traffic using the observations of Yeo et al. [77]. In our model, packet lengths follow an exponential distribution with mean/minimum/maximum sizes being 100/64/1500 bytes, respectively. The mean packet interarrival duration is 10 ms for clients and $(10/N_{\mathrm{c}})$ ms for base stations, where the number of clients per base station, $N_{\mathrm{c}} = 9$ in our simulations.

4.6.2 Comparison of SINR and Modulation Scheme Selection

Figure 4.5 shows a sample variation in the SINR over 100 consecutive packets with $N_{\mathrm{t}} = 30$ nodes. The median SINR lies in the range of $11 - 12$ dB for both OFDM and fOFDM. In a sharp contrast, the median SINR for FBMC is around 30 dB, i.e., FBMC consistently achieves substantially higher SINR in comparison to both OFDM and fOFDM.

With very high SINR, FBMC can reliably support modulation schemes with very high data rates. Figure 4.6 shows the distribution of each modulation scheme. About 87% of the time, FBMC is able to support 256-QAM, whereas OFDM/fOFDM can support it only for about $14\% - 15\%$ of the time. Further, OFDM/fOFDM selects

BPSK, which has the lowest data rate, for about 40% of the time. Moreover, the usage distribution of the different modulation schemes generally fall with increase in the data rate supported by the modulation scheme for OFDM/fOFDM, and which is completely opposite to the distribution of modulation schemes for FBMC. Note that a single 256-QAM symbol encodes 8 bits of information, whereas a BPSK symbol encodes only 1 bit of information. Therefore, using FBMC should allow a node to transmit bits at a much faster rate in comparison to both OFDM and fOFDM.

4.6.3 Comparison of Packet Error Rate

Figure 4.7 shows a comparison of packet error rates with varying number of nodes. FBMC has negligible packet error rates ($\leq 0.113\%$) regardless of the number of nodes sharing the spectrum. The performance of OFDM/fOFDM compares with FBMC only in the case of 10 nodes, as in this case, the amount of traffic is relatively low and as a consequence, the effect of mutual interference among different nodes is minimum even with OFDM. In all the other cases with more nodes, the packet error rate performance of OFDM/fOFDM is dramatically different from that of FBMC, where about 30% packets are in error. This clearly shows the effectiveness of FBMC receivers at filtering out transmissions of undesired senders on all subcarriers in which a receiver is not interested.

4.6.4 Comparison of Transmission and Channel Access Delays

Transmission delay (t_d) depends on the packet size (b), number of subcarriers (s) used for transmission, number of bits encoded in each symbol ($\log_2(M)$), which depends on the modulation scheme, and the coding rate. For all modulation schemes, we select a coding rate of (1/2) using convolutional coding. In $802.11a/g$, when 48 subcarriers are used for data transmission, a data rate of 6 $Mbps$ is achieved when using BPSK and a coding rate of (1/2). Using this fact, the transmission delay (t_d) for our system model is expressed as,

$$t_d = \left(\frac{48}{s}\right) \times b \times \frac{1}{(6 \times \log_2(M))}\ \mu s. \tag{4.7}$$

Figure 4.8 shows a plot of average transmission delay per packet as a function of number of nodes in the network. With 20 or more nodes, FBMC achieves an order of

magnitude reduction in the average transmission delay per packet over both OFDM and fOFDM. Figure 4.9 shows a plot of average channel access delay per packet as a function of number of nodes. Again, with 20 or more nodes, FBMC achieves more than an order of magnitude reduction in the average channel access delay per packet. As an example, with 50 nodes, using OFDM/fOFDM requires a node to wait, on average, more than 10 ms in order to acquire the channel and transmit a packet. In a sharp contrast, under the same setting, FBMC requires a node to wait, on an average, less than 1 ms. Therefore, FBMC will enable upper layer applications, particularly real-time applications, to exchange data with very short delays.

4.6.5 Effective Data Rate Available per Node

To understand the differences in the *rate* at which a node can transmit data bits *successfully*, we define effective data rate available per node as follows.

$$Data\ rate_{eff} = \frac{\sum_{p_s} packet\ length}{\sum_{p}(channel\ access + tx\ delay)} \tag{4.8}$$

where p belongs to the set of *all* packets, and p_s belongs to the set of all packets that are received *successfully* (no packet error). Therefore, the effective data rate measure also accounts for the wasted bandwidth due to those packets received in error. Figure 4.10 shows a plot of effective data rate available per node versus the number of nodes. FBMC consistently outperforms OFDM/fOFDM with more than an order of magnitude difference, when there are 20 or more nodes, in the number of bits transmitted *successfully* per second (which is similar to the measure, *goodput*).

4.6.6 Performance Variation with AIMD-MAC Parameter

In the results we have seen thus far, we have chosen the AIMD-MAC parameter, $\alpha = 2$, and $\beta = (1/2)$; i.e., the number of subcarriers is additively incremented by two (for promising channel), and is halved (for unpromising channel). Figure 4.11 shows how the effective data transmission rate for FBMC varies with α, when $\beta = 1/\alpha$ for the scenario with $N_{\mathrm{t}} = 30$ nodes. While large α (i.e., smaller β) values additively increment the channel size by a large value when it is promising, multiplicative decrease due to unpromising channel causes it to shrink in size more aggressively.

Further, smaller α values yield better performance because it increases the chance of many nodes to share the spectrum.

4.7 Performance in Outdoor, Vehicular Network Settings

In this section, we evaluate the performance of FBMC and fOFDM under outdoor, block fading channels in a vehicular network setup. We describe the components of our simulator and compare the performance of FBMC and fOFDM in a single-hop vehicular network in Section 4.7.1 and Section 4.7.2, respectively. We analyze the multi-hop vehicular network performance in Section 4.7.3. We summarize the main results of our FBMC-based approach for dynamic spectrum access in vehicular networks in Section 4.7.4.

4.7.1 Components of Our Single-Hop Vehicular Network Simulator

4.7.1.1 Outdoor Environment

We consider an area of $A \times A$ square meters, where $A \in \{100, 200\}$, whose approximate central latitude and longitude in decimal degrees notation are 40.764670° and -111.870781°, respectively. It corresponds to the intersection of 700 East and 200 South streets in Salt Lake City. We obtain the road maps for our simulation from the US Census Bureau TIGER GIS database [61]. Each road-side access point is located on one of the four corners of the road intersection, and when $M = 50$, we place the fifth access point on the median of the 700 East street where it meets the northern edge of the intersection.

4.7.1.2 Vehicular Mobility

VanetMobiSim [60] can generate mobility traces over the actual streets of any US city, or a section of the city. Using the TIGER GIS maps as input to VanetMobiSim, we generate realistic vehicular movement patterns, where vehicles - slow down and stop at intersections, follow the posted speed limits, regulate their speed based on other vehicles on the front, accelerate/decelerate obeying the laws of physics, etc.

4.7.1.3 Channel Model

We model the multipath, frequency selective fading wireless channel for an outdoor environment following Rappaport et al. [78]. We assume a block fading channel, in which the wireless channel undergoes changes with each packet transmission. Specifically, whenever any transmitter node, t, where $(1 \leq t \leq M)$, starts a new packet transmission, we generate new fading channels for all node pairs in the set $\{(t, r) \mid r \neq t, 1 \leq r \leq M\}$.

We determine path loss using the model of Cheng et al. [79], which is based on real-world vehicular measurements. As defined in the IEEE 802.11p standard, we use a max. transmit power of $760mW$. Different nodes in the network contend for the 53 subcarriers over a 20 MHz channel in the 5.9 GHz frequency range.

For modeling packet error rate, rate adaptation, and packet traffic, we use the models that we describe in Section 4.6.1.

4.7.2 Single-Hop Network Performance

We compare the performance of FBMC and fOFDM as a function of distance using a number of metrics in Section 4.7.2.1. We examine the impact of blasting at full transmit power versus the use of power control in Section 4.7.2.2. Finally, we investigate the impact of the size of the simulation region in Section 4.7.2.3. We use $M = 40$ nodes in our evaluation and these M nodes exchange a total of $500,000$ packets.

4.7.2.1 Performance Variation w.r.t. Distance

A large outdoor area that we use in our evaluation allows us to compare the variation in performance of FBMC and fOFDM as a function of the distance between the transmitter and receiver nodes. In this subsection, we assume that the simulation region has an area of $A^2 = 100^2 m^2$.

4.7.2.1.1 Comparison of SINR. We make the following interesting observations in Figure 4.12: (i) the median SINR of fOFDM, even at the smallest distance ($\approx$ 19 dB), is still much less than the SINR of FBMC at the farthest distance ($\approx$ 26 dB), (ii) the median SINR achieved at the farthest distance with FBMC is high enough to reliably support 256-QAM, the highest data rate modulation scheme we

use in this work, whereas fOFDM cannot support it even at the smallest distance, and (iii) for any given distance, FBMC consistently achieves almost 19 dB greater SINR than that of fOFDM. Therefore, *FBMC can help in building very high speed vehicular networks covering large distances.*

4.7.2.1.2 Comparison of modulation schemes. With very high SINR, FBMC can reliably support modulation schemes with very high data rates. Figure 4.13 shows the distribution of each modulation scheme. About 89% of the time, FBMC is able to support 256-QAM, whereas fOFDM can support it only for about 5% of the time. In addition, fOFDM selects considerably lower data rate modulation schemes for a significant portion of time. Note that a single 256-QAM symbol encodes 8 bits of information, whereas each symbol in BPSK, a lower data rate modulation scheme, encodes only 1 bit of information. Therefore, using FBMC should allow a node to transmit bits at a much faster rate in comparison to fOFDM.

4.7.2.1.3 Comparison of transmission delay. At larger distances, fOFDM is forced to select lower data rate modulation schemes due to low SINR. Therefore, the packet transmission delay increases significantly with distance for fOFDM, as we see in Figure 4.14. On the other hand, since the median SINR for FBMC is high enough to support 256-QAM at all distances, FBMC can achieve almost no increase in the average transmission delay even at farther distances; this is also evidenced from the fact that close to 90% of the time, FBMC supports 256-QAM (Figure 4.6).

4.7.2.1.4 Comparison of packet error rates. Figure 4.15 shows a comparison of the packet error rates as a function of the distance. For a given modulation scheme, the margin between the minimum required SINR and the achieved median SINR decreases with increase in distance; therefore, signals from undesired transmitters can more easily cause significant mutual interference as the distance between the desired transmitter and receiver grows. This effect causes the increase in packet errors with distance for both fOFDM and FBMC. However, at the shortest distance, the packet error rate of FBMC is about 56× smaller than that of fOFDM. Further, even though the packet error rate of FBMC appears to increase more significantly with distance, although on a much smaller scale in comparison to fOFD-M, at the farthest distance, the packet error rate of FBMC is still about 2.8× smaller than that

of fOFDM. This shows that the packer error performance of FBMC is significantly better than fOFDM for short range communication in outdoor settings.

4.7.2.1.5 Comparison of effective data rate. Our effective data rate measure captures the number of bits successfully transmitted per second; i.e., it considers the overhead due to channel access delays and wastage in bandwidth due to packet errors. As we see in Figure 4.16, the effective data transmission rate decreases slowly with increase in distance for both FBMC and fOFDM. While we can clearly see the differences in their performance across all distances, the performance gap increases with distance - at the smallest distance, FBMC achieves $\sim 11.7\times$ the performance of fOFDM, while at the farthest distance, FBMC achieves $\sim 15.5\times$ the performance of fOFDM.

4.7.2.2 Blasting at Full Transmit Power versus Using Power Control

In this work, we also consider the use of transmit power control at each node. When using power control, nodes adjust their transmit power to compensate for the path loss which increases with distance. In other words, for all pairs of communicating transmitter and receiver nodes, all the receiver nodes receive an equal amount of signal power from their corresponding desired transmitter. The power control is subject to the maximum transmit power constraint of 760 mW, which is set by the FCC; i.e., nodes that are separated by the largest distance use a transmit power of 760 mW, while those that are separated by smaller distances use correspondingly lower transmit power.

Table 4.1 and Table 4.2 show a comparison of blasting at full transmit power vs the use of power control. For the data in Table 4.1 and Table 4.2, we use a simulation region whose area is $A^2 = 100^2 m^2$. While the desired signal power that reaches an intended receiver node (r) is controlled, the mutual interference power due to an unintended transmitter that reaches the node r is not controlled, which is primarily determined by the proximity of the undesired / interfering transmitter to its intended receiver as well as the node r. The performance of fOFDM is largely determined by the effects of mutual interference whether the transmitter nodes blast at full power or use power control, as we see in Table 4.1 that there is little difference

in their performance. With FBMC on the other hand, when all nodes blast at full transmit power, an FBMC receiver that is closer to its desired transmitter can achieve greater SINR since it can effectively filter out the undesired mutual interference power. Therefore, FBMC achieves better performance when blasting at full power and it undergoes a slight degradation in performance with the use of power control. It may thus appear that the use of power control presents a case that is more advantageous for fOFDM. However, FBMC produces significant improvement in performance over fOFDM irrespective of whether the nodes blast at full power (an order of magnitude improvement) or use power control (about 8× improvement). Note that except for the power control results that we show in Table 4.1 and Table 4.2, all the other results in this paper assume that nodes blast at full transmit power since it yields the best performance.

4.7.2.3 Impact of the Size of the Simulation Region

Table 4.3 and Table 4.4 compare the difference in performance due to an increase in the size of the simulation area. For both fOFDM and FBMC, the increase in the size of the simulation area does not very significantly reduce the performance. This is due to the following reason.

The increase in the size of the simulation area from $100m \times 100m$ to $200m \times 200m$ causes the farthest distance between an interferer and a receiver to double, i.e., from $100m$ to $200m$. However, for both $100m \times 100m$ and $200m \times 200m$ cases, as the access points are located on the intersection of two streets, which is near the center of the simulation area, the farthest distance between a desired transmitter and a receiver is increased from about $70m$ to $119m$ only, i.e., the distance between the desired transmitter and receiver is not doubled. As a result, the interfering signals experience greater path loss than the desired signal, on average. Therefore, despite the increase in the size of the simulation area, there is not a significant decrease in performance for both fOFDM and FBMC PHY layers.

4.7.3 Multihop Network Performance

We assume that the multihop network is realized with the help of moving cars and/or road-side relays acting as hops on a multi-hop path. In this subsection, first

we analyze the packet delivery performance for the real-time and broadcast packet traffic that do not use ACKs in Section 4.7.3.1. Then, in Section 4.7.3.2 we compare the end-to-end packet delivery delays when ACKs are used.

4.7.3.1 Comparison of Packet Delivery Probability

In this subsubsection, we consider the following types of traffic that do not specifically depend on link layer ACKs: (i) unicast packet transmissions from delay-sensitive / real-time applications, (ii) broadcast packet transmissions, which are very crucial to the operation of an ad hoc network; for example, reactive routing protocols like AODV [80], LAR [81], etc. depend on route request packet broadcasts to establish an end-to-end route between any pair of source and destination nodes.

Let p_{e} denote the average packet error rate across a wireless link. Let p_{d} denote the probability that a packet gets dropped while crossing hop number, h. Then, $p_{\mathrm{d}} = 1 - (1 - p_{\mathrm{e}})^h$.

Our expression for p_{d} assumes independence of packet transmission errors across each hop. We argue that this is a reasonable assumption for the following reasons: (i) each receiver node has a different spatial relationship (i.e., varying distances) with other transmitters in its vicinity; in other words, the sets of interferers, which can cause strong mutual interference (and consequently bit errors), to each receiver node in a multi-hop path are likely to be different from one another, (ii) further, the same set of interfering nodes are not likely to be actively transmitting when the packet in question is being transmitted across different hops, because upon receiving a packet, a node has to first wait to acquire the channel before it can transmit it across the successive hop, and (iii) finally, we also note that in a dynamic spectrum access setup, transmissions across each hop can be carried out using different sets of subcarriers, which also minimizes the hidden terminal problem. For example, in a chain of multi-hop nodes, $A \rightarrow B \rightarrow C \rightarrow D$, where the A and C are hidden from one another, the potential collisions that can occur at node B due to simultaneous transmissions $A \rightarrow B$, and $B \rightarrow C$ or $C \rightarrow D$ is minimized with the use of a different set of subcarriers for these transmissions.

Figure 4.17 shows the packet drop probability as a function of the number of hops. For the purpose of plotting this figure, we assume a simulation region whose

area is $100^2 m^2$ and that these nodes blast at full transmit power. The packet drop probability rapidly approaches the value of one for fOFDM in comparison to FBMC. For example, even with just 5 hops, the packet drop probability is greater than 75% for fOFDM, whereas it is less than 20% for FBMC. For fOFDM, when route request packets are dropped with such high probabilities, it will be extremely difficult even to establish a useful end-to-end route between any pair of source and destination nodes that are separated by a large enough number of hops.

Assuming that the end-to-end routes are somehow established through some out-of-band means for fOFDM, to ensure an acceptable level of performance, reliability mechanisms like error correction, ACKs, and retransmissions, etc. should be implemented on a hop-by-hop basis for fOFDM, even for the exchange of real-time unicast traffic; however, due to very high packet error rates with fOFDM, using ACKs and frequent retransmissions will also significantly increase the end-to-end packet delivery delays. However, FBMC on the other hand can afford to use such reliability mechanisms on an end-to-end basis while still achieving a good performance.

4.7.3.2 Comparison of End-to-End Packet Delivery Delay

Assume that for reliable packet transmissions, ACKs are used. A packet transmission across a hop is successful if only and if it reaches the receiver node without any errors, and the original sender node correspondingly receives an ACK for the packet, i.e., there is a *cycle* of packet transmissions (actual packet + ACK packet) across each hop. Let p_{e} denote the average packet error rate. Then, the expectation of the number of *cycles* to complete a successful packet transmission across one hop is given by, $n_{\mathrm{c}} = \frac{1}{(1-p_{\mathrm{e}})^2}$.

Let d_{ca} and d_{tx} denote the average channel access and packet transmission delays, respectively. Since there are two packet transmissions in each cycle, the average delay per hop is given by, $d_{\mathrm{h}} = (d_{\mathrm{ca}} + d_{\mathrm{tx}}) \times 2 \times n_{\mathrm{c}}$.

Let $dist$ denote the distance between source and destination nodes. Let $dist_{\mathrm{hop}}$ denote the average distance between two successive hops. Then, the average end-to-end delay is expressed as $d_{\mathrm{ee}} = \lceil \frac{dist}{dist_{\mathrm{hop}}} \rceil \times d_{\mathrm{h}}$.

Figure 4.18 shows the average end-to-end delay as a function of distance between the source and destination nodes. For the purpose of plotting this figure, we assume

a simulation region whose area is $100^2 m^2$ and that these nodes blast at full transmit power. In this figure, the small and large hops correspond to $dist_{\text{hop}} = 35m$ and $dist_{\text{hop}} = 66m$, respectively. For both PHY layers, larger hop distance produces smaller end-to-end delays and vice-versa. More importantly, the FBMC PHY layer yields more than an order of magnitude lower delays in comparison to fOFDM. For example, at $500m$, FBMC achieves about $20\times$ smaller average end-to-end delays than fOFDM - 19 ms vs 389 ms. Thus, the performance gap between fOFDM and FBMC grows further with distance in a multihop network.

4.7.4 Discussion

We summarize and discuss some of the main results of our FBMC-based approach for dynamic spectrum access in vehicular networks as follows.

1. For any given distance, FBMC consistently achieves about 19 dB greater SINR than that of fOFDM. Therefore, FBMC can help in building very high speed vehicular networks covering large distances.

2. Packet error rate performance of FBMC is significantly better than fOFDM for short range communication in outdoor settings given that FBMC can achieve $2.8\times$ to $56\times$ smaller packet error rates in comparison to fOFDM.

3. Due to its ability to successfully transmit $11.7\times$ to $15.5\times$ more bits per second than fOFDM, FBMC can enable the deployment of new applications over vehicular networks that require high data rate (e.g., real-time high definition video streaming).

4. Due to very high packet drop probabilities (e.g., $> 75\%$ with just 5 hops) with fOFDM, it will require reliability mechanisms like error correction, ACKs, retransmissions, etc. to be implemented on a hop-by-hop basis to achieve any reasonably acceptable performance. FBMC, on the other hand, has very low packet drop probabilities (e.g., $< 20\%$ with 5 hops) and hence, it can afford to use such reliability mechanisms on an end-to-end basis to achieve good performance.

5. Finally, in the case of multihop networks, any interactive voice-based communications (e.g., voice-over-IP) will be very difficult with fOFDM, since its end-to-end packet delivery delays are prohibitively high, close to 400 ms, when the communicating parties are separated by 500 m. For the same case, FBMC achieves less than 20 ms delays, which is low enough to easily support real-time voice communications.

These results can serve as guidelines for designing ad hoc, dynamic spectrum access communication standards for vehicular networks.

4.8 Related Work

Some existing work [9, 82] has investigated the effectiveness of SR Nyquist filters from a PHY layer perspective - they have established that SR Nyquist filters exhibit better magnitude responses in comparison to other types of filters, that FBMC has a higher bandwidth efficiency because there is no cyclic prefix, and that FBMC is also very efficient at sensing the channel for free spaces in a cognitive radio setting. Our work is primarily focused on showing the benefits of SR Nyquist pulse from a cross layer perspective - in terms of reliable and efficient packet transmissions. Our analysis on the mutual interference power due to the transmissions of different nodes provides a PHY layer basis for understanding impact of FBMC at the MAC layer. In essence, we provide a very thorough comparison using a very comprehensive set of measures/characteristics that include interference power, SINR, modulation schemes, packet error rates, transmission and channel access delays, and effective data transmission rate available to each node at the MAC layer to show that FBMC is the best choice for dynamic spectrum access networks.

Distributed OFDMA [63] has been proposed for dynamic spectrum access for applications with predictable traffic demands (e.g., high definition media streaming); therefore, it obviates the use of contention-based CSMA and hence avoids incurring any unpredictable medium access delays. Our work, on the other hand, is focused on exchanging more conventional / best-effort traffic, where nodes contend for the subcarriers and backoff if necessary; hence, our approach complements [16]. Moreover, in contrast to [63], we consider nodes with full duplex communication capabilities

operating on frequency selective fading channels, and which can perform rate adaptation by selecting different modulation schemes. We also show that FBMC can reliably support high data rate modulation schemes like 256-QAM for a very large portion of time.

In fine-grained channel access (FICA) [65], the OFDM PHY layer is used for sharing the subchannels with different nodes. Our work differs from FICA in the following significant ways. First, FICA uses *half-duplex radios*; hence, it controls the number of subchannels that a node can use based on the observed *collision level* using an AIMD scheme; but in our system model, the use of *multiple radios* minimizes collisions and therefore, we increase/decrease the number of subcarriers that a node can use through a combination of *channel access delay* and *packet error rate* measures, which together better reflect the overall traffic/contention in the network as well as the error performance of the channel. Second, unlike FICA's channel allocation strategy, our MAC protocol is completely *decentralized* and we do not require any RTS/CTS mechanisms for channel reservations or any dedicated access point for arbitrations. Third, unlike FICA, FBMC does not require all nodes to time-synchronize their packet transmissions; as a consequence, nodes using FBMC PHY (i) can transmit *variable sized packets*, which also avoids additional packet fragmentation and reassembly overheads; (ii) can use conventional random backoff mechanisms; and (iii) can share the spectrum with other colocated nodes that use any different PHY layer. FBMC can achieve these without negatively impacting the performance since it reduces mutual interference to insignificant levels.

FARA [62] implements downlink OFDMA that has only one transmitter, the access point, and hence, synchronization problems do not exist. FARA uses a very wide 100 MHz channel, where the gains could vary significantly between different subcarriers. Hence, a FARA transmitter may select a different modulation scheme for each subcarrier. Since each node in our system uses only a small number of subcarriers from a 20 MHz channel, a transmitter adopts a simpler approach of using the same modulation scheme on all subcarriers.

Back2F [83] considers a very interesting approach to MAC backoffs in the frequency domain, as an alternative to conventional time domain-based backoff mech-

anisms, for OFDM-based WiFi networks. We note that dynamic spectrum access networks can also benefit from a Back2F-like approach when nodes contending for an overlapping set of subcarriers perform frequency domain backoff on those subcarriers. However, any approach based on OFDM, including Back2F, requires synchronization between the transmissions of different nodes, which can be avoided with the use of FBMC.

Channel width [84] and WhiteFi [72] adapt the width of an OFDM channel (e.g., 5, 10, or 20 MHz) that a node uses to increase throughput / range. With a fixed number of subcarriers (equal to 64), wider channels yield higher throughput due to smaller symbol durations. On the other hand, narrower channels achieve higher SNR, and hence larger range, due to the fact that the SNR increases when the transmit power per Hz becomes proportionally higher and that the noise power becomes proportionally lower while reducing channel width. Narrow channels are also more resilient to delays spreads because of larger symbol durations; while it results in reducing packet loss/error rates, it correspondingly lowers the data rate. In our work, we show that the use of FBMC in vehicular network setup can simultaneously achieve far larger range and with at least an order of magnitude improvement over OFDM.

4.9 Conclusion

We examined the use of FBMC for best-effort dynamic spectrum access networks. We analyzed the mutual interference power across subcarriers used by different transmitters. We devised a distributed and adaptive MAC protocol that coordinates data packet traffic among the different nodes in the network. We showed that FBMC consistently outperforms OFDM with *an order of magnitude performance improvement* in terms of packet transmission delays, channel access delays, and effective data transmission rate available at the MAC layer in static, indoor environments. We examined the use of FBMC for dynamic spectrum access in a vehicular network setup as well. Through extensive simulations, we showed that FBMC outperforms OFDM with *an order of magnitude improvement over large distances* in vehicular networks. Finally, we also showed that in the case of multihop vehicular networks, FBMC can

achieve about 20× smaller end-to-end packet delivery delays and relatively low packet drop probabilities in comparison to OFDM. These results can serve as guidelines for designing ad hoc, dynamic spectrum access communication standards for future vehicular networks. We have presented these results in three papers [85, 86, 87].

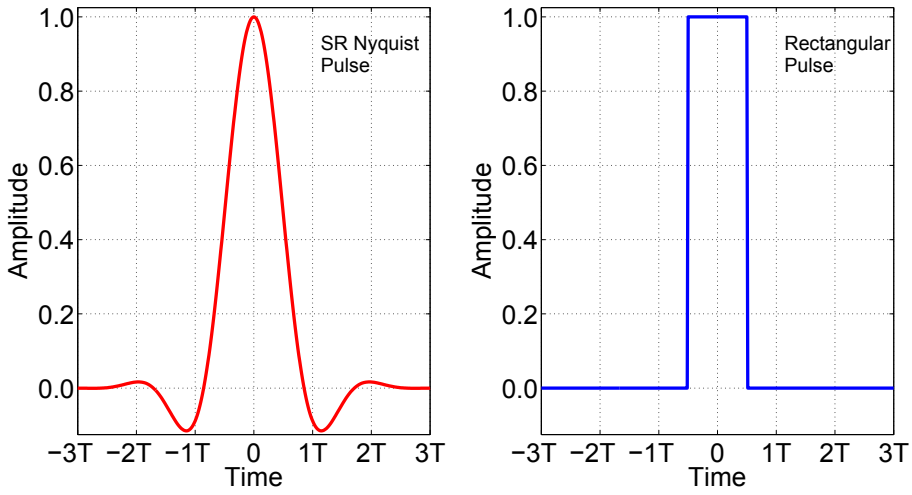

Figure 4.1. Square-root Nyquist (FBMC), and rectangular (OFDM) pulse shapes. T is the symbol duration.

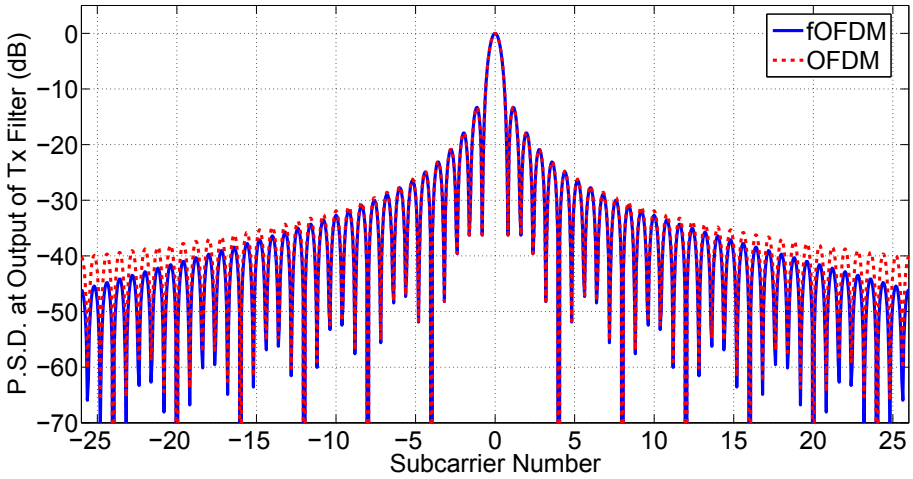

Figure 4.2. PSD of OFDM/fOFDM signal transmitted on subcarrier number 0.

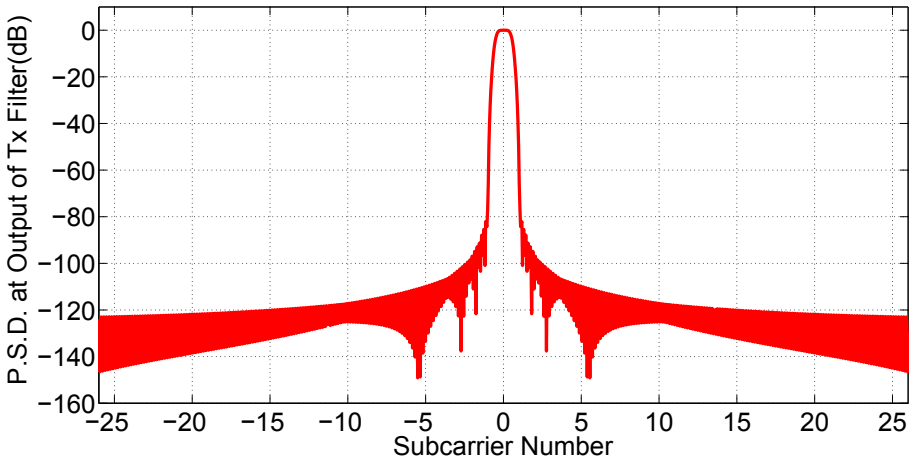

Figure 4.3. PSD of FBMC signal transmitted on subcarrier number 0.

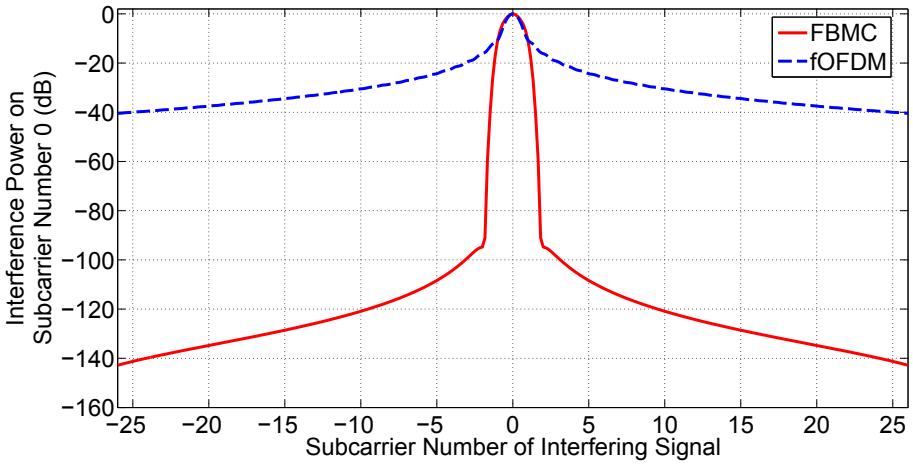

Figure 4.4. Interference power on subcarrier number 0 as a function of the subcarrier number on which the interferer is transmitting.

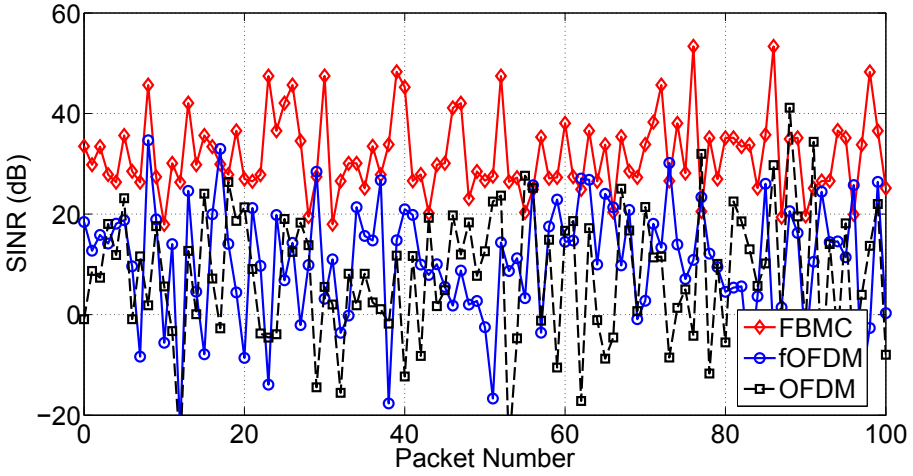

Figure 4.5. Variation of SINR for 100 consecutive packets.

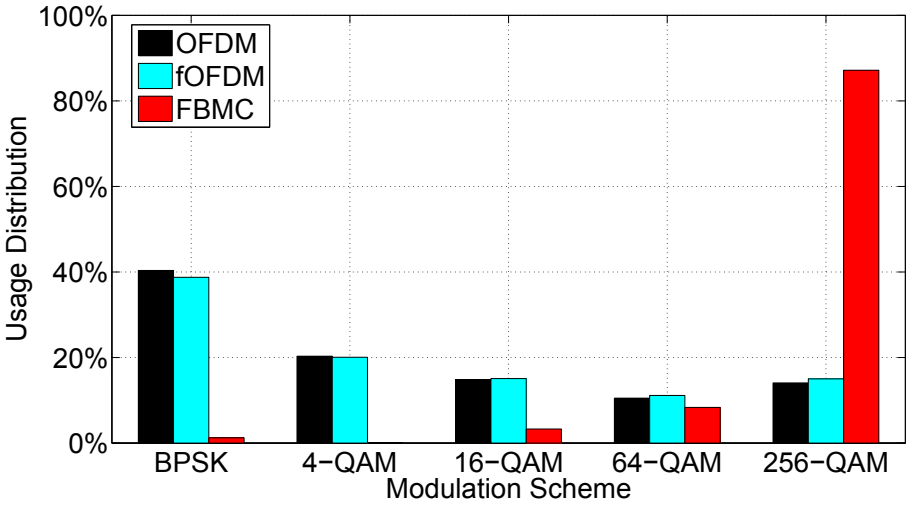

Figure 4.6. FBMC enables modulation schemes with very high data rates.

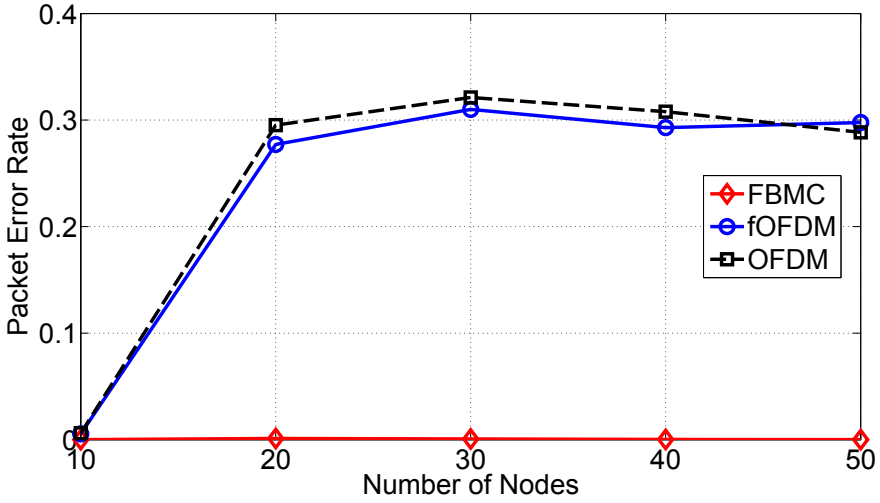

Figure 4.7. OFDM/fOFDM PHY layer produces very high packet error rates, whereas FBMC PHY layer produces practically zero packet error rates.

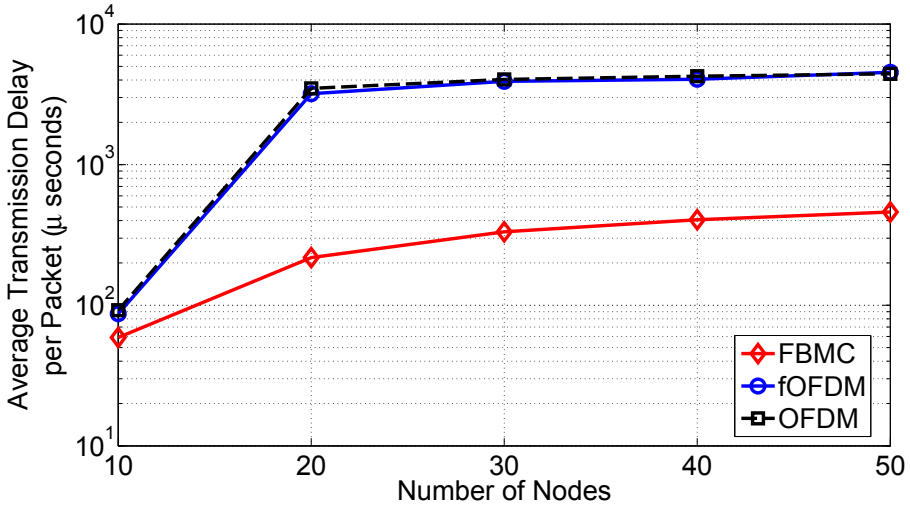

Figure 4.8. Average transmission delay per packet vs number of nodes.

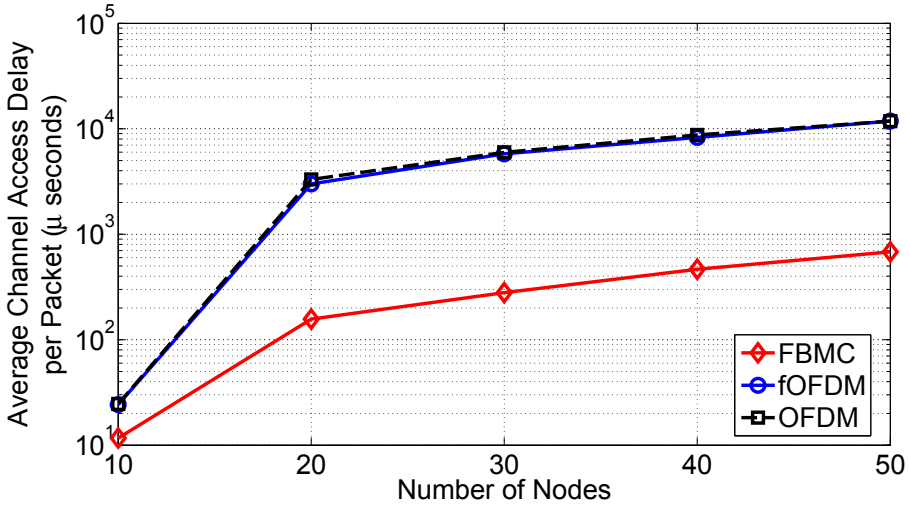

Figure 4.9. Average channel access delay per packet vs number of nodes.

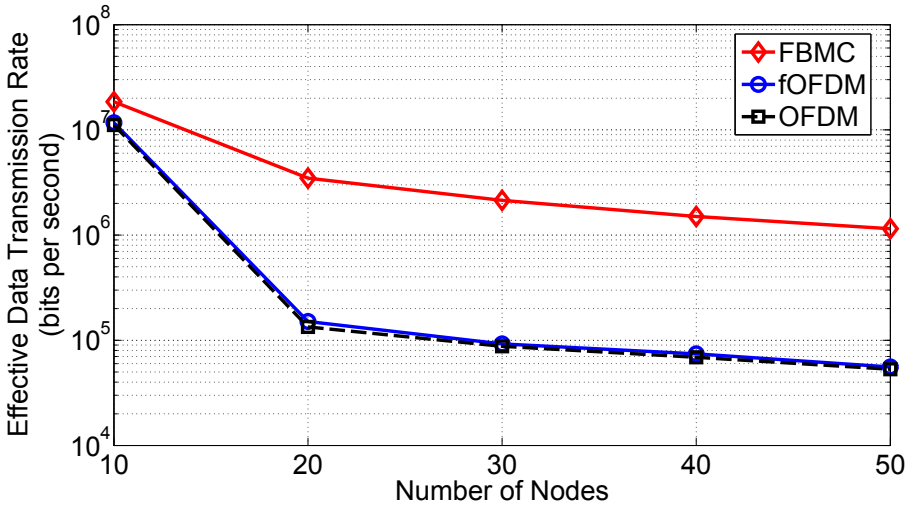

Figure 4.10. Effective data rate vs number of nodes.

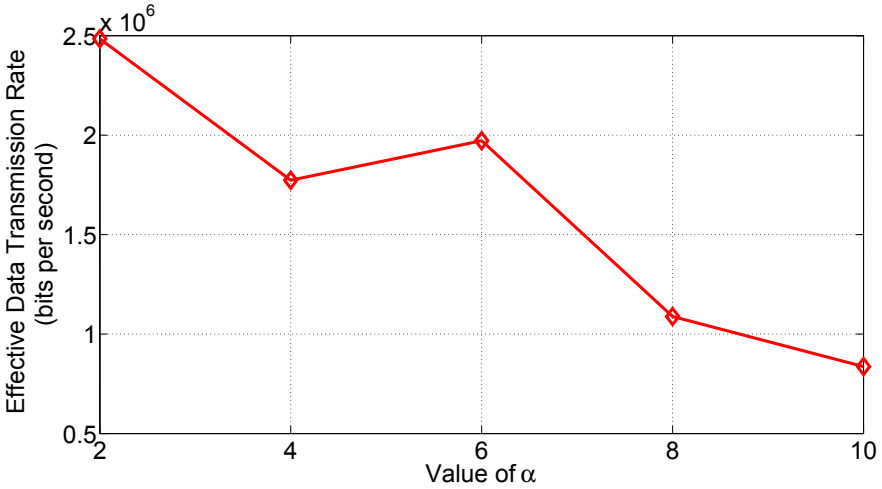

Figure 4.11. Effective data transmission rate as a function of the AIMD MAC parameter, α for the FBMC PHY layer. Here $\beta = 1/\alpha$.

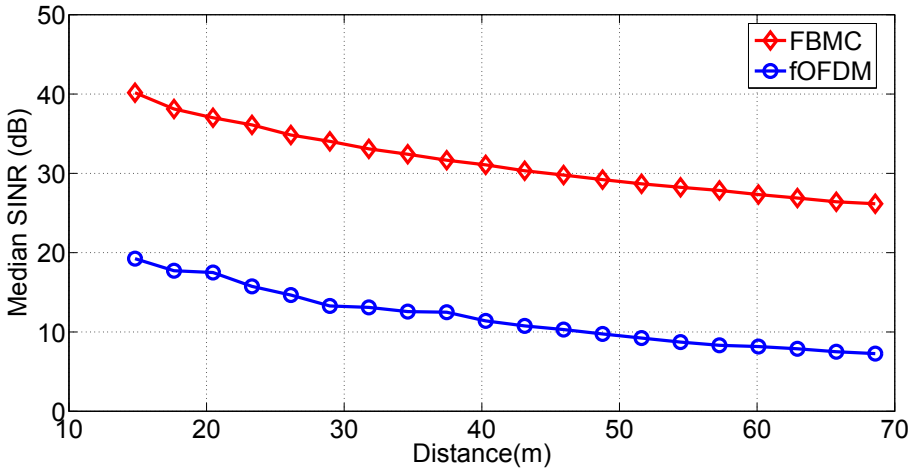

Figure 4.12. Median SINR vs distance between the transmitter and the receiver.

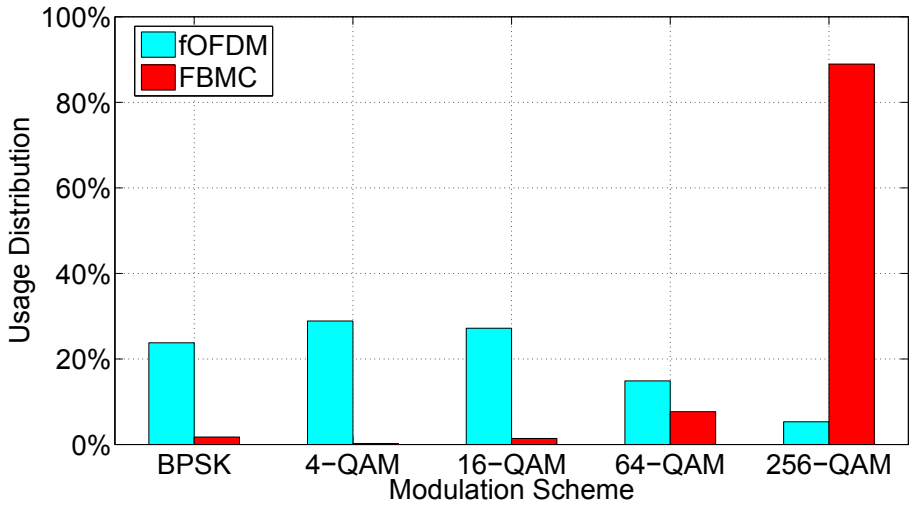

Figure 4.13. FBMC enables modulation schemes with very high data rates.

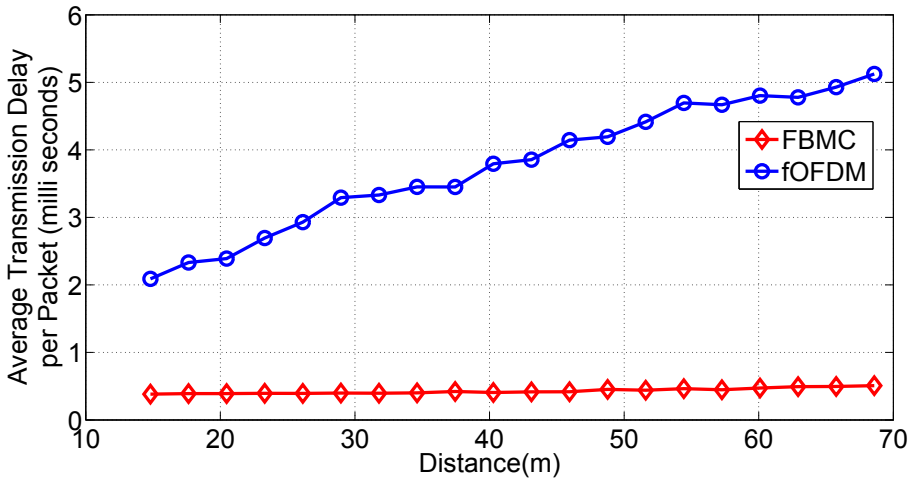

Figure 4.14. Average transmission delay vs distance between the transmitter and the receiver.

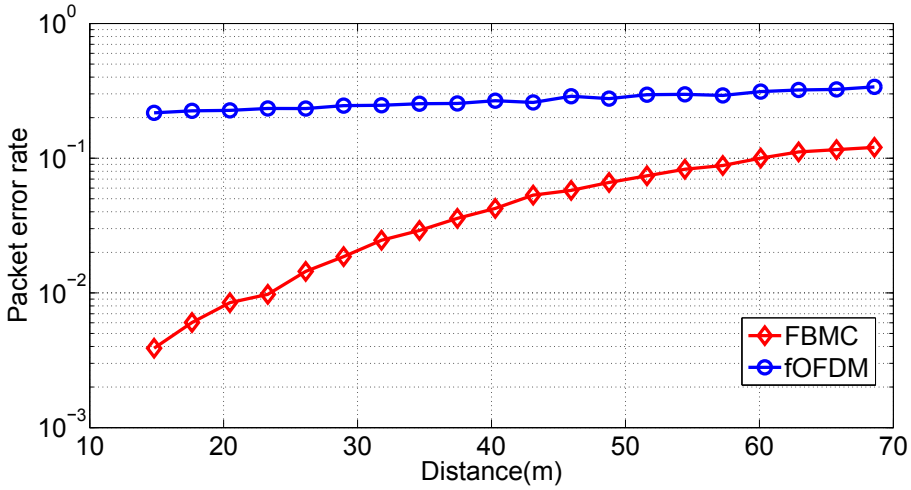

Figure 4.15. Average packet error rate vs distance between the transmitter and the receiver.

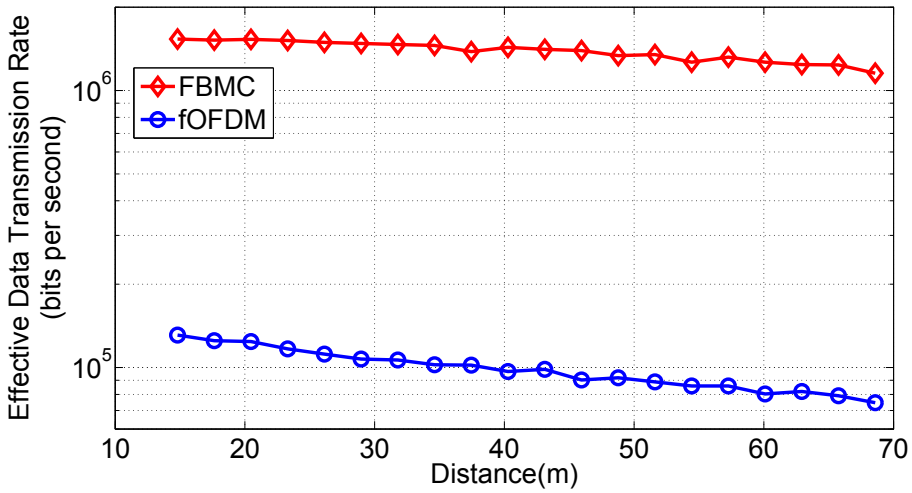

Figure 4.16. Average effective data rate vs distance between the transmitter and the receiver.

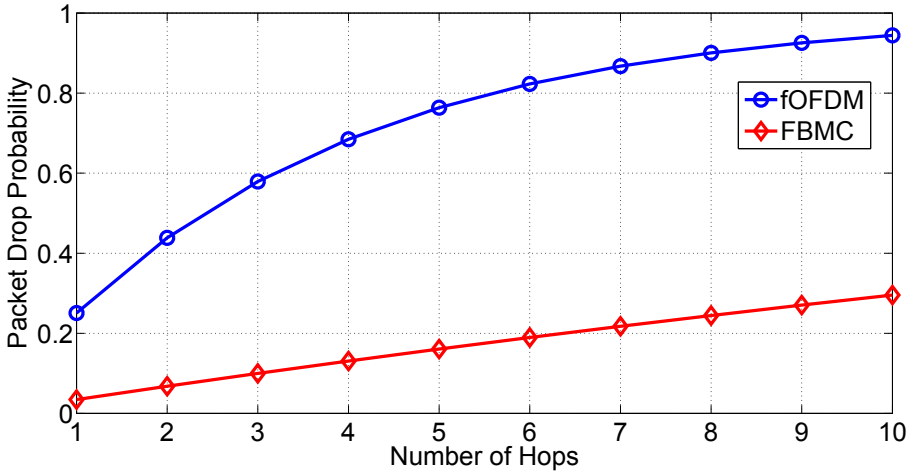

Figure 4.17. Packet drop probability vs number of hops.

Table 4.1. Blasting at full transmit power vs using power control for fOFDM

Performance Metric	**fOFDM (full power)**	**fOFDM (power control)**
Transmission delay (μs)	3210.72	3398.01
Channel access delay (μs)	5964.44	6350.48
Packet error rate (%)	25.06	24.77
Effective data rate (bits per second)	106882	100995

Table 4.2. Blasting at full transmit power vs using power control for FBMC

Performance Metric	**FBMC (full power)**	**FBMC (power control)**
Transmission delay (μs)	411.804	616.203
Channel access delay (μs)	471.023	775.622
Packet error rate (%)	3.443	12.52
Effective data rate (bits per second)	1435185	825269

Table 4.3. Impact of the size of the simulation area for fOFDM

Performance Metric	**fOFDM** ($100m \times 100m$)	**fOFDM** ($200m \times 200m$)
Transmission delay (μs)	3210.72	3422.25
Channel access delay (μs)	5964.44	6563.24
Packet error rate (%)	25.06	27.52
Effective data rate (bits per second)	106882	94739

Table 4.4. Impact of the size of the simulation area for FBMC

Performance Metric	**FBMC** ($100m \times 100m$)	**FBMC** ($200m \times 200m$)
Transmission delay (μs)	411.804	580.142
Channel access delay (μs)	471.023	733.950
Packet error rate (%)	3.443	6.794
Effective data rate (bits per second)	1435185	929976

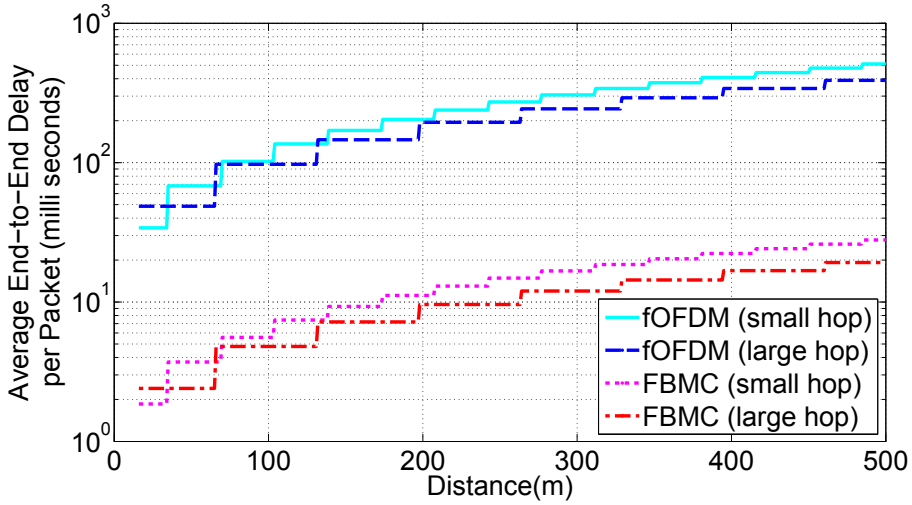

Figure 4.18. Average end-to-end packet delivery delay vs distance between the source and destination nodes. Small and large hop distances are approximately $35m$ and $66m$, respectively.

CHAPTER 5

SUMMARY AND FUTURE WORK

5.1 Summary

We have explored the use of *new cross layer opportunities* to achieve secrecy and efficiency of data transmission in wireless networks. We have shown how our systems oriented, cross layer research enables pervasive wireless devices to efficiently establish private communication channels that are secure from adversaries with unlimited computational power. Additionally, we have also shown how our work enables these devices to efficiently utilize the available wireless spectrum. Our research work has demonstrated how theoretical concepts can be transformed into real-life systems, which in turn can serve as a strong foundation for building innovative, mobile systems and applications.

First, we evaluated the effectiveness of secret key extraction from the received signal strength (RSS) variations in wireless channels using extensive real-world measurements in a variety of environments and settings. Our experimental results showed that bits extracted in static environments are unsuitable for generating a secret key. We also found that an adversary can cause predictable key generation in static environments. However, bits extracted in dynamic environments showed a much higher secret bit rate. We developed an environment adaptive secret key generation scheme and our measurements showed that our scheme performed the best in terms of generating high entropy bits at a high bit rate in comparison to the existing ones that we evaluated. The secret key bit streams generated by our scheme also passed the randomness tests of the NIST test suite that we conducted. We were able to further enhance the rate of secret bit generation of our scheme by extracting multiple bits from each RSS measurement. We have presented these results in two major

papers [14, 43].

Second, we proposed and experimentally evaluated a collaborative secret key extraction scheme for wireless sensor networks. Our experimental results showed that there is a significant increase in secret bit rate per second and per probe as well as per mJ of transmission energy, due to collaboration. We also evaluated the fundamental performance trade-off due to the increased number of measurements versus the increased bit mismatch rate. While it may appear that collaboration requires many nodes, leveraging measurements from many different sensors enables extraction of stronger secret keys at a faster rate and in an energy efficient manner. While the hierarchical approach uses more bandwidth (i.e., more than one channel), it correspondingly reduces the duration over which the channels are occupied. We have presented these results in three papers [55, 43, 56].

Finally, we examined the use of FBMC for best-effort dynamic spectrum access networks. We analyzed the mutual interference power across subcarriers used by different transmitters. We devised a distributed and adaptive MAC protocol that coordinates data packet traffic among the different nodes in the network. We showed that FBMC consistently outperforms OFDM with *an order of magnitude performance improvement* in terms of packet transmission delays, channel access delays, and effective data transmission rate available at the MAC layer in static, indoor environments. We examined the use of FBMC for dynamic spectrum access in a vehicular network setup as well. Through extensive simulations, we showed that FBMC outperforms OFDM with *an order of magnitude improvement over large distances* in vehicular networks. Finally, we also showed that in the case of multihop vehicular networks, FBMC can achieve about 20$\times$ smaller end-to-end packet delivery delays and relatively low packet drop probabilities in comparison to OFDM. These results can serve as guidelines for designing ad hoc, dynamic spectrum access communication standards for future vehicular networks. We have presented these results in three papers [85, 86, 87].

5.2 Future Research Directions

5.2.1 Pervasive Adoption of Secret Key Extraction

Achieving unconditional information security still remains a holy grail. A secret key establishment system using the randomness in the wireless channel is *only a step* towards achieving unconditional security. While our secret key establishment method is capable of producing arbitrarily long secret keys, which when used as one-time pad, can provide security against adversaries with unlimited computational power, more efficient methods of estimating the wireless channel are necessary for wide-spread adoption of this approach. Next, while secret key establishment is only one aspect of secure communication between any two wireless devices, the other primary aspect is authentication, which also remains a major challenge. We can anticipate a pervasive deployment of secret key extraction when it works in conjunction with novel authentication mechanisms such as remote device fingerprinting [27, 26], or human-verifiable authentication using camera phones [88].

5.2.2 Secret Key Extraction Using Feature-rich Measurements

Secret key establishment using wireless channel characteristics can benefit from using a *feature-rich* set of channel measurements obtained from, for example, *ultra-wide-band transceivers*. Additionally, recent developments have made it possible to measure the *channel frequency response* over a *wide-band and MIMO channel* using off-the-shelf Intel WiFi Link 5300 802.11n wireless cards [89]. Each measurement in such datasets captures the attenuation of the wireless channel as a function of time or frequency, and can offer significantly higher entropy for potentially faster key establishment. It will be interesting to explore these new avenues for key extraction in different environments.

5.2.3 Secret Key Extraction under Hidden Terminal Interference

When there are hidden terminals in an 802.11 network, the packet loss rate can increase significantly due to repeated packet collisions/interference at a common receiver. An increase in packet loss rate subsequently reduces the channel sampling rate and hence negatively impacts secret key extraction. *ZigZag decoding* [90] exploits

different interference-free stretches across successive collisions for a given pair of packets to successfully recover them. Gollakota et al. [90] show that the packet loss rates under hidden terminal scenarios can be reduced by two orders of magnitude (from 73% to 0.7%). Thus, when Alice and Bob implement ZigZag decoding, they can obtain estimates of the wireless channel more often over the interference-free stretches than a typical 802.11 receiver and hence can substantially improve the secret key extraction performance even in the presence of hidden terminal interference.

5.2.4 High SNR Measurements for Secret Key Extraction

The number of secret bits that can be obtained from a sample depends on the mutual information between the channel measurements of Alice and Bob. Mutual information, in turn, increases with SINR. Ye et al. show that the mutual information increases roughly at the rate of one secret bit per sample for every 3 dB increase in SNR [30, 42]. In this dissertation, we have shown that square-root Nyquist pulse has very good interference rejection characteristics and can achieve very high SINR in comparison to the rectangular pulse, which is widely used in standards such as 802.11. It will be interesting to explore secret key extraction under square-root Nyquist pulse and determine the increase in secret key generation rate using an actual FBMC implementation.

5.2.5 Real-world Adoption of FBMC

In this dissertation, we have evaluated the cross layer performance of FBMC using simulation models for the case of single input single output systems. However, widespread adoption requires successful demonstration of a real-world FBMC implementation over an actual dynamic spectrum access network built using our AIMD MAC protocol. Furthermore, it also depends on the development of new FBMC methods that enable MIMO capabilities for the enhancement of data rate and/or robustness to errors. While deployment of MIMO technique is better understood in the case of OFDM (e.g., IEEE 802.11n standard), there are only a limited number of existing studies on the development of MIMO-FBMC systems [67]. However, as Farhang et al. [67] have pointed out, FBMC systems can offer the same flexibility as OFDM in adopting the various MIMO techniques. A solid evaluation of MIMO-

FBMC systems is required to investigate its cross layer performance.

5.2.6 Coexistence of FBMC with Legacy-OFDM Systems

While FBMC is promising for dynamic spectrum access networks, it is foreseeable that a full-adoption of FBMC will occur over a certain period of time in the future and until then, users of FBMC and OFDM systems are likely to coexist. Given this scenario, it is important to evaluate the coexistence of OFDM and FBMC users on the same channel during this transition period. Some of the important issues of relevance include – (i) evaluating the performance as a function of the *separation* between the subcarriers of FBMC and OFDM users, (ii) evaluating the performance as a function of the ratio of the numbers of OFDM/FBMC users, (iii) finding new ways to adapt the behavior of the MAC protocol due to the disparity in the performance of OFDM and FBMC users.

5.2.7 Enhancing the Range, Throughput of an FBMC Network

Channel width [84] and WhiteFi [72] adapt the width of a channel (e.g., 5, 10, or 20 MHz) that a node uses to increase throughput or range for the case of an OFDM system. With a fixed number of subcarriers (equal to 64), wider channels yield higher throughput due to smaller symbol durations. On the other hand, narrower channels achieve higher SNR, and hence larger range, due to the fact that the SNR increases when the transmit power per Hz becomes proportionally higher and that the noise power becomes proportionally lower while reducing channel width. Narrow channels are also more resilient to delays spreads because of larger symbol durations; while it results in reducing packet loss/error rates, it correspondingly lowers the data rate. It will be interesting to evaluate the impact of adapting the channel width on the range and throughput on a network when different nodes use the FBMC PHY layer in the context of a dynamic spectrum access network. An important avenue for further exploration will be to find the right balance between the desirable throughput and range on the basis of the available channel width and the surrounding environment. In addition to assigning different subsets of subcarriers to different nodes in a dynamic spectrum access scenario, varying the channel width offers another dimension for the development of new adaptive MAC protocols.

REFERENCES

[1] V. Srivastava and M. Motani, "Cross layer design: a survey and the road ahead," *Communications Magazine, IEEE*, vol. 43, no. 12, pp. 112 – 119, dec. 2005.

[2] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross layer optimization for video streaming over wireless networks," *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 122 – 130, jan. 2006.

[3] J. Gross, J. Klaue, H. Karl, and A. Wolisz, "Cross layer optimization of ofdm transmission systems for mpeg-4 video streaming," *Computer Communications*, vol. 27, no. 11, pp. 1044 – 1055, 2004.

[4] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross layer protection strategies for robust scalable video transmission over 802.11 wlans," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 10, pp. 1752 – 1763, dec. 2003.

[5] M. van Der Schaar and N. Sai Shankar, "Cross layer wireless multimedia transmission: challenges, principles, and new paradigms," *Wireless Communications, IEEE*, vol. 12, no. 4, pp. 50 – 58, aug. 2005.

[6] Q. Liu, S. Zhou, and G. Giannakis, "Cross layer scheduling with prescribed qos guarantees in adaptive wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 5, pp. 1056 – 1066, may 2005.

[7] T. A. Weiss and F. K. Jondral, "Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency," *IEEE Communications Magazine*, vol. 42, no. 3, pp. S8–S14, Mar. 2004.

[8] D. J. Schaefer, "Wide area adaptive spectrum applications," in *IEEE MILCOM*, 2001.

[9] B. Farhang-Boroujeny, "A square-root nyquist (m) filter design for digital communication systems," *IEEE Trans. Signal Proces.*, 56(5), '08.

[10] B. Saltzberg, "Performance of an efficient parallel data transmission system," *IEEE Trans. on Comm. Tech.*, vol. 15, no. 6, Dec. 1967.

[11] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *J. Cryptol.*, vol. 5, no. 1, pp. 3–28, 1992.

[12] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.

[13] L. Greenemeier, "Election fix? switzerland tests quantum cryptography," *Scientific American*, October 2007. [Online]. Available: http://www.sciam.com/article.cfm?id=swiss-test-quantum-cryptography

[14] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *MOBICOM*, 2009.

[15] G. D. Durgin, *Space-Time Wireless Channels*. Prentice Hall PTR, 2002.

[16] M. A. Tope and J. C. McEachen, "Unconditionally secure communications over fading channels," in *Military Communications Conference (MILCOM 2001)*, vol. 1, Oct. 2001, pp. 54–58.

[17] G. Brassard and L. Salvail, "Secret key reconciliation by public discussion," *Lecture Notes in Computer Science*, vol. 765, pp. 410–423, 1994.

[18] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *STOC*, 1989, pp. 12–24.

[19] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, Nov. 2007, pp. 401–410.

[20] S. Mathur, W. Trappe, N. B. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MOBICOM*, 2008, pp. 128–139.

[21] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," in *Proc. 5th ACM Workshop on Wireless Security (WiSe'06)*, Sept. 2006, pp. 33–42.

[22] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776–3784, Nov. 2005.

[23] "NIST," A statistical test suite for random and pseudorandom number generators for cryptographic applications. http://csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501.pdf. 2001.

[24] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *INFOCOM*, 2010, pp. 1837–1845.

[25] C. Kaufman, R. Perlman, and M. Speciner, *Network security: private communication in a public world, second edition*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2002.

[26] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "PARADIS: Wireless device identification with radiometric signatures," in *MOBICOM*, 2008.

[27] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized access points using clock skews," in *MOBICOM*, 2008.

[28] M. Bloch, J. Barros, M. Rodrigues, and S. McLaughlin, "Wireless Information-Theoretic Security," *IEEE Transactions on Information Theory*, vol. 54, no. 6, Jun 2008.

[29] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Info. Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[30] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly gaussian random variables," in *2006 IEEE International Symposium on Information Theory (ISIT'06)*, July 2006, pp. 2593–2597.

[31] "ipwraw," http://homepages.tu-darmstadt.de/ p_larbig/wlan/.

[32] "radiotap," http://www.radiotap.org.

[33] "Converting signal strength percentage to dbm values," http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf.

[34] J. Croft, "Shared secret key establishment using wireless channel measurements," *Ph.D. Dissertation, University of Utah Department of Electrical and Computer Engineering*.

[35] J. E. Hershey, A. A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 3–6, Jan. 1995.

[36] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," *Elsevier Digital Signal Processing*, vol. 6, p. 207212, 1996.

[37] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *ICASSP*, April 2008, pp. 3013–3016.

[38] M. G. Madiseh, M. L. McGuire, S. W. Neville, and A. A. B. Shirazi, "Secret key extraction in ultra wideband channels for unsynchronized radios," in *CNSR*, May 2008.

[39] U. M. Maurer and S. Wolf, "Unconditionally secure key agreement and the intrinsic conditional information," *IEEE Trans. Info. Theory*, vol. 45, no. 2, pp. 499–514, 1999.

[40] C. Ye, A. Reznik, G. Sternberg, and Y. Shah, "On the secrecy capabilities of ITU channels," in *IEEE VTC*, Oct. 2007, pp. 2030–2034.

[41] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in UWB channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, Sept. 2007.

[42] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 240–254, 2010.

[43] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," in *IEEE Transactions on Mobile Computing (preprint)*, 2012.

[44] J. W. Wallace, C. Chen, and M. A. Jensen, "Key generation exploiting mimo channel evolution: Algorithms and theoretical limits," in *EuCAP*, Mar. 2009.

[45] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, May 2009.

[46] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, and A. Reznik, "Secret Key Generation for a Pairwise Independent Network Model," *IEEE Transactions on Information Theory*, vol. 56, no. 12, Dec 2010.

[47] I. Csiszar and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Transactions on Information Theory*, vol. 50, no. 12, Dec 2004.

[48] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Transactions on Mobile Computing*, 2009.

[49] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *IEEE Elecronics Letters*, vol. 27, no. 23, pp. 2145 – 2146, Nov 1991.

[50] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. Wiley, 2004.

[51] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4024 – 4036, Aug 2009.

[52] M. Hulle, "Edgeworth approximation of multivariate differential entropy," *Neural computation*, vol. 17, no. 9, pp. 1903–1910, 2005.

[53] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "A modeling approach on the telosb wsn platform power consumption," *Journal of Systems and Software*, vol. 83, no. 8, pp. 1355 – 1363, 2010, performance Evaluation and Optimization of Ubiquitous Computing and Networked Systems. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0N-4Y6S7GJ-1/2/f4a1e725916b1342a88de24d893098d6

[54] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, Apr. 2005.

[55] S. N. Premnath, S. K. Kasera, and N. Patwari, "Secret key extraction in mimo-like sensor networks using wireless signal strength," *Mobile Computing and Communications Review*, vol. 14, no. 1, pp. 7–9, 2010.

[56] S. N. Premnath, J. Croft, N. Patwari, and S. K. Kasera, "Efficient high rate secret key extraction in wireless sensor networks using collaboration," *ACM Transactions on Sensor Networks (under review)*, 2013.

[57] B. Farhang-Boroujeny, "Multicarrier modulation with blind detection capability using cosine modulated filter banks," *IEEE Trans. Commun.*, 51(12), 2003.

[58] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, ser. VANET '04, 2004, pp. 76–85.

[59] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, ser. VANET '07, 2007, pp. 59–68.

[60] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "Vanetmobisim: generating realistic mobility patterns for vanets," in *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, ser. VANET '06, 2006, pp. 96–97. [Online]. Available: http://doi.acm.org/10.1145/1161064.1161084

[61] U. S. Census Bureau, TIGER Census Files, FIPS 49035, 2006.

[62] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and mac protocols," in *MobiCom*, 2009.

[63] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng, "Supporting Demanding Wireless Applications with Frequency-agile Radios," *NSDI'10*.

[64] 3GPP TS 36.201-820:. Evolved universal terrestrial radio access (E-UTRA); long term evolution (LTE) physical layer; general description.

[65] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless lan," in *SIGCOMM*, 2010.

[66] H. S. Sourck, Y. Wu, J. W. Bergmans, S. Sadri, and B. Farhang-Boroujeny, "Complexity and performance comparison of filter bank multicarrier and OFDM in uplink of multicarrier multiple access networks," *IEEE Trans. on Signal Processing*, Apr. 2011.

[67] B. Farhang-Boroujeny, "OFDM versus filter bank multicarrier," *IEEE Signal processing Magazine*, To appear in May 2011.

[68] B. Hirosaki, "An Orthogonally Multiplexed QAM System Using the Discrete Fourier Transform," *IEEE Trans. Commun.*, vol. 29, no. 7, '81.

[69] M. Morelli, C.-C. J. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proceedings of IEEE*, vol. 95, no. 7, pp. 1394–1427, Jul. 2007.

[70] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comp Networks and ISDN Sys*, vol. 17, pp. 1–14, 1989.

[71] V. Jacobson, "Congestion avoidance and control," *SIGCOMM*, 1988.

[72] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with wi-fi like connectivity," in *SIGCOMM*, 2009.

[73] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993.

[74] O. Awoniyi and F. A. Tobagi, "Packet error rate in ofdm-based wireless lans operating in frequency selective channels," in *INFOCOM*, 2006.

[75] J. Keenan and A. Motley, "Radio coverage in buildings," *British Telecom Technology Journal*, vol. 8, no. 1, pp. 19–24, Jan. 1990.

[76] J. Medbo and J. Berg, "Simple and accurate path loss modeling at 5 ghz in indoor environments with corridors," *VTC*, 2000.

[77] J. Yeo, M. Youssef, and A. Agrawala, "Characterizing the IEEE 802.11 Traffic: Wireless Side," *CS-TR 4570. U Maryland. CS.*, Mar. 2004.

[78] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[79] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, 2007.

[80] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, ser. WMCSA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 90–100. [Online]. Available: http://portal.acm.org/citation.cfm?id=520551.837511

[81] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wirel. Netw.*, vol. 6, pp. 307–321, July 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1019106118419

[82] B. Farhang-Boroujeny and R. Kempter, "Multicarrier communication techniques for spectrum sensing and communication in cognitive radios," *IEEE Communications Magazine, Special Issue on Cognitive Radios for Dynamic Spectrum Access*, Apr. 2008.

[83] S. Sen, R. R. Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," *MobiCom*, 2011.

[84] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "A case for adapting channel width in wireless networks," in *Proceedings of the ACM SIGCOMM conference on Data communication*, 2008, pp. 135–146. [Online]. Available: http://doi.acm.org/10.1145/1402958.1402975

[85] S. N. Premnath, D. Wasden, S. K. Kasera, B. Farhang-Boroujeny, and N. Patwari, "Beyond ofdm: Best-effort dynamic spectrum access using filterbank multicarrier," in *Proceedings of the Fourth International Conference on Communication Systems and Networks (COMSNETS)*, Jan. 2012.

[86] S. N. Premnath, S. K. Kasera, B. Farhang-Boroujeny, and N. Patwari, "Efficient dynamic spectrum access in vehicular networks using filterbank multicarrier," in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief*, ser. ACWR '11. New York, NY, USA: ACM, 2011, pp. 169–176.

[87] S. N. Premnath, D. Wasden, S. K. Kasera, N. Patwari, and B. Farhang-Boroujeny, "Beyond ofdm: Best-effort dynamic spectrum access using filterbank multicarrier," in *IEEE Transactions on Networking (preprint)*, 2012.

[88] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," *Int. J. Secur. Netw.*, vol. 4, no. 1/2, pp. 43–56, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1504/IJSN.2009.023425

[89] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1925861.1925870

[90] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 159–170. [Online]. Available: http://doi.acm.org/10.1145/1402958.1402977