SNUPI:
Sensor Network Utilizing Powerline Infrastructure

Gabe A. Cohn

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2014

Reading Committee:

Shwetak N. Patel, Chair

Joshua R. Smith

Gaetano Borriello

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

SNUPI:
Sensor Network Utilizing Powerline Infrastructure

Gabe A. Cohn

Chair of the Supervisory Committee:
Assistant Professor Shwetak N. Patel
Electrical Engineering and Computer Science & Engineering

Over the past few decades, wireless sensor networks have transformed the way in which we collect information about the world around us. The low cost of the sensor nodes and the simplicity of installation make wireless sensor networks highly scalable, and therefore they are able to obtain rich sets of sensor data over a wide area at relatively low cost. As a result, wireless sensor networks have been widely deployed for a number of sensing applications including environmental, health, home, commercial, industrial, and military applications. Despite the success of many wireless sensor networks, their usage is often severely constrained by limited battery life and the maintenance burden of frequently changing batteries. This limitation arises from a well-known trade-off between the lifetime of a sensor node and its wireless range. As the communication power is reduced to improve the lifetime, the wireless range of the node is also reduced.

This dissertation describes an indoor wireless sensor network called SNUPI (Sensor Network Utilizing Powerline Infrastructure) that overcomes the traditional power/range trade-off by leveraging the existing power line infrastructure in buildings. In this new wireless sensor network, a single base station receiver is connected directly to the power line (*i.e.*, plugged into an outlet). Each node in the sensor network transmits wireless signals that couple to nearby power lines, thus creating signals that travel through the power

line infrastructure to the base station receiver. In this way, the sensor nodes can transmit at much lower power because the signals do not need to propagate over-the-air for the entire path to the base station receiver; they only need to propagate to the nearest power line. In this dissertation, I build upon earlier work using the power line as a receiving antenna for communications, to explore the details required to make such a wireless sensor network both practical and robust. I explore this new power line coupled wireless communications channel and investigate the designs and application-specific optimizations of each element of the sensor network, including the communications protocol, the sensor node, and the base station receiver.

In loving memory of my grandfather Ben Cohn,
who sparked my interest in electrical engineering.

ACKNOWLEDGMENTS

I would sincerely like to thank and acknowledge all of the people that have helped me to reach this great achievement in my life. I could not have done it alone, and am truly grateful for all of your constant support. This section does not even begin to express my gratitude for all of my supporters, nor are there enough pages in this entire dissertation to convey my thanks.

First, I'd like to thank all of my family and friends for their love and support. You have all helped me to continue to work towards my passion, and I am truly grateful. I'm so glad to have spent that last 5 years working toward my Ph.D. in Seattle, surrounded by all of my family and friends. To my parents who have always supported me and given me the strength to continue working hard. To all of my friends who have distracted me from my work and made my graduate school career incredibly enjoyable. Thank you all so much!

The strongest force in my professional life has been my advisor, Shwetak Patel. Thank you Shwetak for making me part of the Ubicomp lab and teaching me so much, both professionally and personally. Your creativity, excitement, and optimism have been a true inspiration to me. Thank you for all of the long meetings and late night phone calls in which you have given me advice and empowered me to find solutions to problems I thought to be impossible. When I first met with you before deciding to come to UW, I told you that I loved to solve problems thought to be impossible, and you have shown me how to do this time and time again!

I am so lucky to have been part of the Ubicomp lab, which I truly believe is the best lab in the world. We have such a unique culture of creativity, intellect, academic diversity, and kindness. Most of all, I'd like to thank the original Ubicomp lab: Jon Froehlich, Eric Lar-

son, Sidhant Gupta, and Tim Campbell. Jon, my writing and presentation style was shaped by you more than anyone else, and I would like to thank you immensely for teaching me how to effectively communicate as a researcher. Eric, your signal processing expertise has been invaluable throughout my time in graduate school, and I will call upon you when I have questions for many years to come. Most of all, I admire your academic integrity and rigor, and know that these traits will make you an excellent researcher and advisor to many students. Sidhant, we have gone through graduate school side-by-side, and I honestly couldn't have done it without you. You have a research style which is very complimentary to mine and makes it a real pleasure to work closely with you. I'm incredibly excited to continue working along side you at Microsoft Research. Tim, your excitement and creativity have been an inspiration to me, and I enjoyed every minute with you in the lab. You accomplished so much while just a "freshman," and I know you will continue to accomplish so much more as your career progresses. Finally, I'd like to thank Mayank Goel who has taken over as my "Tim" for the past few years. Mayank, you are an incredible researcher and even better friend.

I'd also like to thank all of my academic collaborators that have helped to develop the SNUPI technology. I'd like to thank Erich Stuntebeck and Gregory Abowd from Georgia Tech as well as Jagdish Pandey and Brian Otis from UW. A special thanks goes to Matt Reynolds who has been instrumental in the development of SNUPI. Matt, you are the best engineer I have ever met, and I have learned so incredibly much from you. Now that you are in Seattle, I hope to continue working closely with you and learning more everyday.

Along with the academic work on SNUPI, I have been closely involved in the commercialization of the technology, and the development of the WallyHome product. Although both have already been thanked, I'd like to acknowledge Shwetak Patel and Matt Reynolds for working so hard to commercialize this research. I have learned so much from both of you through my experiences at SNUPI Technologies. I would also like to thank many of the

first employees who worked hard to transition my research into reality: Peter Mullen, Don Smyth, Patrick Vilbrandt, Joe Baird, and Aanand Esterberg. In particular, I'd like to thank Peter for helping me collect much of the data that was used throughout this dissertation. It was incredibly tedious, boring, and sometimes painful work, but you never complained, and it was always nice having your company during the data collection.

Finally, I'd like to thank my second research group: the CUE group at Microsoft Research. Although my dissertation covers only SNUPI, the majority of my research was done at MSR with all of you, and in particular with Desney Tan and Dan Morris. Desney, you have been like a second advisor to me, and I am grateful to have been able to work so closely with you for the past 4 years. I am incredibly excited to continue working in the CUE group after I graduate, and I know that we will continue to do great things!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ADC Analog-to-Digital Converter

ALOHA Random access network communications protocol

AM Amplitude Modulation

ARM Low-power computer processor architecture

ARQ Automatic Repeat Query/Request

ASK Amplitude Shift Keying

AVR Low-power microcontroller architecture produced by Atmel

AWG American Wire Gauge

AWGN Additive White Gaussian Noise channel

BCH Cyclic error-correction codes invented by Bose, Chaudhuri, and Hocquenghem

BER Bit Error Rate

BJT Bipolar Junction Transistor

BKSK Binary Frequency Shift Keying

BLE Bluetooth Low Energy

BPL Broadband over Power Lines

BPSK Binary Phase Shift Keying

C/N Carrier to Noise Ratio

COTS Commercial Off-The-Shelf

CPU Central Processing Unit

CRC Cyclic Redundancy Check

$\mathbf{E_b/N_0}$ Energy per bit to noise spectral density ratio

ECC Error Correction Code

EIRP Effective Isotropic Radiated Power

FCC Federal Communications Commission

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FET Field Effect Transistor

FFT Fast Fourier Transform

FM Frequency Modulation

FSK Frequency Shift Keying

HF High Frequency radio spectrum: 3 MHz–30 MHz

IC Integrated Circuit

IEEE Institute of Electrical and Electronics Engineers

IF Intermediate Frequency

IoT Internet of Things

ITU International Telecommunications Union

ITU-R International Telecommunications Union, Radiocommunication Sector

ISM Industrial, Scientific, and Medical radio bands

LG Line-to-Ground connection to the power line

LN Line-to-Neutral connection to the power line

LNA Low Noise Amplifier

LO Local Oscillator

M-FSK M-ary Frequency Shift Keying

M-PSK M-ary Phase Shift Keying

MAC Media Access Control layer of networking stack

MF Medium Frequency radio spectrum: 300 kHz–3 MHz

MOSFET Metal Oxide Semiconductor Field Effect Transistor

MSK Minimum Shift Keying

MSP430 Low-power mixed-signal microcontroller family produced by Texas Instruments

NG Neutral-to-Ground connection to the power line

OOK On-Off Keying

OS Operating System

PCB Printed Circuit Board

PHY Physical layer of networking stack

PLC Power Line Communication

PLCW Power Line Coupled Wireless channel

PLL Phase-Locked Loop

PSD Power Spectral Density

PSK Phase Shift Keying

QAM Quadrature Amplitude Modulation

RCPC Rate-Compatible Punctured Convolutional codes

RF Radio Frequency

RFID Radio Frequency Identification

RLL Run-Length Limited

RSSI Received Signal Strength Indicator

SMU source measure unit

SNR Signal to Noise Ratio

SNUPI Sensor Network Utilizing Powerline Infrastructure

STFT Short-Time Fourier Transform

TDMA Time Division Multiple Access

TI Texas Instruments, a semiconductor manufacturer

UHF Ultra High Frequency radio spectrum: 300 MHz–3 GHz

USRP2 Universal Software Radio Peripheral 2, by Ettus Research

VHF Very High Frequency radio spectrum: 30 MHz–300 MHz

WSN Wireless Sensor Network

Chapter 1

INTRODUCTION AND MOTIVATION

Over the past few decades, wireless sensor networks have transformed the way in which we collect information about the world around us. A wireless sensor network (WSN) is a collection of many sensing devices which collect data from on-board sensors and communicate the sensed information to a centralized base station. Since the sensing devices, commonly called *sensor nodes*, are small, low-cost, and low-power, they can be produced and easily deployed in large numbers. Since the network communicates the data wirelessly, these networks can be very easy to install (*i.e.*, there is little to no infrastructure required). The low cost of the sensor nodes and the simplicity of installation make WSNs highly scalable, and therefore they are able to obtain rich sets of sensor data over a wide area at relatively low cost. As a result, WSNs have been widely deployed for a number of sensing applications including environment and habitat, health and wellness, home, commercial and industrial, and military.

Although it is very easy to *deploy* very large networks of wireless sensor nodes, these networks can be very difficult to *maintain*. The maintenance issue arises from the constant need to change the batteries on the sensor nodes. As the network grows in size, the maintenance issue becomes even more severe. As a result, researchers and engineers have been working for years to optimize the power consumption of WSNs [6, 251]. Some sensor nodes even harvest energy from their environment using solar cells, vibration, temperature gradients, human movement, electromagnetic radiation, and pressure changes [76, 139, 157, 176, 182, 183, 195, 231]. Although considerable research continues to explore energy harvesting sensor nodes, these devices are still very limited by the amount of energy that can be extracted from the environment, and the constrained environments in

which they must operate. The vast majority of practical WSNs still rely on a small battery as the primary power source.

To maximize the lifetime of the small on-node battery, every aspect of the sensor node's power consumption must be optimized. The power consumption of the sensor node is typically divided into three domains: *sensing*, *communication*, and *computation*. The majority of the power is consumed by the communication domain, which is responsible for transmitting and receiving the wireless signals [6]. As a result, there is well-known trade-off between the lifetime of a node and its wireless range. As the communication power is reduced to improve the lifetime, the wireless range of the node is also reduced. Many WSNs use a multi-hop mesh network topology, where each node only needs to communicate with nearby nodes, rather than communicating the full distance to the base station. This approach reduces the power consumption of the transmitter but also requires significant power to continuously run a receiver on every node.

This dissertation describes an indoor wireless sensor network that overcomes the traditional power/range trade-off by leveraging the existing power line infrastructure in buildings. In this new WSN, a single base station receiver is connected directly to the power line (*i.e.*, plugged into an outlet). Each node in the sensor network transmits wireless signals that couple to nearby power lines, thus creating signals that travel through the power line infrastructure to the base station receiver. In this way, the sensor nodes can transmit at much lower power because signals do not need to propagate over-the-air for the entire path to the base station receiver; they only need to propagate to the nearest power line. This method of using the power line as a receiving antenna for communications was first introduced by Stuntebeck *et al.* [44, 210]. In this dissertation, I build upon this early work, explore the details required to make such a WSN both practical and robust, and demonstrate an optimized, application-specific implementation of such a network. Throughout this document, this type of WSN will be referred to as **SNUPI: Sensor Network Utilizing Powerline Infrastructure**.

SNUPI does not refer to a single implementation of a WSN; but rather an entire class of

ultra-low-power wireless sensor networks that leverage the indoor power lines as part of the communications channel. This dissertation therefore serves as a design guide for creating SNUPI sensor networks. Throughout the dissertation, I will discuss how to make design decisions for a variety of different applications. In this way, a researcher or engineer can use this document to design a new type of SNUPI network for their specific application. I have implemented a few specific instances of SNUPI sensor networks in this dissertation, and many of the design decisions that I made were tailored to those specific applications. However, it should be clear that these are just examples taken from a large set of potential SNUPI networks.

1.1 Dissertation Goals

Stuntebeck *et al.* found that the power line infrastructure of a home can be used as a receiving antenna to pick up wireless signals produced by low-power sensor nodes in different locations around the home. They also showed that such a network overcomes the traditional power/range trade-off governing traditional over-the-air WSNs. In other words, a sensor node which couples its wireless signals to a power line can consume less power, while still obtaining the same range inside a home as a traditional over-the-air sensor node [44, 210]. This phenomenon, which Stuntebeck *et al.* referred to as "using the power line as a receiving antenna", is really just a new wireless communications channel. I will call this new communications channel the **power line coupled wireless (PLCW)** channel.

In this dissertation, I study this newly discovered PLCW channel and explore the design and power-optimization of the SNUPI sensor network, which is enabled using the PLCW channel. Specifically, I study the properties of the PLCW channel, and explore the design and application-specific power optimization of each element in the SNUPI sensor network (*i.e.*, the communications protocol, the sensor node, and the base station receiver). Throughout this dissertation, I provide support for my thesis statement:

> The power line infrastructure of a building can be leveraged to enable *practical* and *robust* **long-lived**, **maintenance-free**, indoor wireless sensor networks in

> which **simple**, **ultra-low-power** sensor nodes communicate **low-throughput**, **unidirectional** data to a power line connected base station receiver.

The words in bold represent the design constrains on the SNUPI wireless sensor network. It is therefore a requirement that the network is both **long-lived** and **maintenance-free**. This means the network must be operational without human intervention (*i.e.*, without changing batteries) for at least 5 years[1]. In addition, the network is optimized for **low-throughput**, **unidirectional** data. Low-throughput implies applications requiring a per-node data throughput of less than 10 bits per second. Unidirectional means that data only flows from the nodes to the base station receiver, so a multi-hop network topology is not possible. Finally, the node must be **simple** and **ultra-low-power** so that many nodes can be produced at low cost. The simplicity of the node means that much of the complexity needed to make the network robust must to be deferred to the base station receiver (*i.e.*, the network complexity is not balanced between the transmitter and receiver).

In addition to the design constrains, the words in italics in the thesis statement represent the goals of the network. To demonstrate that the network is *practical* I have fully implemented a multi-node SNUPI network for a specific application, and have shown it to be *robust* enough to fully enable the intended application.

Throughout this dissertation, I explore various different aspects of the SNUPI network, and describe how the design parameters can be tuned for specific applications. I also present the optimal design parameters for a few chosen application scenarios, and demonstrate a fully working SNUPI sensor network for those specific applications. Although there are many details in this dissertation which are tailored to a few specific applications, it should be clear that SNUPI does not refer to any single implementation of a WSN; but rather an entire class of ultra-low-power wireless sensor networks that leverage the PLCW channel. This dissertation is intended to serve as a design guide for creating any SNUPI sensor network.

---

[1] Instead of actually demonstrating a network that has lasted over 5 years without human intervention, I show that the power consumption of the sensor nodes is low enough so that the network will last that long.

1.2 Dissertation Contributions and Organization

There are two main aspects of the SNUPI network which enable its ultra-low-power capabilities. In this dissertation, I individually address each of these aspects:

1. Use of the power line coupled wireless (PLCW) channel (Chapter 3)
2. Application-specific power optimization of every component of the network (Chapters 4–6)

The following sections provide a road map for this document, and describe the goals and contributions for each part of the dissertation. In Chapter 2, I provide a survey of existing wireless sensor networks and contrast them with SNUPI, and then give all necessary background and related work on using the power line for data communication.

1.2.1 Power Line Coupled Wireless Channel Analysis

Although Stuntebeck *et al.* introduced a new communications channel which I call the power line coupled wireless (PLCW) channel (see [44, 210]), there has been very little analysis of the channel. As a result, a number of important questions regarding the channel arise:

- What is the best way to model the channel's background noise?
- What is the path loss of the channel?
- What is best frequency of operation?
- What bandwidth does the channel support?
- What is the best way to couple wireless signals onto the power lines?
- What is the best way to extract communications signals from the power lines?

The answers to many of these questions are needed in order to design a practical and robust WSN that uses the channel. In Chapter 3, I describe my analysis of the power line coupled wireless channel and provide answers to many of the questions listed above.

1.2.2 Design and Optimization of Network

In addition to performing a detailed analysis of the power line coupled wireless (PLCW) channel, this dissertation describes the SNUPI sensor network, which uses the PLCW channel. The design of the SNUPI network can be broken down into three main parts: (1) the communications protocol, (2) the sensor node, and (3) the base station receiver. In Chapters 4–6, I explore the different design parameters for each part of the SNUPI network. In addition, I describe how each of the design parameters can be tuned to optimize the power consumption of the network for a given application.

Chapter 4 describes the design parameters related to the wireless communications protocol. In particular, this chapter discusses each of the following aspects of the communications protocol, as well as how they can be tuned to optimize power consumption for a given application: modulation scheme, bit rate, and forward error correction.

Chapter 5 describes the design parameters related to the sensor node. In particular, this chapter discusses each of the following aspects of the sensor node design, as well as how they can be tuned to optimize power consumption for a given application: antenna, power supply, microcontroller, and transmitter.

Chapter 6 describes the design parameters related to the base station receiver. In particular, this chapter discusses each of the following aspects of the receiver design, as well as how they can be optimized for a given application: connection to the power line, analog receiver hardware, and software to detect and decode packets.

Chapter 2

BACKGROUND AND RELATED WORK

This dissertation introduces a new class of wireless sensor networks (WSN) that operate at much lower power than many existing networks for a specific set of environmental constraints and applications. Distributed sensor networks have been used in the ubiquitous computing (ubicomp) community for decades to obtain rich sensor data from the environment. Within the last several years, a new trend in ubicomp, called infrastructure mediated sensing, has enabled large-scale data collection from only a few sensors strategically placed on the infrastructure of the sensing environment [41, 72, 83, 163–167, 211]. This approach can capture sensor data over a large area of coverage, similar to a distributed sensor network, but only requires a few sensors. Although infrastructure mediated sensing helps address many practical obstacles to deployment of home sensing solutions, it is inherently limited by what information can be practically and reliably extracted from a home's infrastructure. Thus, a need still exists for distributing sensor nodes in cases where obtaining certain data in the home is impossible using just the infrastructure. Although SNUPI is a type of distributed sensing solution, it takes some inspiration from the infrastructure mediated sensing approach. In SNUPI, the coverage and power consumption of the distributed sensor network is improved by utilizing the existing infrastructure in the building (*i.e.*, the power lines). This chapter is divided into two sections, one to discuss the background and related work for wireless sensor networks, and another for the background and related work on communicating information over the power lines.

Section 2.1 provides an in-depth survey of existing wireless sensor network technologies so that the reader can appreciate the contributions of SNUPI wireless sensor networks. The application space of WSNs is also described, and it is noted that SNUPI can only be

used for a subset of the application areas listed for WSNs in general. This section also presents a list of low-power sensors that can be used with SNUPI and other WSNs. SNUPI sensor networks are quite different from traditional WSNs, and many of those differences are highlighted throughout this section.

Section 2.2 provides a summary of the background and related work for communicating data over the power line. In the last section (*i.e.*, Section 2.2.3), a description of the past work on the power line coupled wireless (PLCW) channel is provided. This is the channel that enables the SNUPI technology, and is one of the most important aspects in understanding why SNUPI sensor networks are so low power. This section describes the previous work done on the PLCW channel, whereas Chapter 3 discusses my analysis of the PLCW channel in detail.

2.1 Wireless Sensor Networks

Wireless sensor networks (WSN) are anything but new. As with many technologies of the late 20th century, sensor networks were invented during the Cold War era for military surveillance. Commercial WSNs began appearing in the 1980s and every since there has been significant research and development on them [206]. WSN research is a very large field and I do not intend for this section to even begin to give a comprehensive overview of it. For a comprehensive overview of WSNs, please see one of the many books on the subject at your local library. For a survey of some of the past research papers on the subject, please see [6] and [251].

In 2007, the authors of a new book titled "Wireless Sensor Networks" categorized all of the recent academic publications related to WSN research. I have grouped their fine-grained classification into four main categories: applications and use, networking (*i.e.*, data link, network, and transport layers), software (*i.e.*, security, database, storage, compression, and middleware), and hardware (*i.e.*, architecture, wireless radio, physical layer, and power optimization). As of 2007, 38.48% of publications focused on applications and use, 31.82% on networking, 20.91% on software, and the remaining 8.79% on hardware [206].

In contrast to much of the literature, this dissertation will focus primarily on hardware, and the lower layers of the networking stack (*i.e.*, primarily the physical and data link layers). One focus of this work is that application-specific optimization and simplicity results in considerable performance gains. As such, I will abandon much of the complexity of the networking stack that much of the literature discusses, and instead focus on simplicity and optimization. As a result, this related work section does not discuss complex network topologies, media access control (MAC), routing, security, compression, or time synchronization. Instead it focuses on the past work more generally related to sensing and sensor node hardware platforms. More specific examples of past work related to individual elements of the sensor network can be found throughout the dissertation in the appropriate section.

In this section, I intend to give an overview of certain aspects of existing WSNs, while simultaneously pointing out the differences between many of the existing WSNs and the SNUPI technology. SNUPI is quite different from existing WSNs in several ways, and this section is intended to point out many of those differences.

In Section 2.1.1, I give a brief overview of the application space for WSNs, and highlight the subset of application areas in which SNUPI networks can be used. In Section 2.1.2, I give a list of typical sensors that WSNs employ, and again highlight the subset of sensors applicable to SNUPI. In Section 2.1.3, I describe the physical layer communications systems typically used in WSNs, and contrast those with SNUPI. Finally, Section 2.1.4 presents a survey of notable wireless sensor nodes, highlights trends among the existing nodes, contrasts the SNUPI node with most of the existing work.

2.1.1 Applications

The application space for wireless sensor networks is nearly limitless, and researchers have been applying WSNs to new applications for decades. In the words of David Culler, one of the creators of Berkeley Mote family of sensor nodes, “The applications are everywhere!” [94]. In 2007, the authors of a book titled “Wireless Sensor Networks” listed over 220

applications of the WSNs that had already been implemented [206]. This section lists some of the most common applications areas for WSNs and provides a list of citations for each area so that the interested reader can dig deeper.

In general, the applications of WSNs can be divided into two main categories: *monitoring* and *tracking* [251]. The applications of WSNs can also be divided into several application domains, as shown below: *military*, *environment and habitat*, *health and wellness*, *home*, and *commercial and industrial* applications.

Military Applications:

- surveillance and reconnaissance [3, 10, 28, 53, 64, 87, 105]
- monitoring friendly forces, equipment, and supplies [53, 65, 94, 105, 198]
- threat detection and targeting [10, 53, 201, 230, 238]
- detection of nuclear, biological, or chemical weapons [105]

Environment and Habitat Applications:

- monitoring and studying air, water, and soil
  [3, 5, 28, 33, 55, 64, 78, 94, 99, 136, 147, 152, 172, 196, 209, 226, 227, 239, 240, 247]
- studying biological systems and populations
  [3, 24, 31, 32, 35, 64, 94, 105, 129, 172, 215, 225, 227, 232, 238, 247, 253]
- agriculture [13, 18, 28, 30, 31, 64, 116, 129, 138, 156, 224, 234, 237, 238]
- natural disaster (*i.e.*, flood, earthquake, and volcano) detection [7, 25, 78, 239, 240]

Health and Wellness Applications:

- monitoring telemetry and physiology (at home or in hospitals) [9, 12, 14, 17, 28, 29, 34, 37, 56, 64, 97, 113, 121, 131–134, 137, 149, 153, 155, 172, 174, 194, 198, 208, 214, 216, 228]
- elder care [14, 17, 34, 56, 93, 98, 113, 151, 153, 214, 229]
- tracking personnel, equipment, and supplies in hospitals [17, 65, 110, 113, 137, 174]

- public health monitoring [5, 55, 109, 113, 152, 226]
- accessible technologies [14, 113, 238]
- drug administration [174, 199]
- search & rescue and disaster relief [25, 96, 113]

Home Applications:

- smart environments (home and commercial)
  [1, 28, 63, 64, 86, 94, 154, 168, 173, 174, 193, 198, 205, 207, 216, 219, 238, 241, 248]
- home automation [74, 77, 119, 241]
- energy monitoring [62, 86, 101, 235, 241]
- home security [95, 111, 172, 174]

Commercial and Industrial Applications:

- asset tracking and inventory control and management
  [16, 25, 28, 64, 65, 141, 173, 174, 181, 197, 198, 238]
- smart factories, equipment surveillance and preventative maintenance, and structural monitoring [16, 22, 25, 28, 64, 65, 94, 104, 105, 114, 115, 118, 172, 174, 238, 245]
- environmental control in buildings [94, 154, 174, 193, 198, 205, 235, 241]
- infrastructure monitoring [40, 62, 75, 82, 122, 127, 140, 190]
- traffic monitoring and control [25, 28, 64, 79, 94, 198, 202]

SNUPI Applications It should be noted that the SNUPI WSN cannot be used for all of these domains. The SNUPI network only works inside of buildings and thus any WSN application that requires nodes to be placed outdoors will not work with SNUPI. This eliminates most military, environment and habitat applications. Many of the home, commercial, and industrial applications seem very well suited to SNUPI. Furthermore, SNUPI can be used for several health applications. Note that SNUPI can even be used for wearable and implantable health and wellness applications. In these scenarios, the SNUPI node would

move around with the person, and thus would not always be within a building. However, this can still work because the SNUPI node can log data while outside of the home, and then transmit the data to the base station when the user returns home each day. A more thorough discussion of the applications of SNUPI can be found in Chapter 7.

2.1.2 Sensors

Wireless sensor nodes can carry a wide variety of sensor types; however, are often limited by the power consumption of the sensor. The power required for sensing is sometimes greater than the power required to run the CPU, and is at times even greater than the power required to run the wireless transceiver. As a result, long-lived, low-power sensor nodes are limited in the types of sensing that can be performed. Sensors can be divided into two main categories: active and passive. Active sensors typically emit some type of signal into the environment and then sense the environment's response to the new signal. This type of sensor can consume considerable power in the production of the probing signal. As a result, low-power sensor nodes typically must rely only on passive sensors. Passive sensors are typically resistive or capacitive sensors in which the resistance or capacitance changes as a function of some environmental condition. Common examples of passive sensors include light, temperature, humidity, barometric pressure, force, strain, and stress sensors. Table 2.1 compares several common sensors [107]. As can be seen from the table,

Sensor	**Accuracy**	**Sample Rate [Hz]**	**Startup [ms]**	**Current [µA]**
Photoresistor	N/A	2000	10	1235
Thermistor	5 K	2000	10	126
Thermopile	3 K	2000	200	170
I2C Temperature	1 K	2	200	150
Barometric Pressure	1.5 mbar	10	500	10
Humidity	2 %	500	500–3000	775

Table 2.1: Characteristics of example low-power sensors. Reproduced from [107].

power consumption of the sensor itself is not the only factor that contributes to the total sensing power. Sensors often have a very long startup times and thus a significant amount of energy can be consumed during the sensor startup. In addition, it is important to notice the sampling rate for each sensor type. The power consumption of the signal conditioning hardware and analog-to-digital converter (ADC) can also be quite significant, and typically scale with the sampling rate or signal bandwidth.

2.1.3 Physical Layer Communications

This section discusses aspects of the RF communications physical layer used by many wireless sensors nodes. Although most WSNs communicate using an RF link, some WSN use an optical [105, 227, 238], acoustic or ultrasonic [227] channel instead.

Carrier Frequency

The choice of carrier frequency is very important as it impacts many aspects of the sensor network including: power consumption, range, bandwidth, antenna design, and bit error rate. It is most convenient for sensor networks to operate in unlicensed radio bands. In particular, the industrial, scientific, and medical (ISM) unlicensed bands are ideally suited for WSNs. The ISM bands are defined by the ITU-R[1] (International Telecommunications Union, Radiocommunication Sector) and are listed in Table 2.2. Due to the trade-off between power consumption and antenna efficiency the UHF region of the radio spectrum (300 MHz–3 GHz) is best for WSNs [171]. The only ISM bands in this range are 433.92 MHz, 915 MHz, and 2.45 GHz. Since the 433 MHz band is only available in Europe, it is common in the USA to use the 315 MHz band for short range devices (SRD), specified in Section 15.231 of the FCC[2] (Federal Communications Commission) Rules & Regulations [67]. Similarly, the 915 MHz ISM band is only available in the Americas, so the 868 MHz SRD band is commonly used in Europe, as specified in ERC-REC 70-03

[1] http://www.itu.int/en/publications/ITU-R

[2] http://www.fcc.gov

Center Freq.	Freq. Range	Bandwidth	Availability	ITU-R Reg.
6.78 MHz	6.765 – 6.795 MHz	30 kHz	Subject to local acceptance	5.138
13.56 MHz	13.553 – 13.567 MHz	14 kHz	Global	5.150
26.12 MHz	26.957 – 27.283 MHz	326 kHz	Global	5.150
40.68 MHz	40.66 – 40.70 MHz	40 kHz	Global	5.150
433.92 MHz	433.05 – 434.79 MHz	1.84 MHz	Region 1: Europe & Africa, Subject to local acceptance	5.138
915 MHz	902 – 928 MHz	26 MHz	Region 2: Americas	5.150
2.45 GHz	2.4 – 2.5 GHz	100 MHz	Global	5.150
5.8 GHz	5.725 – 5.875 GHz	150 MHz	Global	5.150
24.125 GHz	24.00 – 24.25 GHz	250 MHz	Global	5.150
61.25 GHz	61.0 – 61.5 GHz	500 MHz	Subject to local acceptance	5.138
122.5 GHz	122 – 123 GHz	1 GHz	Subject to local acceptance	5.138
245 GHz	244 – 246 GHz	2 GHz	Subject to local acceptance	5.138

Table 2.2: Unlicensed Industrial, Scientific, and Medical (ISM) radio bands. Regulated by the International Telecommunications Union Radiocommunication Sector (ITU-R) [102].

of the CEPT ECC[3] (European Conference of Postal and Telecommunications Administrations, Electronic Communications Committee) [66]. Almost all wireless sensor networks use one of the UHF ISM or SRD bands. In fact, all of the wireless sensor nodes listed in Table 2.3, with the exception of SNUPI, use one of these bands. SNUPI also uses an ISM band, but it uses an HF (3 MHz–30 MHz) band, which is two orders of magnitude lower in frequency than the UHF band. SNUPI's use of the HF ISM bands comes from the properties of the power line coupled wireless channel and is discussed in detail in Chapter 3.

Network Standards

As WSNs have become more popular, a number of standards have been developed to allow different sensor nodes to interface with a variety of networks. The traditional wireless network standards have been used as a baseline, but a number of standards have been developed specifically with the low-power consumption requirement needed for WSNs. The more common standards include IEEE 802.15.4, ZigBee, WirelessHART, ISA100.11, IEFT

[3] `http://www.cept.org/ecc`

6LoWPAN, IEEE 802.15.3, and Wibree [251]. This section gives a brief overview of the most popular standards.

IEEE 802.15.4 The IEEE 802.15.4 standard is maintained by the IEEE 802.15.4 Task Group[4] and specifies the physical (PHY) layer and media access control (MAC) layer for low-rate wireless personal area networks (LR-WPAN). It is the most popular standard for WSNs today, and is the base standard on which ZigBee, WirelessHART, and ISA100.11 are built atop. These additional standards just specify the higher layers in the networking stack, and rely on 802.15.4 for the lowest layers. The physical layer specifies the following carrier frequencies: 868 MHz (Europe), 915 MHz (North America), and 2.4 GHz (Worldwide). The maximum bit rate supported is 250 kb/s. The MAC layer specifies either a star or peer-to-peer topology. This allows simple communications with a central base station, or complex ad hoc and self-organizing networks.

ZigBee The ZigBee standard is maintained by the ZigBee Alliance[5] and specifies the network and application layers on top of the IEEE 802.15.4 standard. Like 802.15.4, ZigBee is optimized for simple, low-power, and low-rate communications. ZigBee networks often use multi-hop mesh networks to allow data communication to propagate over large distances using many short range transmissions. ZigBee is simpler and lower power than other common wireless personal area networks like Bluetooth, and has thus become the primary standard used by WSNs. Almost every major analog semiconductor company now makes a ZigBee-compliant transceiver. ZigBee has become common in home automation, energy monitoring, traffic management, industrial monitoring, and consumer sensing and actuation devices.

---

[4] http://www.ieee802.org/15/pub/TG4.html

[5] http://www.zigbee.org

Power Consumption

Since wireless sensor nodes operate on a limited power supply (*i.e.*, a small battery or harvested energy source), optimizing for power consumption is often the most difficult engineering challenge. The power consumption of a sensor node is often divided into three domains: *sensing*, *computation*, and *communication*. The sensing power is governed by the type of sensor being used and is discussed in Section 2.1.2.

It is generally accepted that the energy required for communication is significantly higher than the energy for computation. It is common to express to the ratio of the energy required to transmit one bit over the energy required to execute one instruction. This ratio is 1500–2700 for the UCLA/Rockwell WINS node, 220–2900 for the UCLA Medusa MK-II node, and about 1400 for the Sensoria WINS NG 2.0 node [178]. The WeC Berkeley Mote has a ratio around 190 [89]. Regardless of the exact ratio, it is clear that for traditional radios it is an order of magnitude less expensive for computation than for wireless communication. The low frequency and short transmission distance of SNUPI changes this trend, and puts computation and communication on the same order of magnitude (see Chapter 5 for details).

Traditionally, in wireless systems it was assumed that transmitting consumed more power than receiving. In fact, it was assumed that the average power consumption, which determines the lifetime, was dominated by the transmit power. For the short ranges that WSNs communicate, the required transmit power is much lower than these traditional wireless systems. As a result, it has been shown that the transmit and receive power are roughly the same, and in many cases the receive power is even greater than the transmit power [178]. This trend can be seen in Table 2.3. Since receive power is so high, it is impractical in many systems to simply have the receiver on at all times when the node is not transmitting. Duty cycling the receiver is thus an added complexity which must be dealt with in any bidirectional sensor network. Note that the SNUPI network is unidirectional, and thus there is no receiver on the node.

In addition to the transmitter and receiver power consumption being almost equal, the

idle power is typically about the same as the receive power [178]. However, the power consumption in sleep is typically 3 orders of magnitude lower than the idle or active power (*e.g.*, see Table 2.3). As a result of this, it is important to duty cycle the node in order to maximize the time in sleep. Sensor nodes are typically active or idle only 1% of the time or less. However, there is also a time and energy cost for transitioning between power states. For the μAMPS node, it takes 466 μs to start up the transmitter and consumes 58.7 mW during this time [142, 143]. This means that 27.4 μJ of energy is expended each time the node transitions from sleep to transmitting. Note that the μAMPS node consumes 151 nJ/bit while transmitting—thus the startup energy penalty is equivalent to transmitting 181 bits! This transition energy cost is therefore significant and care must be taken to trade-off between the gains of switching to a lower power mode and the loss from transitioning between modes.

2.1.4 Wireless Sensor Node Survey

Practical, low-power wireless sensor nodes have been developed in the academic and commercial research communities since the early 1990s. This section lists some of the more notable WSN platforms, highlights the popular trends, and contrasts the typical sensor node designs with the SNUPI design. Table 2.3 compares many of the notable sensor nodes in approximate chronological order. The dates were determined based on publication dates and other announcements, so they may not be accurate. Note that the power numbers were taken from publications and datasheets, and each author's definition of active or sleep power is different, thus it is very difficult to compare power numbers. In addition, it is not fair to compare power consumption without considering the range, robustness, or computational capabilities of each node. Despite this, I have included the transmit (Tx), receive (Rx), and sleep power numbers that I could find for each of the nodes as a general point of comparison. A blank cell indicates that I could not find any information.

There are a number of different sensor nodes available. Table 2.3 primarily only lists the research platforms. In addition, there are a number of companies making wireless sensor

Sensor Node (Creator)	Date	Frequency	RF Radio (Standards)	CPU	OS	Tx Power	Rx Power	Sleep Power	References
WINS (UCLA/Rockwell)	1996	900 MHz	RDSSS9M	StrongARM	MicroC/OS	1080 mW	752 mW	416 mW	[4, 172, 189]
WeC Mote (Berkeley)	1998	916 MHz	TR1000	AVR	TinyOS	51 mW	24 mW	45 µW	[89, 170]
René / René2 Mote (Berkeley)	1999	916 MHz	TR1000	AVR	TinyOS	51 mW	24 mW	45 µW	[88, 90, 170]
Dot Mote (Berkeley)	2000	916 MHz	TR1000	AVR	TinyOS	51 mW	24 mW	45 µW	[170]
µAMPS (MIT)	2001	2.4 GHz	LMX3162 (Bluetooth)	StrongARM	µ-OS	1040 mW	270 mW	10 mW	[198]
Smart-Its (ETH Zurich)	2001	2.4 GHz	ROK101007 (Bluetooth)	AVR		660 mW	330 mW	106 mW	[108]
Mica Mote (Berkeley)	2001	916 MHz	TR1000	AVR	TinyOS	37.5 mW	31.5 mW	60 µW	[91, 170]
Medusa MK-II (UCLA)	2002	916 MHz	TR1000	ARM7+AVR	eCos	27.5 mW	22.2 mW	20 µW	[189]
iBadge (UCLA)	2002	2.4 GHz	ROK101007 (Bluetooth)	AVR					[36]
EYES (U. of Twente)	2002	868 MHz	TR1001	MSP430	EYES				[57]
Mica2 Mote (Crossbow)	2002	916 MHz[a]	MPR400	AVR	TinyOS	105 mW	54 mW	48 µW	[49, 50]
Intel Mote (Intel Research)	2003	2.4 GHz	TC2001P (Bluetooth)	ARM7	TinyOS	65 mW	65 mW	300 µW	[88, 112, 148]
BTNode (ETH Zurich)	2003	2.4 GHz	ROK101007 (Bluetooth)	AVR	TinyOS-like	160 mW	160 mW	500 µW	[20, 21]
MicaZ Mote (Crossbow)	2004	2.4 GHz	MPR2400 (802.15.4)	AVR	TinyOS	76.2 mW	83.1 mW	48 µW	[51, 88]
Telos (Berkeley)	2004	2.4 GHz	CC2420 (802.15.4)	MSP430	TinyOS	58.5 mW	65.4 mW	15.3 µW	[170]
TCM 120 (EnOcean)	2004	868 MHz	TCM 120			275 mW	165 mW	200 µW	[59, 192]
ScatterWeb (Free U. of Berlin)	2005	868 MHz	TR1001	MSP430		24 mW	21 mW	24 µW	[191]
uPart (U. of Karlsruhe)	2005	315 MHz	rfPIC12F675	PIC		21 mW	No Rx	2.4 µW	[19]
XSM (Berkeley)	2005	433 MHz	CC1000	AVR	TinyOS	72 mW	48 mW	33 µW	[58]
Eco (UC Irvine)	2005	2.4 GHz	nRF24E1	8051		48 mW	55 mW	180 µW	[161]
XBee (Digi)	2005	2.4 GHz	(802.15.4)			149 mW	165 mW	33 µW	[54]
MITes (MIT)	2006	2.4 GHz	nRF24E1	8051					[220]
SunSPOT (Sun)	2006	2.4 GHz	CC2420 (802.15.4)	ARM9	Squawk Java VM	120 mW	120 mW	99 µW	[212, 213]
Iris Mote (Crossbow)	2007	2.4 GHz	XM2110 (802.15.4)	AVR	TinyOS	78 mW	75 mW	24 µW	[48, 103]
Imote2 (Crossbow)	2007	2.4 GHz	CC2420 (802.15.4)	XScale	TinyOS, Linux	297 mW	297 mW	1755 µW	[47, 120]
3rd Gen. Z-Wave (Zensys)	2007	868/915 MHz	ZW0301 (Z-Wave)	8051		108 mW	69 mW	7.5 µW	[252]
Dolphin (EnOcean)	2009	903 MHz[b]	TCM 300 / STM 300	8051		72 mW	99 mW	0.6 µW	[60, 61, 242]
SNUPI (UW)	2010	27 MHz	Custom	MSP430	None	1 mW	No Rx	1.5 µW	[44]

Table 2.3: Table of commonly used wireless sensors nodes, sorted in approximate chronological order.

[a] Also available in 315, 433, and 868 MHz. There is also a smaller version (*i.e.*, Mica2Dot) with the same specifications.

[b] Also available in 315 and 868 MHz.

nodes for commercial use. Some of the earliest companies include: Crossbow[6], EnOcean[7], Dust Networks[8], Moteiv[9], Ember[10], Sun Labs[11], Z-Wave[12], Millennial Net[13], Sensicast[14], and Sensoria[15]. With the recent upsurge in the Internet of Things (IoT) and smart home movements, there are many more startup companies and established companies competing in the wireless sensor network space. In addition, almost every major analog semiconductor manufacturer now makes a ZigBee[16] compliant transceiver. In the research community, the most popular nodes are the family of Berkeley Motes. The family includes the WeC, René, René2, Dot, Mica, MicaDot, Mica2, Mica2Dot, MicaZ, Imote, Telos, and eXtreme Scale Mote (XSM), as well as many derivatives of these. These nodes were developed jointly between UC Berkeley, Crossbow, and Intel Research, and have been successfully commercialized by Crossbow and Moteiv.

Wireless Communication There is an important trend concerning the type of wireless communication used by most WSNs. The early nodes relied primarily on the 915 MHz ISM band, and favored the RF Monolithics (RFM) TR1000 radio transceiver. In the early 2000s, there was a shift to the 2.4 GHz ISM band, and in particular the Bluetooth standard. The most popular Bluetooth transceiver was the Ericsson ROK101007. Around 2004, Bluetooth was replaced by the much lower power IEEE 802.15.4 standard. The 802.15.4 standard is the basis for ZigBee as well as other low-power standards for WSNs (see Section 2.1.3 for

---

[6]Crossbow Technology, Inc., `http://bullseye.xbow.com:81/index.aspx`

[7]EnOcean, `http://www.enocean.com`

[8]Now part of Linear Technology `http://www.linear.com/products/wireless_sensor_networks`

[9]Now part of Sentilla, `http://www.sentilla.com`

[10]Now part of Silicon Labs, `http://www.silabs.com/products/wireless/zigbee`

[11]Now part of Oracle, `http://www.sunspotworld.com`

[12]Z-Wave Alliance, `http://www.z-wave.com/`

[13]Millennial Net, `http://www.millennialnet.com`

[14]Now Sensaphone, `http://www.sensaphone.com`

[15]No longer has an online presence

[16]ZigBee Alliance, `http://www.zigbee.org`

details). These 802.15.4 compliant nodes also used the 2.4 GHz ISM band, and typically employed the Chipcon[17] CC2420 transceiver. Although not represented in the research nodes shown in Table 2.3, there has recently been a prevalence for using Bluetooth Low Energy (BLE), a new protocol targeted at even lower power consumption than the Bluetooth standard. Recent work has shown BLE to be even more energy efficient than ZigBee [52]. Like ZigBee and Bluetooth, BLE also uses the 2.4 GHz ISM band for communication. In addition to the prevalence of the 915 MHz and 2.4 GHz ISM bands, it should be noted that all of the nodes in Table 2.3, with the exception of SNUPI, use the unlicensed UHF radio bands (*i.e.*, 315 MHz, 433 MHz, 868 MHz, 915 MHz, or 2.4 GHz). SNUPI's operating frequency, which is two orders of magnitude lower than all of the other nodes, seems very out of place. SNUPI's use of the HF ISM bands comes from the properties of the power line coupled wireless channel and is discussed in detail in Chapter 3.

Computation There is also an interesting trend in the CPUs used on the nodes in Table 2.3. There are three popular processor families: the ARM, AVR, and MSP430. From earliest to the most recent nodes, the ARM and AVR CPU architectures keep appearing. The ARM is used on nodes that need more processing power, and the Atmel AVR is used when low power consumption is more important. More recently, the TI MSP430 microcontroller has replaced the AVR in many nodes due to its ultra-low-power consumption. Like several other nodes, SNUPI uses an MSP430 as described in detail in Chapter 5.

Operating System The only trend in the embedded operating systems (OS) used on wireless sensor nodes is the prevalence of TinyOS. TinyOS was developed at the University of California Berkeley as an operating system specifically for their Mote series of nodes [89, 125]. Over time, the TinyOS platform has expanded and is now available on a wide variety of hardware architectures. SNUPI on the other hand, does not have an operating system, and instead has custom, application-specific firmware. This application-

[17] Now part of Texas Instruments

specific customization allows SNUPI to operate at lower power by removing all of the general purpose code in an OS.

Power Consumption The final trend to discuss is the power consumption of the nodes. In the early years, researchers were more concerned with getting something that worked robustly than with power optimizations. As a result, nodes like the WINS and μAMPS consume more than 1 W while transmitting, and consume 10–500 mW when in sleep. The Berkeley Motes consumed significantly less power (*i.e.*, 10s of mW while active, and 10s of μW while in sleep). Over the last 15 years, nothing major has really changed from a power point of view. The newest nodes still consume 10s of mW when active and 10s of μW while in sleep. Many of these newer nodes have significantly more capabilities than the early nodes, but their power consumption, and thus their lifetime, has not been significantly improved. For over a decade, WSN researchers have written visionary papers in which they dream of sensor nodes which consume on the order of 10–100 μW [105, 175, 176]. In particular, the SmartDust [105] and PicoRadio [176] projects at the University of California, Berkeley have been working for decades to make these nodes a reality. Despite major advances in integrated circuit design and fabrication, these nodes still have not become a reality. SNUPI however, *is* beginning to operate within the realm of these dreams—an entire order of magnitude lower power than the other nodes. SNUPI's power consumption is on the order of 1 mW when transmitting and only 1 μW during sleep.

Energy Harvesting As the power consumption of a sensor nodes drops, it becomes more and more feasible to replace the battery on the node with a harvested energy source. Energy harvesting or scavenging is a method by which the node converts ambient potential energy into electrical power. Over the past decade, there has been significant research on wireless sensor nodes which use a harvested energy source either in combination with or in place of a battery [76, 139, 157, 176, 182, 183, 195, 231]. The most popular source of harvested energy is photovoltaic (*i.e.*, solar) energy [8,23,45,78,106,117,177,179,200,233,235,253].

In addition to solar energy, WSNs have been powered using temperature gradients [92, 123, 124], vibrations [183, 243], human movement [158, 159], low-frequency electromagnetic energy [126, 246], and high-frequency RF energy [26, 27, 80, 130, 150, 162, 169, 185, 187, 188, 203, 204, 217, 218, 221, 249, 250]. Air flow, liquid flow, and ambient pressure changes have also been suggested as potential power sources [182, 183].

Backscatter Communication Of these harvested energy sources, RF energy is of particular interest because backscatter communication can be used to send data in addition to providing power. Backscatter communication is the operating principle behind passive Radio Frequency Identification (RFID). RFID has been widely commercialized as a wireless method for sending an ID number, and has been applied to personal identification badges, banking cards, and asset tracking (*i.e.*, as a bar-code replacement) among other things. In each of these systems, the device containing the ID number is called a *tag* and can be completely powerless (*i.e.*, it does not have a battery or other power source). In order for this system to work, an RFID *reader* is used to interrogate the tags by sending a high power RF signal. The tags harvest RF energy from the interrogation signal and then communicate data back to the reader by modulating the reflection of the RF interrogation signal. The modulation of the reflected signal can be done at very low power, by simply changing the impedance of the antenna on the tag.

With recent advances in ultra-low-power electronics, researchers have made wireless sensor nodes that are completely powered from RFID readers and communicate their data over the RFID backscatter communication channel [26, 27, 169, 185, 187, 188, 203, 204, 236, 249, 250]. By moving beyond the RFID standard, researchers have achieved even higher performance from self-powered sensor nodes that use backscatter communication [80, 217, 218, 221, 222]. Although these backscatter communication systems allow ultra-low-power nodes which can run indefinitely without the need for batteries, they require significant instrumentation of the sensing area in order to have sufficient RF coverage for energy harvesting. In other words, backscatter readers are required to be placed everywhere

around the sensing region. In addition, these readers transmit RF energy at very high power, thus making the total system power consumption much higher than traditional WSNs, despite the fact that the power requirements of the sensors themselves are orders of magnitude lower. Most recently, backscatter communication has been shown to work using ambient RF sources (*e.g.*, the nearby HDTV tower) rather than high-power RF readers placed in the sensing region [80, 130].

2.2 Communication over the Power Line

The power lines inside of homes and buildings were designed only to carry high voltage AC (*i.e.*, 50 or 60 Hz, depending on the country) signals for power distribution. However, they can also be used to communicate data using higher frequency signals. The power lines have been used both as a medium for these high frequency signals (*i.e.*, a transmission line) and as an antenna to both transmit and receive high frequency signals. The following sections discuss the power line's ability to operate in each of these modes.

2.2.1 Power Line as a Transmission Line

Power line communication (PLC) is a common method for sending data over a power line while simultaneously using the line for AC power distribution. In a PLC network, both transmitters and receivers are connected directly to the power line and communicate their data directly over the line. High-pass filters are used to block the high power, low frequency 50 or 60 Hz signals also present on the power line. PLC has been used most in the home automation community, which regularly uses the PLC standards X10 and Insteon. The X10[18] standard has been around since the 1970s and communicates using a 120 kHz carrier frequency. Insteon[19] is a newer standard meant to replace X10, and communicates using a 131.65 kHz carrier frequency. X10 and Insteon have very low data rates (*i.e.*, typically less than 1 kb/s in practice) and are thus only useful for home automation tasks which only

[18] http://www.x10.com

[19] http://www.insteon.net

require infrequent messages about appliance state changes.

A high data rate version of PLC is now available called broadband over power lines (BPL). BPL systems use the same principle as the earlier X10 and Insteon technologies, but they use a much higher carrier frequency and much more bandwidth. The HomePlug[20] standard uses BPL over a frequency range of 2–28 MHz, and achieves data rates up to 200 Mb/s. Many companies are now producing HomePlug-compliant transceivers, and products are being sold for home automation, audio and video streaming, and Ethernet over the power line.

2.2.2 Power Line as a Transmitting Antenna

It has been known for almost a century that the power lines surrounding us in all of our buildings can also be used as antennas. Carrier current is a method for using the power lines as a transmitting antenna for low-power AM broadcasts in the MF range (*i.e.*, 300 kHz–3 MHz). At low transmit power, a carrier current system can cover an entire building or group of buildings. It is therefore ideal for localized radio, and was popular for college and high school radio stations [135].

Ubicomp researchers have also taken advantage of the power line as a transmitting antenna. The Power Line Positioning (PLP) system, which tracks the location of small active tags throughout a home, makes use of the home power line as a transmitting antenna for signals between 447 kHz and 20 MHz [167,210,211]. In this work, a signal was injected into the power line so that it would be transmitted; however, other work has taken advantage of existing signals on the power line. The power line also transmits the electromagnetic interference (EMI) or noise signals that are present on the line. This emitted noise has been used for detecting the presence of objects and people, as well as, the position and actions of people inside the building [42, 43, 166].

---

[20] HomePlug Alliance, `http://www.homeplug.org`

2.2.3 Power Line as a Receiving Antenna

After completing the Power Line Position system which uses the power line as a transmitting antenna, Stuntebeck *et al.* postulated that the power line could also be used as a receiving antenna. Indeed, both medium wave AM radio broadcasts and VHF FM radio broadcasts can be demodulated and heard using the power line as a receiving antenna. In fact, many inexpensive AM and FM radios use their own power cord as the antenna [69, 223]. Stuntebeck *et al.* also found that the residential power line can be used to receive communications signals [210]. Stuntebeck *et al.* refer to this concept as "using the power line as a receiving antenna." I will instead name this new communications channel the *power line coupled wireless* (PLCW) channel. Since the communications signals are transfered from the node to the power line in the near-field, it does not seem appropriate to consider the power line an antenna. Instead, these near-field signals are *coupled* onto the power line. It would also be appropriate to call the channel used by PLC systems as "power line coupled"; however, the SNUPI channel is a *wirelessly* coupled channel rather than being directly coupled. For this reason, I have added the word "wireless" to yield the name: "power line coupled wireless" channel.

Stuntebeck *et al.* had two major findings regarding the power line coupled wireless channel. The first is related to the frequencies of operation for the channel, and the second is regarding the relationship between the range and power efficiency of the channel. The following sections describe these findings. I will present a more in-depth analysis of the PLCW channel in Chapter 3.

PLCW Channel Operating Frequency

Stuntebeck *et al.* considered all of the unlicensed ISM bands up to 2.45 GHz for the PLCW channel: 6.78, 13.56, 27.12, 40.68, 433.92, 915.00 MHz, and 2.45 GHz. 433.92 MHz and 915.00 MHz were immediately eliminated because they are not globally available, and are heavily used in regions where they are available. The HomePlug BPL standard uses three of the unlicensed bands: 6.78, 13.56, and 27.12 MHz. Stuntebeck *et al.* therefore tested

for interference at those frequencies due to HomePlug by measuring the power level on the power lines at several locations in a home using a spectrum analyzer. They found that HomePlug devices generate significant interference at 6.78 and 13.56 MHz, yet no interference is seen at 27.12 MHz. Due to the growing popularity of HomePlug, they eliminated 6.78 and 13.56 MHz as options for the PLCW channel. For the remaining frequency bands, they conducted experiments in which they injected signals into the powerline at 27.12, 40.68 MHz, and 2.45 GHz using a signal generator connected to a custom power line interface, and measured the signal levels at various locations of a 3 floor, 371 square meter home using a spectrum analyzer. They concluded that 27.12 and 40.68 MHz signals travel through the power line with little attenuation in comparison to 2.45 GHz signals [210].

The analysis by Stuntebeck *et al.* found that both the 27.12 MHz and 40.68 MHz ISM bands would work well for a PLCW channel. I have repeated the analysis done by Stuntebeck *et al.* in Chapter 3 and found similar results. A more detailed analysis and frequency selection is described in Chapter 3.

PLCW Channel Range/Power Relationship

Stuntebeck *et al.* found that use of the PLCW channel allowed the sensor network to have "whole-home" range, even when using a very low transmit power. For testing purposes, they transmitted a 27.12 MHz signal at −35 dBm using a commercial 27.12 MHz Citizens' Band antenna. Using a hand-held radio scanner and a "rubber-ducky" antenna they found that the transmitted signal was too weak to pick up when the receiver was outside of the same room as the transmitter. However, when the receiver was connected to the power line using a custom interface, a signal could be received even as the transmitter was moved all around the house. With a transmit power of −35 dBm, they found that they could pick up a signal on the receiver as long as the transmitter was within 60 cm from an outlet, wall switch, or other electrical wiring [210]. Clearly, the PLCW channel allowed them to significantly extend the range of their 27.12 MHz transmitter.

Stuntebeck *et al.* also compared the range and power efficiency of their node to existing

commercial ZigBee nodes. Stuntebeck *et al.* built a simple 27.12 MHz wireless node to compare to the Sun SunSPOT and the Crossbow MicaZ. They found that the ZigBee nodes only had an indoor range of about one room when their transmit power was −25 dBm. In addition, their 2.4 GHz signals did not see any improvement from using the PLCW channel. They also calculated the energy consumed per useful bit for all nodes, and found that their 27.12 MHz PLCW node consumed 2–7% of the energy of the ZigBee nodes when achieving similar range [210]. This is the key finding that use of the PLCW channel in the HF range overcomes the traditional power/range trade-off limitation.

Chapter 3

POWER LINE COUPLED WIRELESS CHANNEL

As the name SNUPI (*i.e.*, Sensor Network Utilizing Powerline Infrastructure) implies, the use of the power line infrastructure is a key element in any SNUPI network. In fact it is the use of the power line infrastructure as a communications channel that sets SNUPI apart from all other sensor networks. All SNUPI networks have a single base station receiver that is connected directly to the power line (*i.e.*, plugged into an outlet). Each node in the sensor network transmits wireless signals that couple to nearby power lines, thus creating signals that travel through the power line infrastructure to the base station receiver. In this way, the sensor nodes can transmit at much lower power because signals do not need to propagate over-the-air for the entire path to the base station receiver; they only need to propagate to the nearest power line. This method of using the power line as a receiving antenna for communications was first introduced by Stuntebeck *et al.* in 2010 [44, 210].

Stuntebeck *et al.* found that the power line infrastructure of a home can be used as a receiving antenna to pick up wireless signals produced by low-power sensor nodes in different locations around the home. They also showed that such a network overcomes the traditional power/range trade-off governing traditional over-the-air WSNs. In other words, a sensor node which couples its wireless signals to a power line can consume less power, while still obtaining the same range inside a home as a traditional over-the-air sensor node [44, 210]. This phenomenon, which Stuntebeck *et al.* referred to as "using the power line as a receiving antenna", is really just a new wireless communications channel. I will call this new communications channel the **power line coupled wireless (PLCW)** channel. Since the communications signals are transfered from the node to the power line in the near-field, it does not seem appropriate to consider the power line an antenna. Instead,

these near-field signals are *coupled* onto the power line. It would also be appropriate to call the channel used by power line communications (PLC) systems as "power line coupled"; however, the SNUPI channel is a *wirelessly* coupled channel rather than being directly coupled. For this reason, I have added the word "wireless" to yield the name: "power line coupled wireless" channel. This chapter will study this new channel in detail.

Stuntebeck *et al.* had two major findings regarding the power line coupled wireless channel. The first is related to the frequencies of operation for the channel, and the second is regarding the relationship between the range and power efficiency of the channel. The findings of Stuntebeck *et al.* are described in Section 2.2.3, while this chapter presents a more in-depth analysis of the PLCW channel.

Since the power line coupled wireless (PLCW) channel has not been studied previously, there are a number of questions that need to be answered in order to design a practical and robust wireless sensor network that uses this channel. These questions include:

- What is the best way to model the channel's background noise?
- What is the typical path loss of the channel?
- What is best frequency of operation?
- What bandwidth does the channel support?
- What is the best way to couple wireless signals onto the power lines?
- What is the best way to extract communications signals from the power lines?

To fully study this channel would require multiple Ph.D. dissertations; however, this chapter begins to answer many of these questions. Section 3.1 describes early experiments to find the best frequencies of operation for SNUPI networks. Section 3.2 describes the data collection and analysis of the background noise in PLCW channels, while Section 3.3 describes the data collection and analysis of the path loss of a typical PLCW channel. Finally, Section 3.4 provides a summary of the important findings in this chapter.

3.1 Initial Frequency Selection

Since all of the background noise and path loss data collection and analysis described in this chapter needs to be done at every frequency of interest, we first need to limit the number of frequency bands to test. First, we'll limit our search to the unlicensed ISM bands, as described in Section 2.1.3. Stuntebeck *et al.* considered all of the unlicensed ISM bands up to 2.45 GHz for the PLCW channel: 6.78, 13.56, 27.12, 40.68, 433.92, 915.00 MHz, and 2.45 GHz. They immediately eliminated 433.92 MHz and 915.00 MHz because they are not globally available, and are heavily used in regions where they are available. Stuntebeck *et al.* tested for interference due to the HomePlug power line communications (PLC) system at 6.78, 13.56, and 27.12 MHz by measuring the power level on the power lines at several locations in a home using a spectrum analyzer. They found that HomePlug devices generate significant interference at 6.78 and 13.56 MHz, yet no interference is seen at 27.12 MHz. Due to the growing popularity of HomePlug, they eliminated 6.78 and 13.56 MHz as options for the PLCW channel. For the remaining frequency bands, they conducted experiments in which they injected signals into the powerline at 27.12, 40.68 MHz, and 2.45 GHz using a signal generator connected to a custom power line interface, and measured the signal levels at various locations of a 3 floor, 371 m^2 home using a spectrum analyzer. They concluded that 27.12 and 40.68 MHz signals travel through the power line with little attenuation in comparison to 2.45 GHz signals [210].

I repeated the experiments conducted by Stuntebeck *et al.* in several homes using all of the unlicensed ISM bands below 3 GHz: 6.78, 13.56, 27.12, 40.68, 433.92, 915.00 MHz, and 2.45 GHz. In my experiments, I used several different antennas to wirelessly transmit tones at each of these frequencies. The tones were produced by a signal generator which was powered from a battery pack and an AC inverter rather than being powered from the home's power lines. This ensured that the RF signal was not conducted onto the power line through the signal generator's power cord. A spectrum analyzer connected to the power line through a custom power line interface allowed me to test the received carrier to noise ratio at various locations around the home.

I found that the frequencies above 100 MHz did not work well, either because the RF signal did not couple well to the power line, or because they were attenuated too much inside the power line. 6.78 MHz also did not work, as the noise floor was too high. Although 27.12 MHz appeared best from these measurements, I decided to continue to consider 13.56, 27.12, and 40.68 MHz for all future analysis. Throughout this chapter, most of the data collection and analysis was therefore conducted at 13.56, 27.12, and 40.68 MHz.

3.2 Background Noise

In order to select the best frequency of operation, design a SNUPI receiver, and design an optimal communications protocol, it is crucial to understand the nature of the background noise on the channel. Due to the popularity of broadband over power lines (BPL) over the last few years, there have been a number of academic papers studying the noise on the power lines between 1–30 MHz [11, 144]. This section builds on this work by looking specifically at the narrow-band SNUPI channel at 13.56, 27.12, and 40.68 MHz. The goal is to understand the typical noise level as well as the limits of this noise. This is crucial for designing the receiver and making decisions about which frequency of operation is best, and which power line connections are best. Section 3.2.1 describes the data collection of background noise in 15 homes, while Section 3.2.2 analyzes the data collected.

3.2.1 Noise Data Collection

In order to analyze the background noise, I setup an experiment to collect background noise data in 15 homes around the greater Seattle area. Multiple days of continuous data would need to be collected in each home at 13.56, 27.12, and 40.68 MHz. Since crystals are more commonly available at 27.00 MHz rather than 27.12 MHz, and since 27.00 MHz is still inside the ISM band, data was also collected at that frequency. Since there are three possible connections to the power line: line-to-neutral (LN), line-to-ground (LG), and neutral-to-ground (NG), my data collection hardware simultaneously captures data from all three possible connections. The hardware consists of (1) a custom made power line interface,

Figure 3.1: Hardware used for background noise data collection, including a custom made power line interface, two software defined radios (SDR), and a data collection laptop.

(2) two software defined radios (SDR), and (3) a data collection laptop, as shown in Figure 3.1. The custom made power line interface, described in Section 6.1, filters out the 60 Hz component and its harmonics, lets the RF signals pass through, can enable measurements from all three connections to the power line simultaneously. For the SDRs, I used the Ettus Research USRP2[1]. Two USRP2s are needed in order to simultaneously capture from 3 independent channels (*i.e.*, corresponding to the three power line connections: LN, LG, and NG). The laptop processes the streaming data in real-time and logs statistics about the data to file.

The USRP2s collect data at 250 kS/s, corresponding to a 250 kHz bandwidth around each of the center frequencies of interest: 13.56, 27.12, and 40.68 MHz. The hardware cannot simultaneously collect data from each of those frequency bands, so they are collected in sequence. Data is collected on a single frequency for an entire day (*i.e.*, 24 hours), and then it switches to the next frequency. Three simultaneous channels of 32-bit floating point complex data at 250 kS/s corresponds to 5.72 MB/s of data, or 482.8 GB/day! This is way too much data to simply log the raw samples to disk for post processing, so the data needs to be processed in real-time and only the statistics on the data are logged to file.

[1]Ettus Research Universal Software Radio Peripheral 2, `https://www.ettus.com/product/category/USRP_Bus_Series`

The real-time processing computes the power spectral density (PSD) estimate using a $32,768$-point short-time Fourier transform (STFT) with $32,768$-point Hamming window and 50% overlap. This results in 66 ms resolution in the logged PSD data. Since it takes too much disk space to log the PSD itself to file, statistics on the PSD are logged instead. These statistics are taken over different bandwidths around the center frequency: 50 Hz, 100 Hz, 500 Hz, 1 kHz, 5 kHz, 10 kHz, and 50 kHz. This allows analysis over different bandwidths to see which bandwidth works best on the PLCW channel. Statistics are taken both on the instantaneous PSD estimate, and on a time-averaged PSD estimate. The instantaneous PSD is best for capturing the peaks in the PSD, which are useful for analyzing the interferrors on the channel, and the time-averaged PSD is best for capturing the background noise floor. A number of statistics are captured about the PSD for each 66 ms time window. These statistics include the mean, variance, 25^{th} percentile, median, 75^{th} percentile, 90^{th} percentile, and the maximum value. All of these statistics represent 2.06 GB/day of data. In total, I collected 123 days of data from the 15 different homes listed in Table 3.1.

It is very important to get a wide variety of different homes to get a sense for the breadth of noise environments present. Table 3.1 gives the demographic information for the 15 homes in which data was collected. Since we are also interested in the effect that power line communication (PLC) devices have on the background noise, the types of PLC devices present in each house is also noted in the table.

Data Collection Implementation Details

There are some subtleties with the USRP-based data collection. First, we need to make sure that the noise data being collected is calibrated. Since the USRP is not a calibrated instrument, extra care needs to be taken. To test the calibration of the USRP, a calibrated signal generator was used to perform a frequency sweep across the input band with a fixed amplitude. The power spectral density (PSD) of the values received using the USRP are shown in Figure 3.2 (*top*), with a zoomed in plot on the right side. As shown in the plots, the signal power is not flat as would be expected. This is because the anti-aliasing filter

Home	Type	Size (ft^2)	# Floors	Days of Data	# Tx Loc.	PLC Devices
1	Condo	1,550	1	9	11	–
2	Home	3,000	2	9	16	–
3	Home	3,000	2	10	17	–
4	Home	2,900	2	9	15	HomePlug, TED, X10
5	Home	3,400	2	7	16	–
6	Home	2,600	2	7	16	–
7	Condo	750	1	6	7	Atheros, Belkin HD
8	Condo	960	1	7	9	–
9	Home	14,000	4	7	18	–
10	Apt.	595	1	9	8	–
11	Home	4,000	3	9	17	TED, Altenergy Power Inv.
12	Home	1,800	3	6	12	Telephone & Networking
13	Apt.	900	1	10	7	–
14	Apt.	750	1	8	5	–
15	Home	3,150	2	10	19	–

Table 3.1: Demographic data for homes used in background noise and path loss data collection. "# Tx Loc." refers to the number of transmitter locations tested in the path loss experiments.
Home – Single Family Home
Condo – Condominium in Commercial Building
Apt. – Apartment Building

inside the USRP attenuates the signal at the edges of the band. Due to this fall-off, the frequencies of interest must be kept toward the center of the band (*i.e.*, closest to the center frequency tuned into the USRP). In addition to the flatness issues, the absolute value of the signal recorded by the USRP is 10 dB low, so a $+10$ dB calibration factor needs to be added to incoming data.

In addition to the calibration factor and the fall-off at the edge of the band, there is also an issue at the center of the band. Figure 3.2 (*bottom*) shows the PSD when a white noise source is applied. In this case, there is a non-linearity at the center of the band, and therefore measurements need to be taken away from the center. As a result of this, the USRP was tuned to a center frequency 60 kHz offset from the center of our frequency band of interest. To clarify, the center frequencies of 13.50, 27.06, and 40.62 MHz were used to capture noise in frequency bands of interest centered at the 13.56, 27.12, and 40.68 MHz. It should also be noted that our other frequency of interest 12.00 MHz falls 60 kHz below the USRP center frequency of 27.06 MHz. Therefore by setting the USRP to a center frequency of 27.06 MHz, the 27.12 MHz band is found at $+60$ kHz, and the 27.00 MHz band is found at -60 kHz. The largest bandwidth that we are considering is 50 kHz wide, and is represented by the green shaded regions on the plots in Figure 3.2. From these plots, it can be seen that the signal flatness is within 0.3 dB, and the noise flatness is within 1 dB. In other words, the noise data collected can be compared to within 1 dB.

Once the calibration of the USRP is known, we can setup the data collection software. We collect a full 24 hours of data using a single center frequency tuned into the USRP, and cycle through three center frequencies: 13.50, 27.06, and 40.62 MHz. This allows us to capture statistics around either 13.56, 27.00 and 27.12, or 40.68 MHz. Note that we can capture noise statistics around both 27.00 and 27.12 MHz when using a USRP center frequency of 27.06 MHz. Since each of these center frequencies are collected for 24 hours, it takes 6 full days to collect 2 days of data at each frequency of interest. As seen in Table 3.1, data was collected between 6 and 10 days in each of 15 homes, representing a total of 123 days of background noise data.

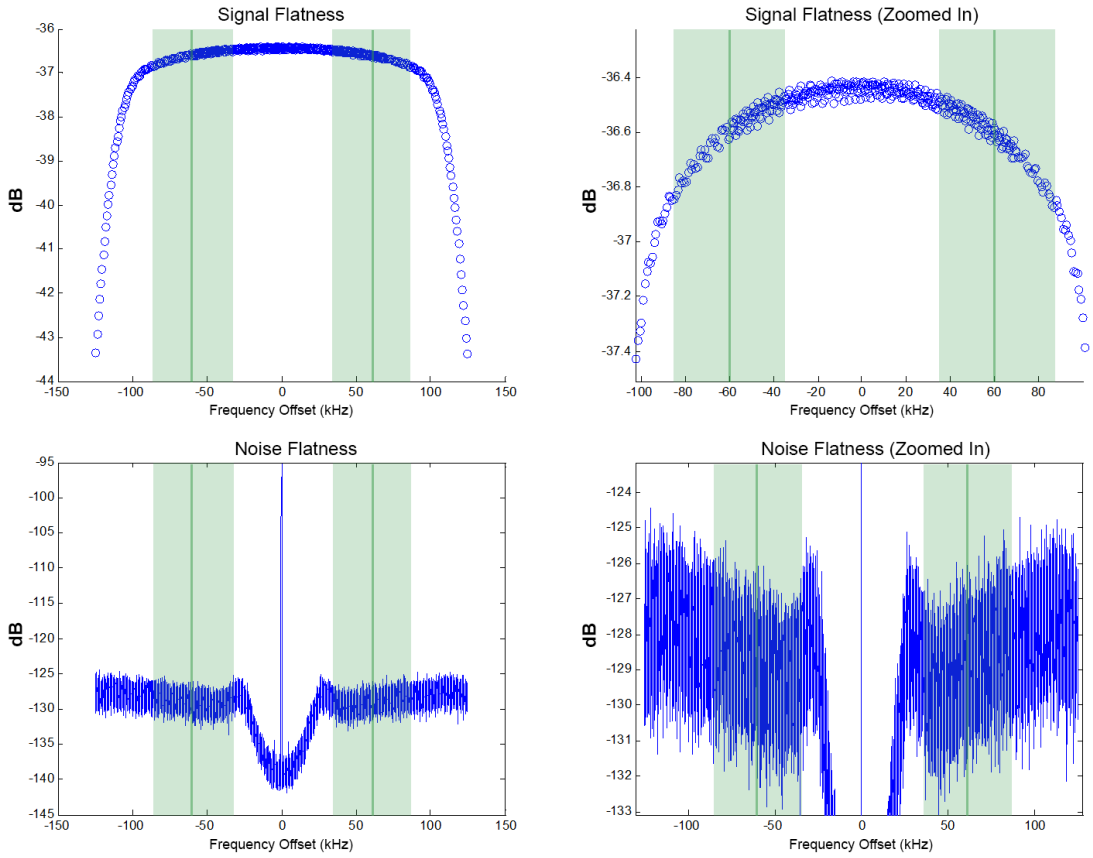

Figure 3.2: USRP calibration curves, showing non-linearities both at the center of the band and at the band edges. The signal flatness curves (*top*) were generated using a constant amplitude frequency sweep input, and the noise flatness curves (*bottom*) were generated using a white noise source input. The x-axes represents the offset from 27.06 MHz; however, the same behavior was also seen at 13.50 and 40.62 MHz. The green bands represent the frequency bands of interest for the data collection, and the green lines represent the center of those bands (*e.g.*, 27.00 and 27.12 MHz).

Target BW	# PSD Bins	Actual BW
50 Hz	7	53.4 Hz
100 Hz	13	99.2 Hz
500 Hz	66	503.5 Hz
1,000 Hz	131	999.5 Hz
5,000 Hz	655	4,997.3 Hz
10,000 Hz	1311	10,002.1 Hz
50,000 Hz	6554	50,003.1 Hz

Table 3.2: Noise bandwidths in which statistics are computed over. The target bandwidth, number of PSD bins used, and actual bandwidth are shown.

Since it is not practical to capture this much raw data, only statistics about the noise can be captured. This means that the raw data needs to be processed in real-time during the data collection. In order to do this, custom signal processing code was written in C++ to process the data in real-time while streaming raw samples in from the USRPs. All statistics were computed from the power spectral density (PSD), which was computed by taking a 32,768-point short-time Fourier transform (STFT) with 32,768-point Hamming window and 50% overlap. With the 250 kS/s raw data, this corresponds to 7.63 Hz wide frequency bins, and the main lobe of the Hamming window is about 32 kHz wide. Therefore, the practical limit on the frequency resolution of the PSD is 32 kHz. The time resolution of the statistics are 66 ms.

Over each of these 66 ms windows, we want to compute a number of statistics about the noise. However, these noise statistics may vary when taken over different bandwidths around the center frequencies of interest: 13.56, 27.00, 27.12, and 40.68 MHz. Therefore, statistics are computed around the following bandwidths: 50 Hz, 100 Hz, 500 Hz, 1 kHz, 5 kHz, 10 kHz, and 50 kHz, as shown in Table 3.2. To do this, the statistics were computed using data from some number of frequency bins in the PSD. Table 3.2 shows the number of bins used for each target bandwidth, and actual bandwidth that each corresponds to.

The data being collected needs to help us understand the noise floor was well as the interferors that are present on the channel. As a result, different statistics are used to compute each. In order to compute the noise floor statistics, 16 consecutive PSDs are averaged using Welch's method. This average represents 1.06 seconds of noise data, and gives an accurate estimate of the noise PSD. Then the following six statistics are computed at each 66 ms window over each of the bandwidths listed in Table 3.2:

1. **mean** over the bandwidth
2. **variance** over the bandwidth
3. **25$^{\text{th}}$ percentile** over the bandwidth
4. **median** (*i.e.*, 50$^{\text{th}}$ percentile) over the bandwidth
5. **75$^{\text{th}}$ percentile** over the bandwidth

These statistics allow us to understand the level and variation in the noise floor over time.

In order to compute the statistics related to the interferors, we care more about the instantaneous peaks rather than the noise floor. As a result, statistics are computed on each 66 ms PSD without taking any averages over time. Then the following six statistics are computed at each 66 ms window over each of the bandwidths listed in Table 3.2:

1. **maximum** value over the bandwidth
2. **90$^{\text{th}}$ percentile** over the bandwidth
3. **75$^{\text{th}}$ percentile** over the bandwidth
4. **mean** over the bandwidth
5. **variance** over the bandwidth

These statistics allow us to understand the amplitude and prevalence of interference above the noise floor.

In addition to the noise statistics, it is also useful to capture raw samples from the channel in order to (1) develop a noise model, (2) to use the background noise data in channel simulations, and (3) explore interference bursts less than 66 ms in duration. As mentioned previously, it would take 482.8 GB/day to capture this data which is not feasible for the entire dataset. Instead, I went back to 4 homes after completing the primary background

noise data collection, and captured raw samples on only the line-to-neutral (LN) and line-to-ground (LG) power line connections. In addition, I only performed this data collection at 13.56, 27.00, and 27.12 MHz, and only captured 1 day of data per frequency. The homes chosen for this raw data collection were those which represented low, normal, and abnormally high noise based on the analysis of the initial background noise data collection. In total, 1.93 TB of raw data samples were collected in 6 days from 4 homes.

3.2.2 Noise Analysis

The primary goal of the noise analysis is to determine what the is the typical noise level at each of the frequencies of interest, and with each of the three possible power line connections (*i.e.*, LN, LG, and NG). To explore this question, we'll first plot histograms of some of the noise statistics calculated. The noise statistics shown in the histograms are:

1. **avgMean** - the mean over a 10 kHz bandwidth of the 1 s time averaged noise data
2. **instMean** - the mean over a 10 kHz bandwidth of the noise in each 66 ms window of noise data
3. **avg75pct** - the 75^{th} percentile over a 10 kHz bandwidth of the 1 s time averaged noise data
4. **inst75pct** - the 75^{th} percentile over a 10 kHz bandwidth of the noise in each 66 ms window of noise data
5. **inst90pct** - the 90^{th} percentile over a 10 kHz bandwidth of the noise in each 66 ms window of noise data
6. **instMax** - the maximum value over a 10 kHz bandwidth of the noise in each 66 ms window of noise data

The avgMean metric is best measure of the noise floor seen at the input of the receiver, while the instMax is the best measure of the maximum interference present at the input of the receiver. In the interest of space, I've highlighted only a few of these histogram plots to show the variety of noise present in the dataset. All histogram plots in Figures 3.3–3.8 show the histograms of these 6 metrics plotted on the same x-axis between -119 and

-85 dBm/Hz. Using this same x-axis therefore makes it easier to compare the plots. There are 9 curves plotted for each metric, representing different frequencies and power line connections. The color indicates the frequency, and the line style represents the connection to the power line.

Figure 3.3 shows fairly typical noise, which was seen in many homes. Notice that the noise floor is different for each frequency, but fairly similar for each of the power line connections within each frequency. The one exception is the highly varying noise level of the LN connection at 13.56 MHz. This same behavior was seen in multiple homes, but is certainly not always the case. Figure 3.4 shows a home in which the noise was generally higher than the other homes without PLC devices; however, the LN connection at 13.56 MHz as particularly low noise. Figure 3.5 shows the noise profile for a home with very low noise. Not only is the noise very low in this home, but there is very little variation in the noise over time.

It is also very important to determine the effect of power line communications (PLC) devices on the noise present to a SNUPI receiver. Some PLC devices operate at very low frequencies below 200 kHz. Homes with devices like these do not present any additional noise in the frequency bands of interest for SNUPI. H11 and H12 both had these types of devices in the home, and as shown in Figure 3.6, there was no detectable change in the noise. However, some PLC devices use the 2–28 MHz frequency range for power line communication, and therefore could present a significant problem to SNUPI networks operating in the same home. H4 and H7 both contained these types of devices, and not surprisingly, Figure 3.7 show much higher noise at 13.56 and 27.12 MHz, although no effect on 40.68 MHz.

Finally, Figure 3.8 plots the aggregate histogram of all data collected from all homes. Although these histograms are useful for visualizing the spread of the noise and comparing between different frequencies or power line connections, they are difficult to use to determine the value of the noise level. This value is useful for making more quantitative comparisons, or for designing the receiver in which the exact noise level needs to be known.

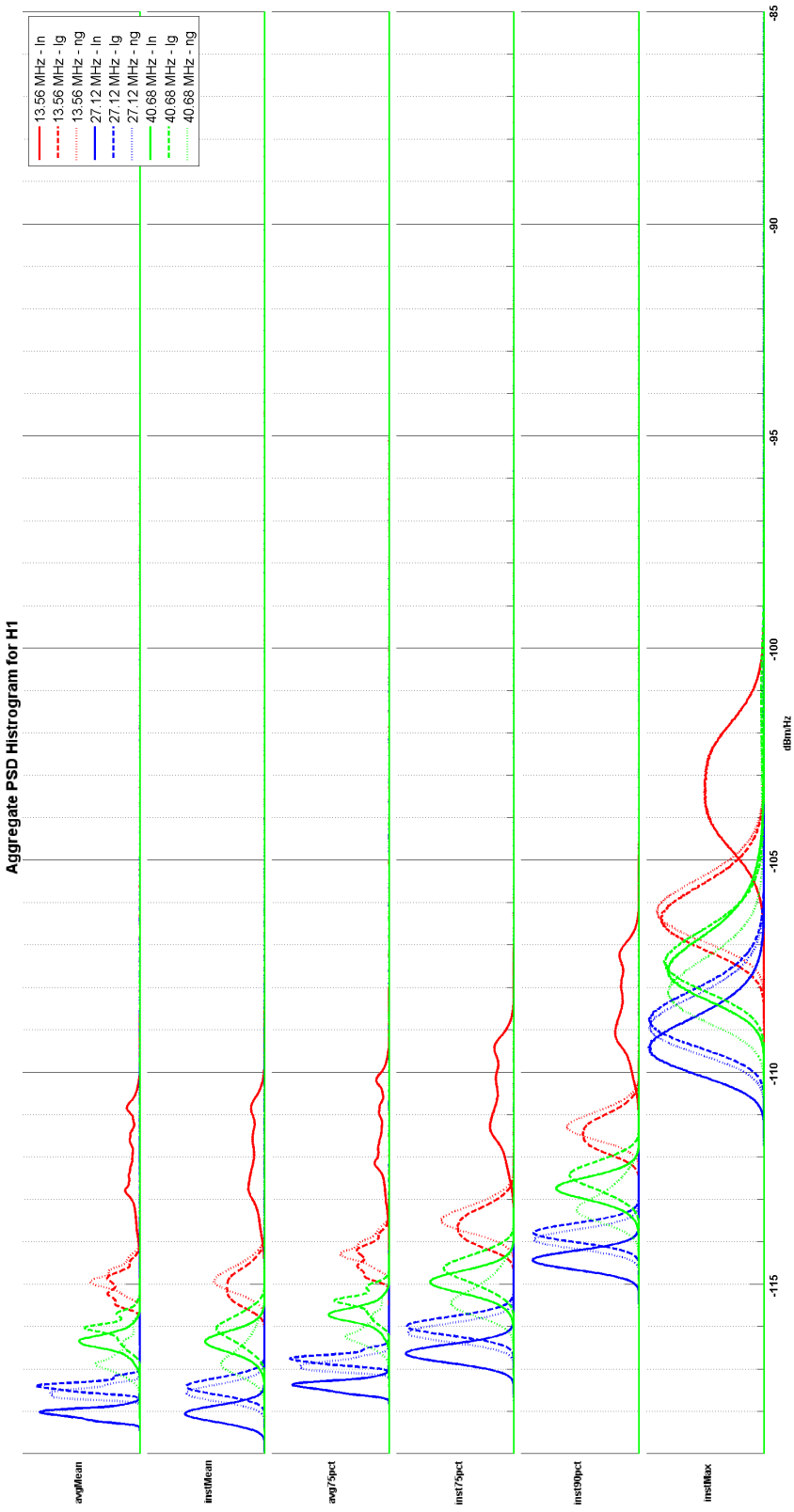

Figure 3.3: Background noise histogram from H1, showing typical noise. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

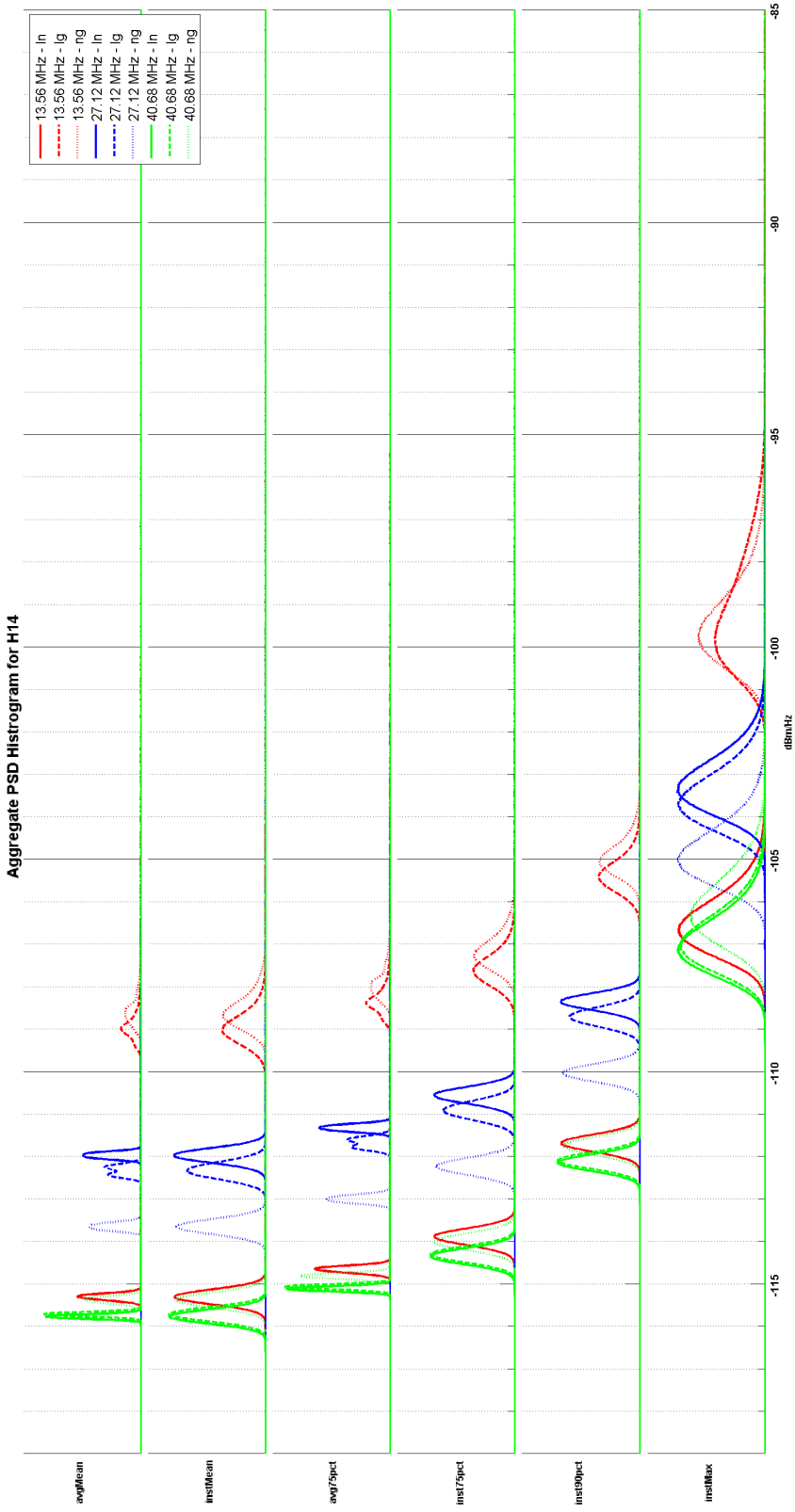

Figure 3.4: Background noise histogram from H14, showing an abnormal noise pattern. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

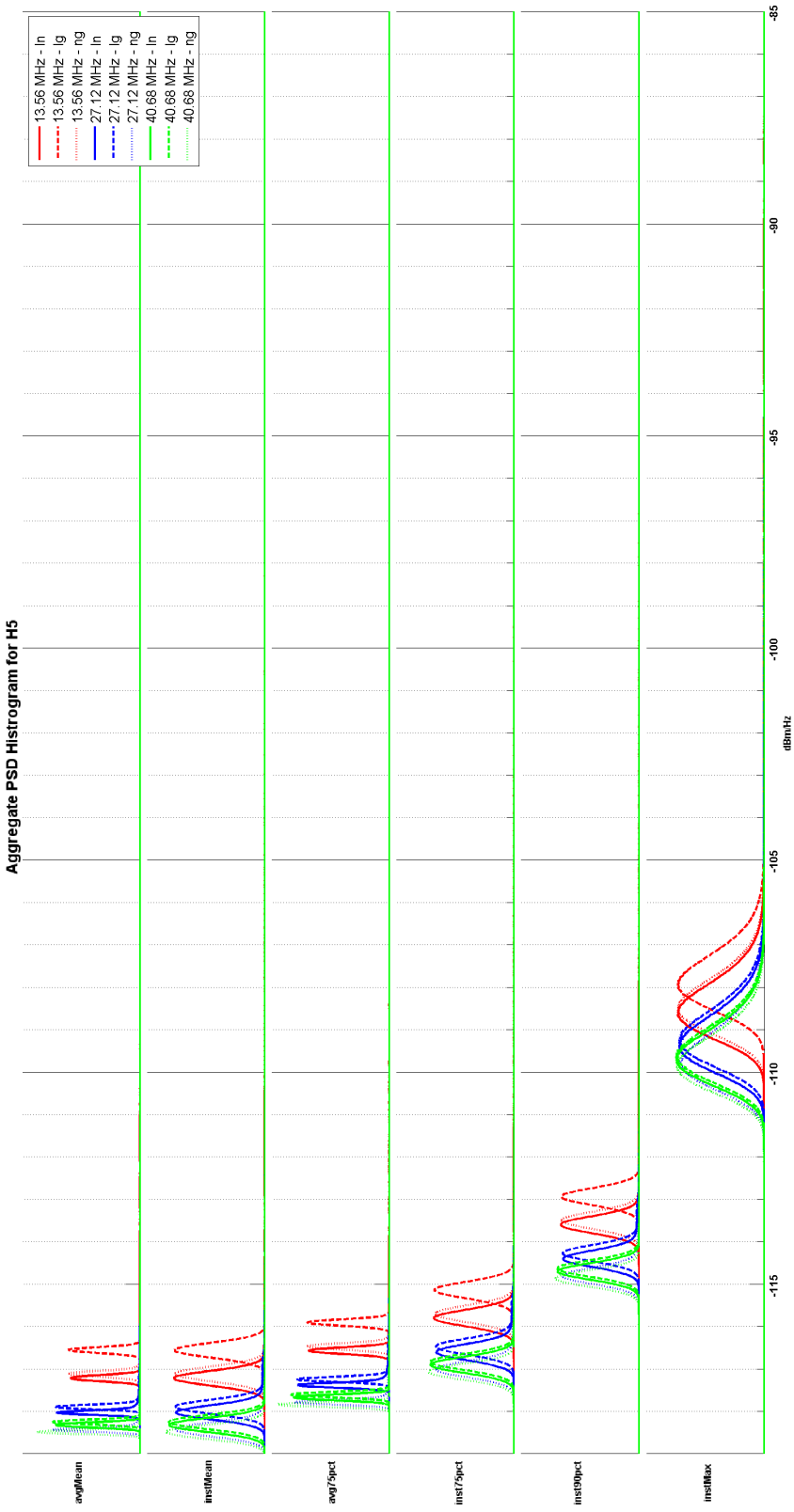

Figure 3.5: Background noise histogram from H5, showing very low noise. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

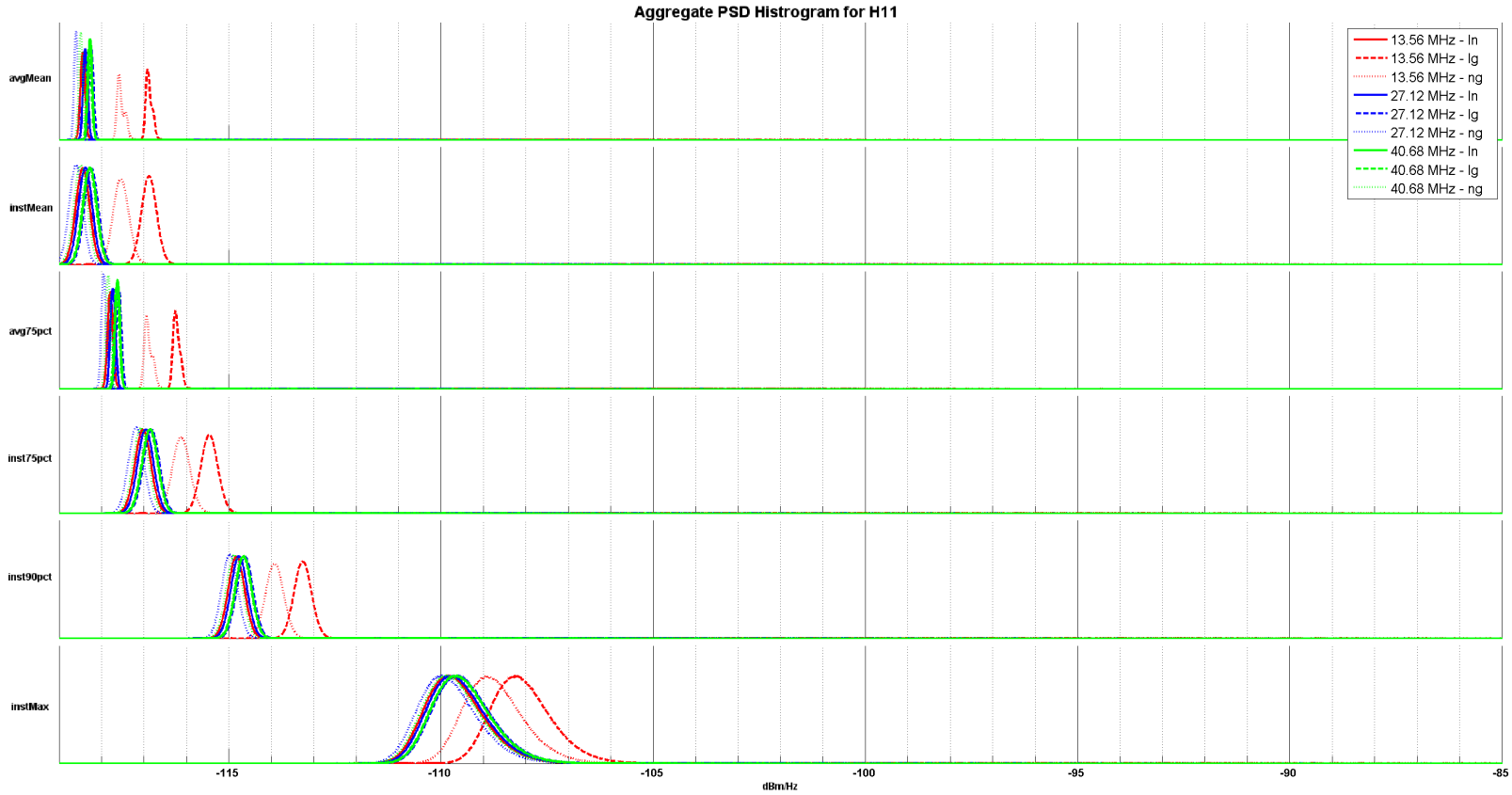

Figure 3.6: Background noise histogram from H11, showing no additional noise from low-frequency PLC devices. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

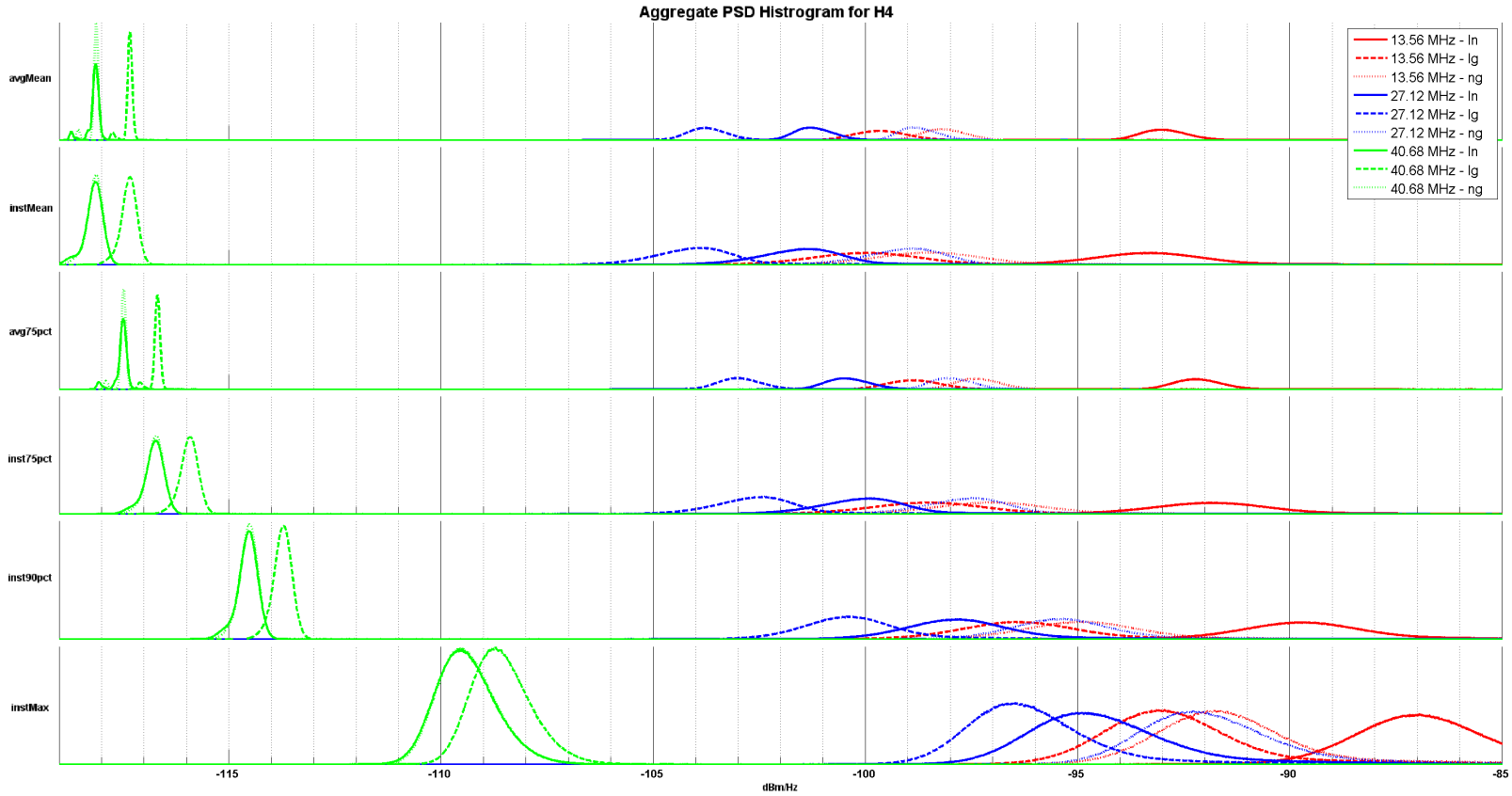

Figure 3.7: Background noise histogram from H4, showing significant additional noise from high-frequency PLC devices. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

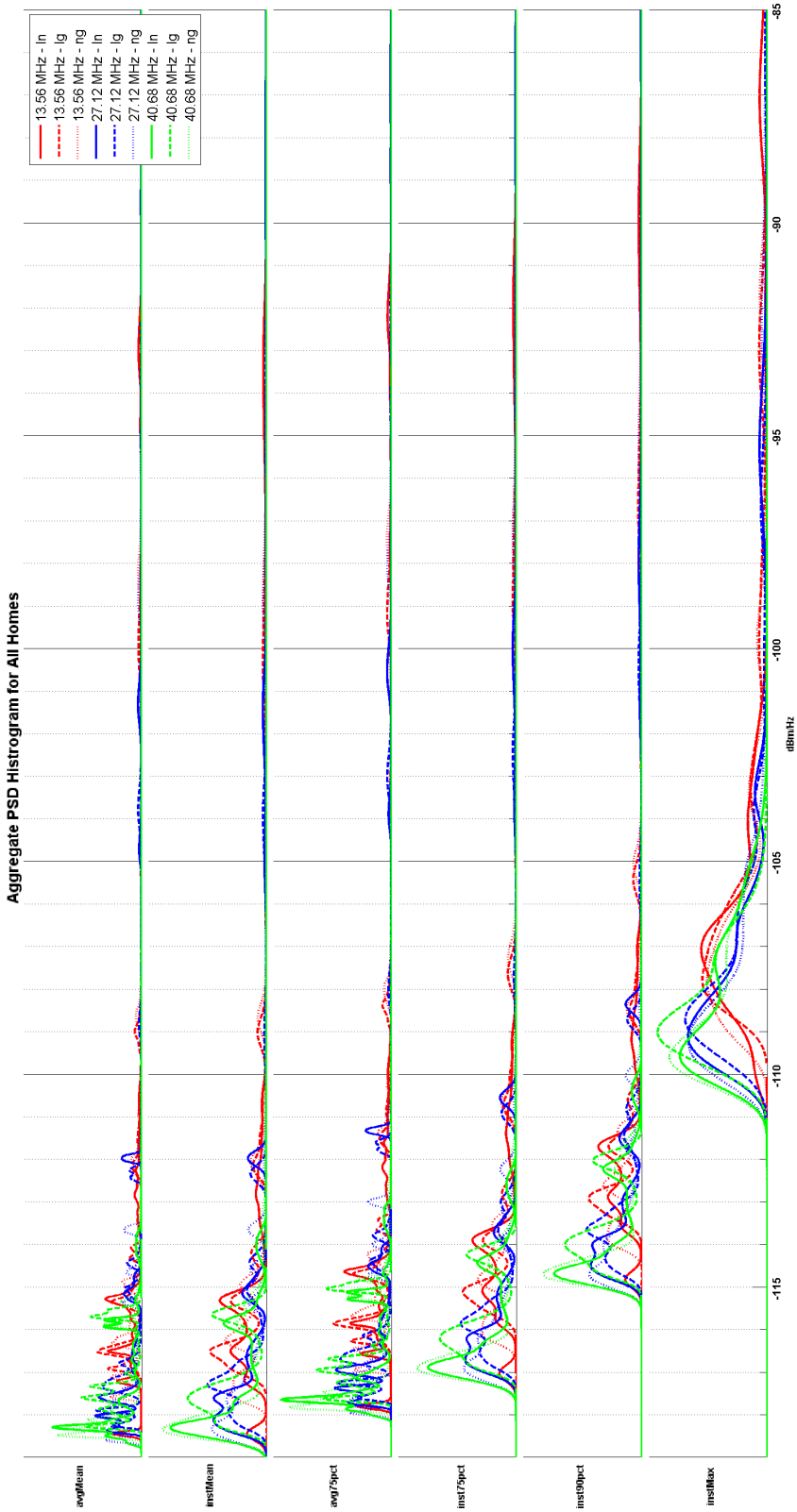

Figure 3.8: Background noise histogram aggregate from all homes. The avgMean metric is best measure of the noise floor, while the instMax is the best measure of the maximum interference present at the input of the receiver. There are 9 curves plotted for each metric, where the color indicates the frequency, and the line style represents the connection to the power line.

Frequency (MHz)	**PL Conn.**	**90% (dBm/Hz)**	**95% (dBm/Hz)**	**99% (dBm/Hz)**	**99.9% (dBm/Hz)**	**99.99% (dBm/Hz)**	**99.999% (dBm/Hz)**
13.56	LN	−98.6	−93.2	−92.1	−89.9	−79.0	−78.7
13.56	LG	−104.6	−99.8	−98.4	−96.3	−84.9	−84.6
13.56	NG	−104.1	−98.4	−97.5	−95.1	−85.1	−84.8
27.12	LN	−101.8	−101.0	−87.7	−86.2	−83.7	−82.2
27.12	LG	−104.6	−103.5	−93.3	−91.7	−89.2	−87.7
27.12	NG	−99.5	−98.5	−95.3	−93.7	−91.2	−89.7
40.68	LN	−115.5	−114.4	−113.8	−113.6	−113.5	−113.0
40.68	LG	−115.6	−115.5	−115.4	−115.3	−115.2	−114.0
40.68	NG	−115.3	−114.1	−113.2	−112.9	−112.7	−112.5

Table 3.3: Aggregate noise levels using a 10 kHz bandwidth. For each frequency and power line connection, the value represents the noise level at which the specific percentage of the noise is below that level (*i.e.*, the first column of data represents the 90^{th} percentile of the noise, while the last column is the level at which 99.999% of the noise is below the level).

LN - line-to-neutral power line connection
LG - line-to-ground power line connection
NG - neutral-to-ground power line connection

Center Frequency	Bandwidth	Max. Field Strength	Max. EIRP
13.56 MHz	14 kHz	15,848 μV/m @ 30 m	+8.77 dBm
27.12 MHz	326 kHz	10,000 μV/m @ 3 m	−15.23 dBm
40.68 MHz	40 kHz	1,000 μV/m @ 3 m	−35.23 dBm

Table 3.4: FCC power and bandwidth limits on SNUPI frequency bands [67].

However, the definition of "noise level" is not absolute. Engineers often define the noise level to be the level at which 99% of the noise is below that level. Sometimes, however they use 99.9%, or 9.99%. Table 3.3 therefore gives these levels for 90%, 95%, 99%, 99.9%, 99.99%, and 99.999%. The noise levels shown are for the aggregation of all data across all 15 homes when looking at a 10 kHz bandwidth.

From this data, we can see that the noise at 40.68 MHz is significantly lower than either 13.56 or 27.12 MHz. This is mostly due to the fact that PLC devices do not produce noise in the 40.68 MHz frequency band. Although this makes 40.68 MHz a clear winner from a background noise point of view, it is actually the least favorable ISM band due to the limited transmit power allowed by the FCC. As shown in Table 3.4, the power limit at 40.68 MHz is −35.23 dBm EIRP, while it is at least 20 dB higher for both 13.56 MHz, and 27.12 MHz. For this reason, 40.68 MHz has been eliminated for further analysis.

From Table 3.3 it can be seen that for both 13.56 MHz and 27.12 MHz the noise is lower for the ground connections (*i.e.*, LG and NG) than the common-mode LN connection. This suggests that in homes in which ground connections are available, they will have lower noise than the LN connection. It's important to note that in older homes, there is no ground connection, and therefore LN is the only option. Between 13.56 and 27.12 MHz, and looking at the 99.9% threshold and the LG power line connection, 13.56 MHz has lower noise on average, but only by about 4–5 dB.

It should be noted that all of the analysis presented here was done at a bandwidth of 10 kHz, as this is the most commonly used bandwidth for SNUPI network to date. However,

the data collection described in this section allows us to analyze the noise over several different bandwidths to determine which is best from a noise point of view. Although the data has already been collected, conducting this analysis remains future work.

3.3 Path Loss

This section studies the path loss over the power line coupled wireless (PLCW) channel. The path loss is the ratio of the transmitted power over the power received at the other end of the channel. To capture this data, I built a simple transmitter shown in Figure 3.9 which sends a continuous tone at either 13.56 or 27.00 MHz. Due to the unavailability of off-the-shelf crystal oscillators, I could not make a test transmitter at 27.12 or 40.68 MHz. Section 3.3.1 describes an experiment to determine the amount of radiated power from the test transmitter, and Section 3.3.2 describes an experiment in which the transmitter is placed at several locations around 15 different homes to measure the amount of radiated power as a function of the placement location. For this data, I can compute average path loss across all of the homes and all of the locations in each home. I will also be able to see if there are significant differences between different connections to the power line, different frequencies of interest, or different orientations of the transmitter. This analysis will therefore answer many of our questions about the new PLCW channel.

3.3.1 Radiated Power Data Collection

The first step in determining the path loss is to determine the amount of radiated power. Do to this, I measured the radiated field strength of the test transmitter shown in Figure 3.9 using the protocol defined by the FCC [67]. The test transmitter was build using the discrete digital transmitter implementation described in Section 5.3.2. To ensure that the test transmitter sends a constant amplitude signal over a long period of time, it was powered from a 5.0 V regulated supply connected to a 9 V battery, and the battery voltage was tested before each test. The antenna used for all path loss experiments is the “3x1-75-2/0.5” spiral PCB antenna described in detail in Section 5.4.1.

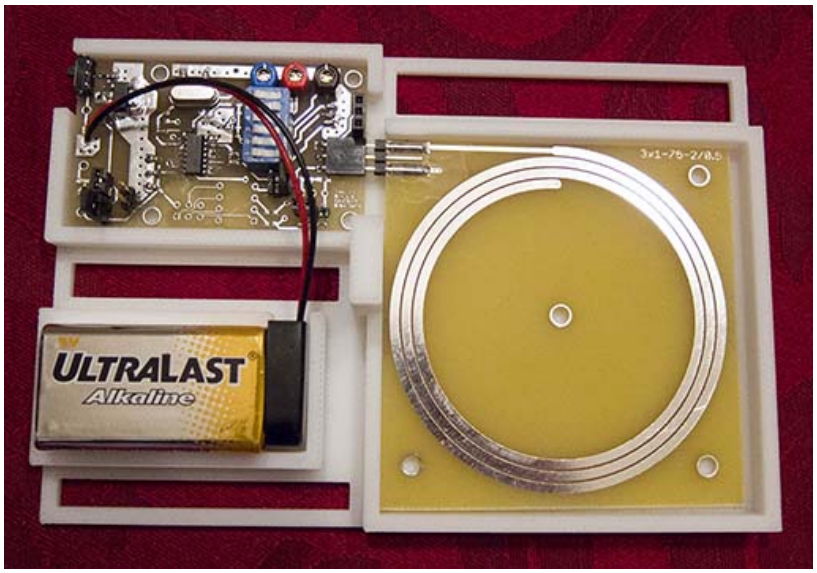

Figure 3.9: Transmitter used for path loss experiments.

Figure 3.10: Experimental setup for field strength measurements. The transmitter and receiver were placed 20 m apart (*left*) for 13.56 MHz, and 3 m apart (*right*) for 27.00 MHz. The FCC recommends 30 m of separation at 13.56 MHz, but the basement used for measurement was not large enough to support measurements at more than 20 m of separation.

To measure the radiated field strength, the test transmitter is placed at a fixed distance away from a calibrated ETS-Lindgren Model 6502 active shielded loop antenna, which is connected to a calibrated spectrum analyzer to measure the received power. The FCC suggests placing the transmitter and receiver 30 m apart at 13.56 MHz, and 3 m apart at 27.12 MHz, as shown in Figure 3.10. Care must be taken to minimize the number of metal objects around the test setup. All tests were conducted in a large open basement, and metal objects were avoided as much as possible. The receiving antenna was placed on a fiberglass ladder, and the transmitter under test was placed on a board of wood supported by two plastic sawhorses. The transmitter was held in a plastic case which allowed it to be propped up at 90° so that the plane of the transmitting loop faced the receiving loop. The transmitter was also placed on a plastic turntable which allowed it to be turned 360° during the test. The turntable was operated by the experimenter pulling on a rope, and thus allowing the operator to stay at least 3 m away from the experimental setup.

To measure the maximum field strength, the following procedure was followed:

1. Connect calibrated receiving antenna to calibrated spectrum analyzer using a calibrated cable
2. Configure spectrum analyzer to average detector with a span of 9 kHz and a reference level of -20 dBm. Also set the center frequency to either 13.56 or 27.00 MHz.
3. Place receiving antenna on top rung of ladder, and turn on.
4. Turn on transmitter and check supply voltage. Replace battery if less than 5.0 V.
5. Place transmitter 3 m away from receiving antenna.
6. Enable Max-Hold on the spectrum analyzer.
7. Spin the transmitter under test around several full revolutions by pulling on rope.
8. Read maximum received power from spectrum analyzer using the peak search function (done using network connection to analyzer to avoid the human body interfering with the experiment).
9. Repeat experiment two more times.
10. Move receiving antenna to each ladder rung as shown in Figure 3.11 and repeat (this

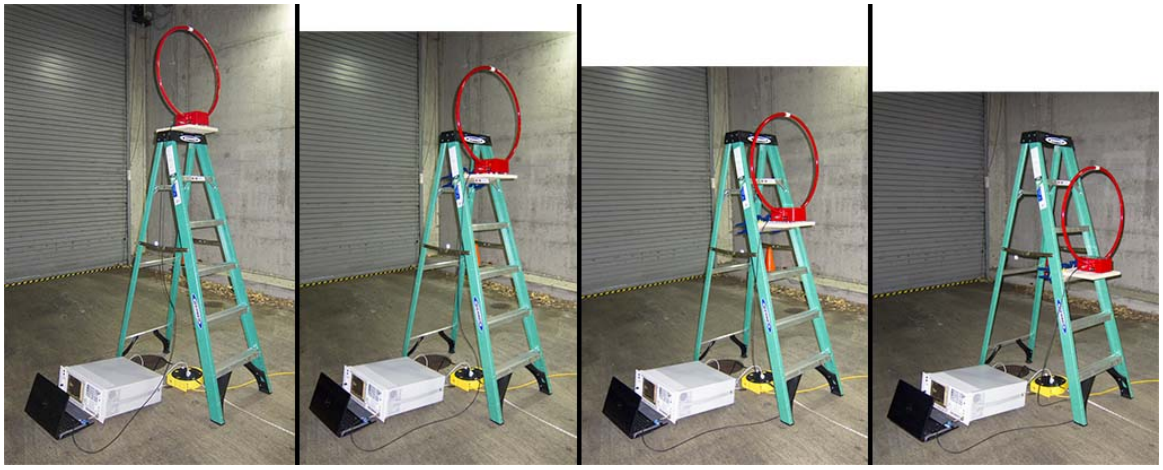

Figure 3.11: Various heights of receiving antenna used to avoid nulls in the field pattern.

is to ensure that the receiving antenna is not in a null).

11. Record the maximum reading, regardless of receiving antenna height.

When comparing the field strength measured to the FCC regulations, it's best to use the same distances as the FCC (*i.e.*, 30 m at 13.56 MHz and 3 m at 27.00 MHz); however, in order to make for a fair comparison between 13.56 and 27.00 MHz, I measured the field strengths when both were at 3 m for all analysis. The effective isotropic radiated power (EIRP) can be computed using the variables shown in Table 3.5 and the following equations:

$$V_{\text{rx}} = P_{\text{rx}} + 90 + 10\log_{10} Z_0$$

$$E_{\text{rx}} = V_{\text{rx}} + C_a + L_c$$

$$P_{\text{EIRP}} = E_{\text{rx}} - 90 + 20\log_{10} d - 10\log_{10} 30$$

Table 3.6 shows all measured and calculated values at both 13.56 and 27.00 MHz. Note that the EIRP reported is computed based on the maximum field strength produced by main lobe the antenna's field pattern. The true radiated power will be less, but the exact value is not known unless the gain of the antenna is known. Therefore, the path loss that will be

P_{rx}	[dBm]	received power measured from spectrum analyzer (measured)
V_{rx}	[dBμV]	received voltage measured from spectrum analyzer (computed)
C_a	$\left[\frac{\text{dBm}}{\text{m}}\right]$	antenna calibration factor (measured by FCC)
L_c	[dB]	loss of cable connecting receiving antenna and spectrum analyzer (measured)
Z_0	[Ω]	terminal impedance of receiving antenna and spectrum analyzer (50 Ω)
E_{rx}	$\left[\frac{\text{dBμV}}{\text{m}}\right]$	field strength at receiving antenna (computed)
d	[m]	distance between transmitter and receiver (measured)
P_{EIRP}	[dBm]	effective isotropic radiated power (EIRP) from transmitter (computed)

Table 3.5: Variables used in calculation of EIRP.

Frequency	**C_a (dBm/m)**	**L_c (dB)**	**P_{rx} (dBm)**	**d (m)**	**V_{rx} (dBμV)**	**E_{rx} (dBμV/m)**	**P_{EIRP} (dBm)**
13.56 MHz	0.21	10.9	−56.36	3	50.63	61.74	−33.49
27.00 MHz	0.33	9.2	−43.77	3	63.22	72.75	−22.48

Table 3.6: Radiated field strength measurements at both 13.56 and 27.00 MHz.

computed in Section 3.3.3 will be an overestimate, as the true radiated power is less.

3.3.2 Received Power Data Collection

Once the radiated power of the test node is known, we must measure the received power at the other end of the channel in order to compute path loss. Since the channel is different at each location in every home, many such measurements were made in each of the 15 homes used for the background noise data collection, and listed in Table 3.1. In order to constrain the search space, we fixed the receiver location in each home as the place where the home's internet modem is. Although this may not be the best place for the receiver, and in fact may have significantly more noise on the power line than other locations, it's a very practical location, assuming that the SNUPI base-station receiver needs to have an internet connection in order to share SNUPI data with the world-wide-web. At this receiver

location, the same receiver hardware used for background noise data collection and shown in Figure 3.1 was used to capture the received signal from the power line. This allowed us to capture the received signal using all three connections to the power line: line-to-neutral (LN), line-to-ground (LG), and neutral-to-ground (NG).

In order to constrain the number of locations in the home to test the path loss, we placed the test transmitter shown in Figure 3.9 at every water appliance in the home. These locations included: sinks, toilets, showers, baths, dishwashers, refrigerators, freezers, laundry machines, hot water tanks, ice makers, boilers, hot tubs, and sump pumps. The number of locations tested in each home is listed in Table 3.1, and vary between 5 and 19 locations, with an average of 13. Since SNUPI operates based on near-field coupling of the RF signal onto the power lines, it is hypothesized that path loss should depend greatly on the orientation of the transmitting antenna. To test this dependence on orientation, the transmitter was placed in four different orientations as shown in Figure 3.12. Although this does not represent a thorough exploration of the effects of orientation, it allow us to see if any of the orientations tested perform statistically better than the others when arbitrarily placed. The assumption is that a practical sensor node will either need to be placed on the horizontal or the vertical. The reason for testing angles 9.5° off horizontal or vertical, is to test to see if there is any benefit in placing the antenna at a slight angle inside the sensor node's enclosure. All locations and orientations were repeated for both 13.56 and 27.00 MHz.

At each transmit frequency, location, and orientation, the receiver captured 3.0 s of data. These 3.0 s of data were divided up into 45 individual 66 ms windows, and the PSD was computed over each window. For each window, the frequency of the greatest peak was measured, and if this peak was within 1 kHz of the intended frequency (*i.e.*, either 13.56 or 27.00 MHz), then it was assumed to be the tone sent by the transmitter, otherwise it was assumed to be noise. The percentage of the 45 time windows in which the signal peak was found is defined as the "coverage," where 100% coverage means that all 66 ms windows contained signal, and 0% coverage means that only noise was measured. This notion of coverage is important because in some locations, the transmitted tone was too weak to be

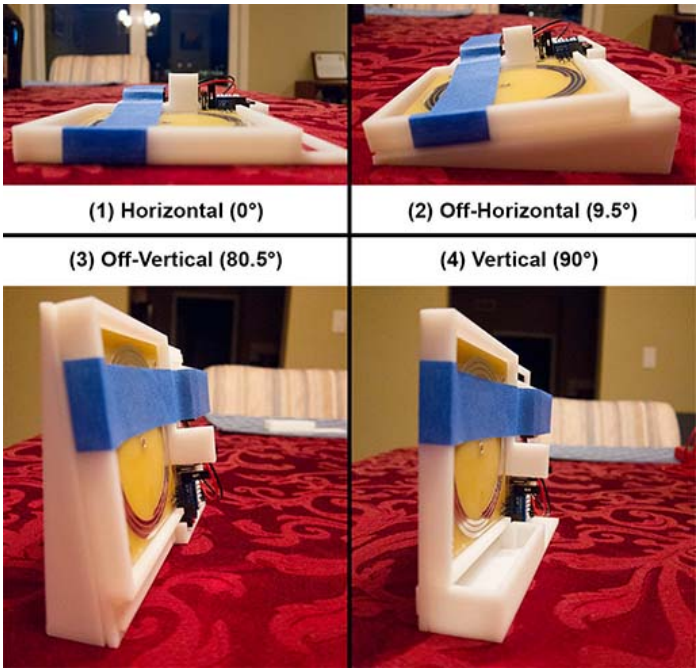

Figure 3.12: Transmitter orientations used in path loss measurements.

seen above the noise floor at the receiver, and therefore these locations will not allow us to measure the path loss. The average of the peak amplitude over all covered windows was then measured to represent the received power.

3.3.3 Path Loss Analysis

First, we want to determine what the average path loss is, and how much it varies between homes. Table 3.7 shows the average path loss for each home at both 13.56 and 27.00 MHz using only the LG power line connection and the horizontal orientation of the test transmitter. This table also allows us to compare the path loss at 13.56 and 27.00 MHz. We can see that the path loss was greater at 27.00 MHz in every home tested, and on average the path loss at 13.56 MHz was about 10 dB lower.

In addition to comparing the path loss of 13.56 and 27.00 MHz, the data gathered allows us to make conclusions about the best power line connection and transmitter antenna

Home	Path Loss 13.56 MHz	Path Loss 27.00 MHz	Path Loss 13.65 – 27.00 MHz
1	47.88 dB	57.15 dB	−9.27 dB
2	47.28 dB	54.00 dB	−6.72 dB
3	53.84 dB	67.96 dB	−14.11 dB
4	51.43 dB	60.25 dB	−8.82 dB
5	53.57 dB	57.48 dB	−3.91 dB
6	49.22 dB	56.98 dB	−7.76 dB
7	42.23 dB	65.31 dB	−23.08 dB
8	47.65 dB	58.77 dB	−11.12 dB
9	43.82 dB	61.91 dB	−18.09 dB
10	45.11 dB	56.97 dB	−11.86 dB
11	58.49 dB	65.65 dB	−7.16 dB
12	48.53 dB	57.15 dB	−8.63 dB
13	54.47 dB	65.81 dB	−11.34 dB
14	38.13 dB	45.83 dB	−7.70 dB
15	47.13 dB	57.91 dB	−10.78 dB
Avg	**49.16 dB**	**59.80 dB**	**−10.64 dB**
Max	**58.49 dB**	**67.96 dB**	**−9.47 dB**

Table 3.7: Path loss measurements at both 13.56 and 27.00 MHz, using only the LG power line connection and the horizontal orientation of the test transmitter.

orientation. To perform these comparisons, we define three metrics for evaluating the binary decisions between two variables. We will refer to these variables as X and Y, where X and Y could represent different power line connections (*e.g.*, LN and LG) or different transmitter orientations (*e.g.*, 0.0° and 90.0°). The three metrics are called "Difference," "Occurrence," and "Coverage." Each metric represents the confidence that Y is better than X, and is defined between -100 and 100, where positive values indicate that Y is better than X, and negative values indicate that X is better than Y.

The difference metric (M_d) represents the magnitude of the difference in received signal strength between the two conditions, and is computed using the following equation where $P_{r,Y}$ are the received signal strength values for all covered locations (*i.e.*, locations in which there was a signal measured and not just noise) under the Y condition, and $P_{r,X}$ are those under the X condition. The mean of the difference between these values for all covered locations is taken, and then scaled such that a 10 dB difference represents a confidence of 100. Note that this metric is limited at -100 and 100, so even if the difference is greater than 10 dB, the absolute value of the difference metric will be 100.

$$M_d = 100 \times \frac{\text{mean}(P_{r,Y} - P_{r,X})}{10 \text{ dB}}$$

The occurrence metric (M_o) represents the frequency with which the signal strength from one condition is significantly greater than the other condition, and is computed using the following equation. $O_{X<Y}$ is the percentage of covered locations (*i.e.*, locations in which there was a signal measured and not just noise) in which the signal strength in the Y condition is at least 3 dB greater than the signal strength in the X condition. Conversely, $O_{X>Y}$ is the percentage of covered locations in which the signal strength in the X condition is at least 3 dB greater than the signal strength in the Y condition. The occurrence metric represents the difference between $O_{X<Y}$ and $O_{X>Y}$, and is therefore inherently limited to -100 and 100.

$$M_o = O_{X<Y} - O_{X>Y}$$

The coverage metric (M_c) represents the frequency with which the coverage (*i.e.*, the number of locations in which there was a signal measured and not just noise) is greater than the other condition, and is computed using the following equation. $C_{X<Y}$ is the percentage of locations in which the coverage is greater in the Y condition. Conversely, $C_{X>Y}$ is the percentage of locations in which the coverage is greater in the X condition. The coverage metric represents the difference between $C_{X<Y}$ and $C_{X>Y}$, and is therefore inherently limited to -100 and 100.

$$M_c = C_{X<Y} - C_{X>Y}$$

The "Confidence" is a metric (C) defined to combine the difference, occurrence, and coverage metrics into a single quantity which can be used for determining which condition is best and with how much confidence. The confidence metric is simply a weighted average of the other three metrics as shown in the equation below, and where the weights are $w_d = 1$, $w_o = 4$, and $w_c = 2$. These weights were determined somewhat arbitrarily, but capture the idea that it is most important to choose a condition which (1) is more often significantly (*i.e.*, at least 3 dB) better than the other condition, and (2) is more likely to receive a signal when the signal is very weak. The weights also capture the idea that the absolute magnitude of the difference in signal power is of least importance.

$$C = \frac{w_d M_d + w_c M_o + w_c M_c}{w_d + w_o + w_c}$$

Table 3.8 shows the comparison of the different power line connections across all homes for both 13.56 and 27.00 MHz. It can be seen from the table that at 13.56 MHz LG is slightly better than NG, and both LG and NG are significantly better than LN (although this is not true in 2–4 homes). At 27.00 MHz, LG is better than both NG and LN, but only by a small margin. NG is slightly better than LN, but they are basically the same, as it is strongly dependent on the home. Therefore overall, we can conclude that LG is the best connection to use when it is available, but in homes that do not have a ground connection LN can be used as well.

Frequency	Angle	Best LN−LG (Conf.)	Best LN−NG (Conf.)	Best LG−NG (Conf.)
13.56 MHz	0.0°	LG (35.11)	NG (37.69)	LG (−0.35)
13.56 MHz	9.5°	LG (36.13)	NG (37.02)	LG (−2.02)
13.56 MHz	80.5°	LG (37.13)	NG (32.04)	LG (−5.32)
13.56 MHz	90.0°	LG (38.75)	NG (33.36)	LG (−6.16)
13.56 MHz	Agg	LG (36.80)	NG (34.99)	LG (−3.50)
27.00 MHz	0.0°	LG (7.63)	LN (−0.89)	LG (−12.42)
27.00 MHz	9.5°	LG (8.34)	LN (−1.05)	LG (−9.78)
27.00 MHz	80.5°	LG (9.02)	LN (−4.83)	LG (−15.15)
27.00 MHz	90.0°	LG (10.89)	LN (−0.47)	LG (−15.06)
27.00 MHz	Agg	LG (8.98)	LN (−1.84)	LG (−13.14)

Table 3.8: Path loss comparison between power line connections. "Conf." represents the confidence in the decision as defined in the text, where a higher absolute value indicates more confidence. "Agg" represents the aggregation of all angles/orientations.

Table 3.9 shows the comparison of the different transmitter orientations across all homes for both 13.56 and 27.00 MHz. It can be seen from the table that for both frequencies, vertical orientations are better than horizontal ones; however, the difference is never more than 2 dB. It should also be noted that this difference may be due to the antenna being detuned by the ground or nearby objects. It should also be noted that there was no significant difference between orientations that were at the horizontal or vertical and ones which were 9.5° off those planes. Therefore it can be concluded that the orientation makes only a small difference in aggregate and therefore either placement is valid.

3.4 Summary

This chapter explored the background noise and the path loss of the power line coupled wireless channel in order to better understand the channel and answer some fundamental questions about which operating frequencies work best, how to best couple signals onto the the power lines, and how to extract the communications signals from the power lines. Through experimentation, it was determined that the unlicensed ISM bands of 13.56, 27.12,

Frequency	PL Conn.	Best $0.0° - 9.5°$ (Confidence)	Best $0.0° - 80.5°$ (Confidence)	Best $0.0° - 90.0°$ (Confidence)	Best $9.5° - 80.5°$ (Confidence)	Best $9.5° - 90.0°$ (Confidence)	Best $80.5° - 90.0°$ (Confidence)
13.56 MHz	LN	$0.0°$ (-1.24)	$80.5°$ (5.39)	$90.0°$ (5.05)	$80.5°$ (9.75)	$90.0°$ (8.28)	$80.5°$ (-1.53)
13.56 MHz	LG	$0.0°$ (-2.99)	$80.5°$ (8.41)	$90.0°$ (6.65)	$80.5°$ (10.99)	$90.0°$ (9.73)	$80.5°$ (-2.65)
13.56 MHz	NG	$0.0°$ (-2.08)	$80.5°$ (3.76)	$90.0°$ (2.97)	$80.5°$ (8.77)	$90.0°$ (5.96)	$80.5°$ (-1.78)
13.56 MHz	Agg	$0.0°$ (-2.12)	$80.5°$ (5.87)	$90.0°$ (4.91)	$80.5°$ (9.85)	$90.0°$ (8.01)	$80.5°$ (-2.01)
27.00 MHz	LN	$0.0°$ (-3.46)	$80.5°$ (10.09)	$90.0°$ (8.11)	$80.5°$ (12.92)	$90.0°$ (11.71)	$80.5°$ (-3.97)
27.00 MHz	LG	$0.0°$ (-2.43)	$80.5°$ (7.62)	$90.0°$ (4.75)	$80.5°$ (9.84)	$90.0°$ (7.95)	$80.5°$ (-3.43)
27.00 MHz	NG	$0.0°$ (-2.84)	$80.5°$ (9.34)	$90.0°$ (6.25)	$80.5°$ (10.78)	$90.0°$ (9.60)	$80.5°$ (-2.67)
27.00 MHz	Agg	$0.0°$ (-2.92)	$80.5°$ (9.03)	$90.0°$ (6.38)	$80.5°$ (11.19)	$90.0°$ (9.76)	$80.5°$ (-3.36)

Table 3.9: Path loss comparison between transmitter orientations. "Confidence" represents the confidence in the decision as defined in the text, where a higher absolute value indicates more confidence. "Agg" represents the aggregation of all power line connections.

Frequency	13.56 MHz	27.00 MHz	Difference
Noise Level (dBm/Hz)	−96.3	−91.7	−4.6
Path Loss (dB)	50	60	−10
FCC EIRP Limit (dBm)	8.77	−15.23	24.00
Link Margin (dB)	55.07	16.47	38.60

Table 3.10: Frequency comparison between 13.56 MHz and 27.00 MHz. The noise level was determined using the 99.9% threshold and the LG power line connection.

and 40.68 MHz are best for use with the power line coupled wireless channel. However, the low power limit imposed the FCC on the 40.68 MHz ISM band makes it impractical for most SNUPI networks. This leaves 13.56 and 27.12 MHz, which are compared in Table 3.10. Note that the table contains data for experiments run at 27.00 MHz rather than 27.12 MHz; however, both lie within the 27.12 MHz ISM band.

It can be seen from Table 3.10 that at 13.56 MHz, the noise level is on average 4–5 dB lower, the path loss is about 10 dB lower, and the maximum allowed transmit power is 24 dB higher. As a result of this, the 13.56 MHz channel has about 39 dB of additional link margin over 27.00 MHz, and therefore it is a clear choice for most SNUPI networks.

Both the background noise analysis and the path loss analysis both showed that the best power line connection is the line-to-ground (LG) connection, although the difference between each of the power line connections is no more than a few dB. It's important to note that in older homes, there is no ground connection, and therefore LN is the only option but performance will not deteriorate too much.

Regarding the orientation of the transmitting antenna, vertical orientations were shown to be up to 2 dB better than horizontal ones; however, the difference is so small that it is likely not significant.

Chapter 4

SNUPI COMMUNICATIONS PROTOCOL

This chapter focuses on the communications protocol that is used to send data between the SNUPI nodes and the base-station receiver that is plugged into the power line. The design of this communications protocol is influenced by the unique properties of the SNUPI network. In particular, the protocol is focused on pushing as much of the complexity as possible to the base-station receiver in order to support many simple, low-power, and low-cost sensor nodes. In addition to supporting ultra-low-power consumption on the SNUPI nodes, the protocol needs to be robust in the face of the unidirectional communication of the network, and the noise and interference present on the power line coupled wireless channel. This poses a significant challenge for the communications protocol to overcome.

Traditional WSNs rely on bidirectional communication between the nodes in order to facilitate complex communications protocols which involve link formation, synchronization, routing, scheduling, and complex error handling. Without bidirectional communication in the network, most of these protocols cannot be implemented, thus making the SNUPI communications protocol seem very simple. Although the simplicity of the SNUPI communications protocol is a significant advantage to the SNUPI network, it's also one of it's greatest disadvantages. Enabling robust communication through a unidirectional network is quite challenging, and therefore poses a significant challenge for the SNUPI communications protocol to overcome. Enabling a back-channel, even a simple one would dramatically improve the robustness of the SNUPI network. Although this remains future work, Section 7.2 discusses ways in which the SNUPI network could be made bidirectional, even if used only for protocol control messages.

This chapter goes through each of the major parts of the communications protocol, with

a strong focus on the bitrate in Section 4.4 and the forward error correction in Section 4.6.

4.1 Network Topology

Traditional WSNs rely on bidirectional communication between the nodes in order to facilitate complex communication protocols which involve link formation, synchronization, routing, scheduling, and error handling. Using this bidirectional communication, there are many different network topologies that can be supported. However, SNUPI networks currently only support unidirectional communication, and as a result the options for the network topology are severely limited. In order to reduce power consumption, SNUPI networks therefore implement the simplest of multi-node network topologies: the star network. A star topology is one in which there is a single base station receiver and many transmitting nodes that communicate directly with the receiver. The power line infrastructure serves as a very convenient channel for this type of network, because the power lines are dispersed to almost every position in a home. Using this topology, SNUPI networks use a unidirectional version of the ALOHA protocol, in which many transmitters communicate in short bursts on the same channel [2]. Since each transmission is very short, the probability of a collision is low. The delay between each transmission is different among each transmitter to ensure that two transmitters do not become synchronized, and therefore cause collisions on every packet transmission. Using this protocol, we can implement a very simple unidirectional single-hop network and minimize the power and time spent during transmission.

4.2 Automatic Repeat Request

Since SNUPI nodes are unidirectional and all share the same channel, collisions are likely to occur. Since SNUPI networks simply use a pure ALOHA protocol, the probability of collisions (P_{loss}) can easily be computed as a function of the number of sensor nodes in the network (N), and the transmit (t_{tx}) and sleep (t_{sleep}) times:

$$P_{\text{loss}} = 1 - \exp(-2N\frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{sleep}}})$$

Since SNUPI nodes are unidirectional, there is no way to detect that packets have collided and therefore no way to know when to resend the packet data if it doesn't make it. This idea of resending packets that don't make it to the receiver is known as automatic repeat request (ARQ), and simply cannot be implemented on a unidirectional network. Although still future work, Section 7.2 discusses ways in which the SNUPI network could be made bidirectional in order to enable ARQ support.

Although unidirectional networks cannot implement ARQ, they can reduce the probability of collisions by always sending multiple copies of the packet data or by using forward error correction (FEC). FEC techniques are widely used in SNUPI networks, and are discussed in Section 4.6. Multiple retransmissions can also be used to increase the probability that data is received without collisions [220]. If the timing between the retransmissions are randomized so that each transmission is independent, then the probability that the data is received (P_{rx}) when using n retransmissions becomes:

$$P_{\text{rx}} = 1 - (P_{\text{loss}})^n$$

In this way, multiple retransmissions can be used to increase the probability that important data makes it to the receiver. Although there is no guarantee that the data will make it to the receiver, an arbitrary level of reliability can be achieved through statistical proof.

4.3 Modulation Scheme

The modulation scheme describes how data is to be encoded on the RF channel, and can greatly effect the power consumption of both transmitter and receiver, as well as the network's robustness to different types of noise and interference. Low power sensor nodes most commonly use binary modulation schemes such as on-off keying (OOK), amplitude shift keying (ASK), binary phase shift keying (BPSK), and binary frequency shift keying (BFSK). In order to increase the efficiency of the communications channel and bring the channel capacity closer to the Shannon limit, more complex WSNs implement M-ary orthogonal modulation schemes in which symbol alphabets larger than 2 are used. The most

common examples are M-ary phase shift keying (M-PSK), M-ary frequency shift keying (M-FSK), as well as several popular quadrature modulation schemes including minimum shift keying (MSK) and quadrature amplitude modulation (QAM).

Although M-ary orthogonal modulation schemes can increase the throughput of communications, they can also have a significant negative impact on the power consumption of the sensor node. As a result, previous work has shown that for most WSNs it is best to use only binary modulation schemes [198]. Furthermore, amplitude-based modulation schemes like OOK and ASK have often been used in very simple WSNs, but commonly suffer from issues related to noise immunity and robustness. The bit error rate in an additive white Gaussian noise (AWGN) channel is double for BFSK compared to BPSK. Although there are performance benefits for using BPSK, BFSK is superior in terms of the power consumption of the transmitter. When using PSK, the power amplifier needs to be linear in phase, and therefore consumes more power than a non-linear amplifier. However, using BFSK does not require the use of a linear amplifier, and therefore can be implmented using a lower power transmitter. As a result, all of the SNUPI implementations to date use BFSK as the modulation scheme.

4.4 Bitrate

The bitrate is the rate at which the channel bits are transmitted over the wireless channel. Although this is a software controlled parameter, it is tied in very closely with physical constraints like the amount of frequency deviation (assuming a BFSK modulation scheme) and the bandwidth of the channel. For coherent FSK (*i.e.*, the two frequencies have the same phase), the bitrate (R_b) is constrained by the amount of frequency deviation (Δf) in the signal [244]:

$$R_b \leq 2\Delta f$$

For a given hardware design, the frequency deviation is fixed, so this puts a limit on the maximum bitrate which can be used. Generally speaking, it is best to maximize the bitrate so that the packets are shorter and therefore occupy the channel less time. This can

therefore reduce the likelihood of having packet loss due to collisions. However, the choice of frequency deviation and bitrate also influence the bandwidth of the channel that is occupied. There are many definitions for bandwidth, but we'll use the 99% necessary bandwidth (B_{99}), which is what the FCC uses for most regulations [67]. The 99% necessary bandwidth is the bandwidth under which 99% of the radiated power lies, and is approximated by the following equation for BFSK [67]:

$$B_{99} = \begin{cases} 1.93\Delta f + 0.27R_b & 0.03 < \frac{\Delta f}{R_b} < 1 \\ 1.2\Delta f + R_b & 1 \leq \frac{\Delta f}{R_b} \leq 2 \end{cases}$$

The necessary bandwidth is important to keep in mind because this is the bandwidth the FCC will not allow the sensor network to violate. For example, the FCC allocated 14 kHz of bandwidth centered at 13.56 MHz to the ISM band; however, in order to be legal, the 99% necessary bandwidth must not spill outside the 14 kHz allocated in that ISM band. This means that the true bandwidth that the network can use is quite a bit lower than 14 kHz. This is because some amount of guard band must be left to ensure that the network will not violate the FCC regulations. In addition the two frequencies generated by the transmitter are highly unlikely to be perfectly centered at 13.56 MHz, and finally there are process, voltage, and temperature tolerances in all of the components in the transmitter, but most notably in the crystal and its load capacitors.

It should also be noted that the bitrate cannot be made any arbitrary value. In fact, it most likely needs to be derived from a crystal oscillator on the node, so this limits the potential bitrate to a list of attainable frequencies. To deal with all of these practical constraints, I wrote a program which can be used to determine if the current bitrate, and frequency deviation are appropriate to fit inside of the band allowed by the regulations. My program can also be used to find the optimal bitrate or frequency deviation given a specified regulatory band and all of the system tolerances.

4.5 Packet Structure

Most generally, SNUPI packets can be structured in the same way as the packets in any other communications system. However, since SNUPI nodes are optimized for ultra-low power consumption, care must be taken to keep the packets as short as possible. This not only saves power but also reduces the probability of packet collisions by occupying the channel for less time. However, there is some startup time during which the SNUPI transmitter turns on the RF oscillator and prepares to send a packet. Since the power consumption during this startup time is approximately the same as the power consumption during transmission, it makes sense to have packets that are approximately the same length as the startup time or longer, so that this wasted power is minimized. This section briefly discusses some of main parts of that packet that are required. Note that all parts of the packet that are discussed here, with the exception of the preamble, are assumed to be before encoding the packet using forward error correction, which is described in Section 4.6. The main parts of the packet are the:

1. Preamble - allows packets to be detected by the receiver
2. ID - to identify which node sent the packet
3. Data - the data payload of the packet
4. CRC - error detection bits

The preamble is import for allowing the packets to be detected at the receiver and identified as something other than noise. Although this can be any sequence of known bits, it is common to use certain sequences with known favorable properties. Many WSNs including some of the current SNUPI implementations use an 11-bit Barker code due to its favorable autocorrelation properties. Using a sequence like a Barker code allows the receiver to run an autocorrelation on the input stream in order to find potential packet headers. This works better the longer the Barker code, and as a result many systems use repeat the Barker code several times to help with packet detection. Although this improves packet detection, it results in higher power consumption and higher packet loss by lengthening the packet size. Therefore, a trade-off must be made between the number of packets which can be

detected using a small preamble and the savings in terms of power and packet collisions. It should also be noted that the preamble should not be encoded using FEC as described in Section 4.6 as this would change its autocorrelation properties.

The ID is used to indicate which sensor node sent the data. Again, there is a trade-off here between the number of bits required to uniquely identify every sensor in the network, and the need to keep the packets as short as possible.

The data bits can represent anything, but again care must be taken to keep the payload as short as possible. As a result, it is common to compress or pre-encode the data into a form in which messages can be represented by fewer bits. In addition, it should be noted that some of the data bits are likely to be used to tell the decoder the meaning of the rest of the payload. It is common therefore to use "type" or "mode" bits inside the payload for this purpose.

Finally, there are some number of bits which are required to detect errors in the decoded packet. In the simplest case, the earliest SNUPI implementations use a single parity bit. However, this only allows for the detection of a single flipped bit, and therefore can let a significant amount of erroneous packets through the decoder. It is therefore more common to use an 8 or 16-bit cyclic redundancy check (CRC). The exact number of check bits needed is a function of the importance of the data's correctness, the number of bits in the packets, and the noise present in the channel. However, like all other parts of the packet, care should be taken to keep the error checking bits as short as possible.

4.6 Forward Error Correction

Forward error correction (FEC) is a method of encoding the data to be transmitted so that it contains enough redundant information to correct erroneous bit transitions either due to noise, interference, or collisions. In this section, I will describe a few different kinds of error-correction codes (ECC), including Hamming codes, BCH codes, Reed-Solomon codes, convolutional codes, and combined ECC/RLL codes. Hamming codes, BCH codes, and Reed-Solomon codes are all different types of linear block codes, while the remaining

Block Codes	
m	
t	number of error bits that can be corrected
b	number of shortened bits
$d_{\min} = 2t - 1$	minimum distance metric
$n = 2^m - 1 - b$	block size
$k = n - mt$	message size
$R = \frac{k}{n}$	code rate
L	length of uncoded data packet (bits)
$M = \lceil \frac{L}{k} \rceil n$	length of coded message (bits)
$R_{\text{eff}} = \frac{L}{M}$	effective code rate
$G = 10\log_{10}(d_{\min} R_{\text{eff}})$	effective coding gain (dB)
Convolutional Codes	
K	constraint length
M	total memory
d_{free}	free distance metric
R	code rate
L	length of uncoded data packet (bits)
$M = L + \frac{K-1}{R}$	length of coded message (bits)
$R_{\text{eff}} = \frac{L}{M}$	effective code rate
$G = 10\log_{10}(d_{\text{free}} R_{\text{eff}})$	effective coding gain (dB)

Table 4.1: Variables used to compare error-correcting codes.

codes are convolutional codes. Table 4.1 lists many of the variables used throughout this chapter to describe and compare these codes.

Throughout the chapter I will refer to the code rate (R) as the number of data bits over the number of encoded bits. For example, the packet data to be sent over the channel will be a twice the length of the original, uncoded data when using a rate $1/2$ code. I will also refer to the effective code rate (R_{eff}) which also takes into account the coding overhead for a specific packet length.

In order to describe the error correcting capability of each code, I will refer to the code's distance metric. The block codes will be described using the minimum distance (d_{min}), while convolutional codes will be described using the free distance (d_{free}).

In order to compare these codes, I will refer to the coding gain (G) which expresses the effective improvement in signal-to-noise ratio (SNR) or E_b/N_0 due to the use of the code. The coding gain makes it easy to balance the trade-off between the gain seen from a high distance metric and the loss seen due to a low rate. The coding gain that will be computed throughout this section will be the theoretical coding gain for an AWGN channel, assuming a soft-decision decoder. Although we know that the PLCW channel is not AWGN, this gives us a crude way of comparing the unlimited numbers of potential codes. It remains future work to develop a metric for evaluating these codes specifically for the PLCW channel.

4.6.1 Hamming Codes

Hamming codes are the simplest type of block codes, and can correct at most 1 bit error (*i.e.*, $t = 1$). For Hamming codes, the minimum distance (d_{min}) is always 3, but the parameter m can be varied as long as $m \geq 2$. Therefore, using the equations in Table 4.1, you can compute the parameters for any arbitrary Hamming code, and evaluate how well it will work on an arbitrary length packet. Table 4.2 computes all of the code parameters for Hamming codes with $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet (*i.e.*, $L = 80$).

From this table, it can be seen that the code rate (R) improves with increasing values of m; however, this does not translate into improved gain (G) because the effective rate (R_{eff}) does not always improve. This is due to the unnecessary overhead in the codes with a longer block size, and can be solved by "shortening" the code. Instead of using a (n,k) Hamming code, b bits of the code do not need to be sent over the channel, thus resulting in the shortened $(n-b,k-b)$ Hamming code. Essentially, both the transmitter and receiver can assume that b-bits of the packet and message are zeros, and thus they don't need to be sent over the channel. This technique of shortening codes enables a code to benefit

m	t	d_{min}	n	k	R	R_{eff}	G (dB)
2	1	3	3	1	0.33	0.33	0.00
3	1	3	7	4	0.57	0.57	2.34
4	1	3	15	11	0.73	0.67	3.01
5	1	3	31	26	0.84	0.65	2.87
6	1	3	63	57	0.90	0.63	2.80
7	1	3	127	120	0.94	0.63	2.76
8	1	3	255	247	0.97	0.31	−0.26

Table 4.2: Example Hamming codes for $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet.

from a long block size without the need to send large packets, and is therefore ideal for wireless sensor networks. Table 4.3 shows the top 10 codes sorted by coding gain when using shortened Hamming codes with $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet (*i.e.*, $L = 80$). It can be seen from the table that the best shortened Hamming code is 1.4 dB better than the best un-shortened code.

4.6.2 BCH Codes

BCH codes are block codes just like Hamming codes, but they give us the option of choosing how many bits the code can correct (t). This can be extremely powerful as it can allow these codes to correct several bits that get corrupt from a single packet; however, they have the downside of being lower rate. The parameters for BCH codes can be computed using the equations in Table 4.1, and like Hamming codes they can be shortened. Table 4.4 shows some example BCH codes both shortened and un-shortened that were generated with $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet (*i.e.*, $L = 80$). The table includes shortened and un-shortened codes with the highest gain, highest minimum distance and highest effective rate.

It can be seen that the overall coding gain can significantly higher than Hamming codes, but the rate is never as high. In fact the BCH codes with the highest effective rates are

m	**t**	**b**	$\mathbf{d}_{\text{min}}$	**n**	**k**	**R**	$\mathbf{R}_{\text{eff}}$	**G (dB)**
7	1	40	3	87	80	0.92	0.92	4.41
7	1	39	3	88	81	0.92	0.91	4.36
8	1	167	3	88	80	0.91	0.91	4.36
7	1	38	3	89	82	0.92	0.90	4.31
8	1	166	3	89	81	0.91	0.90	4.31
7	1	37	3	90	83	0.92	0.89	4.26
8	1	165	3	90	82	0.91	0.89	4.26
7	1	36	3	91	84	0.92	0.88	4.21
8	1	164	3	91	83	0.91	0.88	4.21
6	1	17	3	46	40	0.87	0.87	4.16

Table 4.3: Example shortened Hamming codes for $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet.

m	**t**	**b**	$\mathbf{d}_{\text{min}}$	**n**	**k**	**R**	$\mathbf{R}_{\text{eff}}$	**G (dB)**
8	21	7	43	248	80	0.32	0.32	11.42
8	21	0	43	255	87	0.34	0.31	11.30
7	1	40	3	87	80	0.92	0.92	4.41
4	1	0	3	15	11	0.73	0.67	3.01
8	31	0	63	255	7	0.03	0.03	2.17

Table 4.4: Example BCH codes for $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet.

m	t	b	d_{min}	n (sym)	k (sym)	n (bit)	k (bit)	R	R_{eff}	G (dB)
6	14	21	29	42	14	0.33	252	84	0.32	9.64
6	21	0	43	63	21	0.33	378	126	0.21	9.59
4	2	0	5	15	11	0.73	60	44	0.67	5.23
5	1	13	3	18	16	0.89	90	80	0.89	4.26
8	127	0	255	255	1	0.00	2040	8	0.00	0.00

Table 4.5: Example Reed-Solomon codes for $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet.

actually Hamming codes (*i.e.*, $t = 1$). It should be noted that the best BCH code (*i.e.*, the top row of Table 4.4) strikes a balance between having a high minimum distance and a high effective rate.

4.6.3 Reed-Solomon Codes

Reed-Solomon codes are similar to BCH codes, but instead of operating on individual bits, they operate on m-bit long symbols or blocks of data. While BCH codes can correct t bits, Reed-Solomon can correct t symbols, each being m-bits longs. Reed-Solomon codes are therefore very useful in the face of bursty errors in which noise or interference is more likely to corrupt several consecutive bits rather than random bits. The parameters for Reed-Solomon codes can be computed using the equations in Table 4.1; however, the computed values will all be in units of m-bit symbols. To convert to bits, the number of symbols needs to be multiplied by m. Just like Hamming and BCH codes, Reed-Solomon codes can be shortened by b, m-bit symbols. Table 4.5 shows some example Reed-Solomon codes both shortened and un-shortened that were generated with $2 \leq m \leq 8$, and assuming an 80-bit uncoded packet (*i.e.*, $L = 80$). The table includes shortened and un-shortened codes with the highest gain, highest minimum distance and highest effective rate.

It can be seen that the overall coding gain for Reed-Solomon codes is less than that of the BCH codes; however, the number bit errors that can be corrected is much greater, and Reed-Solomon codes are stronger in the face of bursty errors, which the gain metric does

not capture. Like the BCH codes, it should be noted that the best Reed-Solomon code (*i.e.*, the top row of Table 4.5) strikes a balance between having a high minimum distance and a high effective rate.

4.6.4 Convolutional Codes

Convolutional codes are quite different from block codes in that they are generated using a convolution of the entire packet, and do not rely on blocks of data. They also need to be decoded differently, typically using a Viterbi decoder, described in Section 6.3.2. Since convolutional codes don't use blocks of data, this means that they do not have the same overhead as block codes in that the amount of overhead is not determined by partially used blocks. Instead, they have some number of memory cells which need to be stored in addition to the regular message bits, and as a result it's the number of these extra bits that determine the overhead. This overhead is called the constraint length (K), and is present in all convolution codes. Convolutional codes cannot be algebraically generated like many block codes, and are typically found is tables which list useful codes [146]. The equations in Table 4.1 can be used to compute the effective rate and the coding gain for convolutional codes. Table 4.6 lists the parameters for the most common convolutional codes found in [146] assuming an 80-bit uncoded packet (*i.e.*, $L = 80$).

Like block codes, convolutional codes can be shortened into what are called punctured codes. These punctured codes are therefore higher rate, but otherwise operate like other convolutional codes. Table 4.7 lists the parameters for the most common punctured convolutional codes found in [146] assuming an 80-bit uncoded packet (*i.e.*, $L = 80$).

There is a special class of punctured convolutional codes which are called rateless or rate-compatible punctured convolutional (RCPC) codes [84]. These rateless codes have the special property that a nearly unlimited number of message symbols can be generated from the source packet bits, and the source bits can be recovered from any subset of the message symbols of a certain length. These rateless codes are typically used to continually provide more message bits in subsequent packets to help decode a previously undecodable

K	M	d_{free}	R	R_{eff}	G (dB)	K	M	d_{free}	R	R_{eff}	G (dB)
17		20	0.50	0.42	9.21	8		22	0.25	0.23	7.04
16		19	0.50	0.42	9.03	8		16	0.33	0.30	6.86
14		36	0.25	0.22	8.89	7		20	0.25	0.23	6.68
15		18	0.50	0.43	8.84	7		10	0.50	0.47	6.68
14		26	0.33	0.28	8.68	7		15	0.33	0.31	6.63
13		33	0.25	0.22	8.56	8		10	0.50	0.46	6.63
12		32	0.25	0.22	8.47	4	6	7	0.67	0.65	6.55
12		24	0.33	0.29	8.43	3	6	6	0.75	0.73	6.42
13		16	0.50	0.43	8.42	6		18	0.25	0.24	6.27
13		24	0.33	0.29	8.38	6		13	0.33	0.31	6.06
14		16	0.50	0.43	8.38	4	5	6	0.67	0.65	5.88
12		15	0.50	0.44	8.19	5		16	0.25	0.24	5.81
11		22	0.33	0.29	8.10	5		12	0.33	0.31	5.77
11		29	0.25	0.22	8.09	6		8	0.50	0.47	5.76
6	10	10	0.67	0.63	8.00	3	5	5	0.75	0.73	5.63
11		14	0.50	0.44	7.94	5		7	0.50	0.48	5.23
10		27	0.25	0.22	7.83	3	4	5	0.67	0.65	5.14
10		20	0.33	0.30	7.73	4		10	0.33	0.32	5.03
4	9	8	0.75	0.72	7.62	4		13	0.25	0.24	4.96
6	9	9	0.67	0.63	7.54	2	3	4	0.75	0.74	4.72
9		12	0.50	0.45	7.37	4		6	0.50	0.48	4.61
9		24	0.25	0.23	7.37	3	3	4	0.67	0.65	4.17
9		18	0.33	0.30	7.32	3		8	0.33	0.32	4.11
10		12	0.50	0.45	7.32	3		5	0.50	0.49	3.87
5	7	8	0.67	0.64	7.08	3		10	0.25	0.24	3.87
5	8	8	0.67	0.64	7.08	2	2	3	0.67	0.66	2.98
4	8	7	0.75	0.72	7.04						

Table 4.6: Example convolutional codes from [146], and assuming an 80-bit uncoded packet.

K	M	d_{free}	R	R_{eff}	G (dB)
5		6	0.67	0.63	5.81
6		6	0.67	0.63	5.76
6		5	0.75	0.71	5.48
3		4	0.75	0.73	4.66
4		4	0.75	0.72	4.61
5		4	0.75	0.71	4.56
3		4	0.67	0.65	4.15
4		4	0.67	0.64	4.10
2		3	0.75	0.74	3.47
2		3	0.67	0.66	2.96

Table 4.7: Example punctured convolutional codes from [146], and assuming an 80-bit uncoded packet.

packet. In this way, additional bits can continually be provided until the original message is decoded. These types of codes are therefore ideally suited for networks with bidirectional communication in which the transmitter will continually add more message bits in subsequent packets until it receives the acknowledgment that the data was properly decoded. SNUPI networks are unidirectional, and therefore cannot take advantage of this property; however, rateless codes like RCPC codes are still very useful in recovering from a whole packets being corrupted or missed due to noise or interference, as is commonly seen on the PLCW channel. To do this, every SNUPI packet would need to contain the current message bits plus additional message bit for the past N packets sent. This would not guarantee that data makes it to the receiver, but it would greatly increase the probability. Implementing these codes and testing them on the PLCW channel remains future work.

4.6.5 Run-Length Limited Codes

In bandwidth limited channels, a higher data rate can be achieved by using an run-length limited (RLL) code with a minimum run length constraint as well as a maximum run length

Data		Codeword
10	↔	1000
11	↔	0100
011	↔	000100
010	↔	001000
000	↔	100100
0011	↔	00100100
0010	↔	00001000

Table 4.8: IBM (2,7) RLL encoding table [100].

constraint. This became popular with the rate $1/2$ (2,7) RLL code used by IBM in the 3370 and 3380 high-performance rigid disk files [100]. Table 4.8 shows the encoding used for this RLL code. This code has a minimum run-length of 2 bits and maximum run length of 7 bits, as indicated by the standard (2,7) RLL notation. As a result, 3 channel bits can fit in the minimum transition time, and therefore the channel bits can be sent at a rate which is 3 times faster. However, the code rate is $1/2$, meaning that each data bit produces 2 channel bits. Therefore, it has an the effective code rate $R_{\text{eff}} = 1/2 \times 3 = 1.5$, resulting in a code gain of $10\log_{10}(R_{\text{eff}}) = 1.76$ dB.

Combined ECC/RLL Codes

Although a code gain of 1.76 dB is very promising, the problem with this type of code is its susceptibility to errors. With these codes, a single bit error will put the state machine in the wrong state and will allow the error to propagate to all of the remaining bits. In addition, Lin and Wolf have shown that using standard error correction codes (ECC) in combination with these RLL codes can actually lead to degraded performance since the ECC codes only consider bit errors in the original data sequence, not the RLL encoded sequence [128].

To solve this problem, combined ECC/RLL codes have been suggested which have both distance preserving (*i.e.*, error correction) and run-length limiting properties [70, 71].

In one SNUPI implementation, I decided to use code 1c presented in the 1990 paper by French and Lin, which I will call "French90-1c" [71]. This code has only 4 states, shown in Figure 6.8(a). This is a convolutional code with a constraint length of 3, a free distance of 10–12, a code rate of $1/8$, and (2,6) RLL properties. Like the IBM (2,7) RLL code, this code has a minimum run-length of 2, and therefore has an effective code rate $R_{\text{eff}} = 1/8 \times 3 = 0.375$, resulting in a code gain of $10 \log_{10}(R_{\text{eff}}) = -4.26$ dB. This negative gain is balanced however by the error correction code gain of $10 \log_{10}(d_{\text{free}})$ =10.00–10.79 dB. The overall code gain is therefore 5.74–6.53 dB.

4.6.6 Monte Carlo Simulations

This section compares two different combined ECC/RLL codes, the French90-1c code described in the previous section, and another similar code described by Lin and Wolf and labeled "Lin_R2/3_K6_M9_27RLL" [128]. Figure 4.1 shows the error correcting ability of each code by plotting the decoded bit error rate (BER) as a function of the channel bit error rate. In can be seen from the figure that the French90-1c code can bring the decoded BER down to 10^{-3} when the channel BER is as high as 10^{-1}. However, the channel error rate needs to be below 4×10^{-2} before the decoded BER reaches the same level.

Figure 4.2 plots the BER of each code as a function of the strength of the signal (*i.e.*, the energy per bit over the noise spectral density, E_b/N_0). It can be seen from the figure that the French90-1c code has simulated coding gain of about 4 dB, while the Lin_R2/3_K6_M9_27RLL code has a coding gain almost a full dB less.

4.6.7 Summary

This section described several different error-correction codes for potential use in SNUPI networks; however, all of these codes were compared and evaluated using an AWGN channel model, which is known to be different from that of the PLCW channel used by SNUPI. It remains future work, to test, implement, and evaluate most of these codes both in Monte Carlo simulations and in real-world experiments using a PLCW channel.

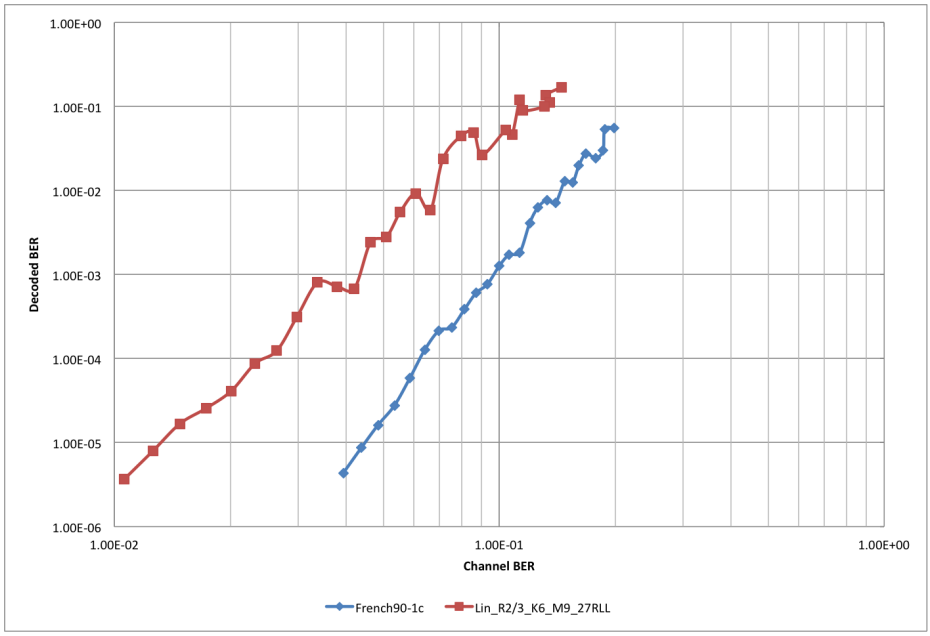

Figure 4.1: Correction rate plots for combined ECC/RLL codes.

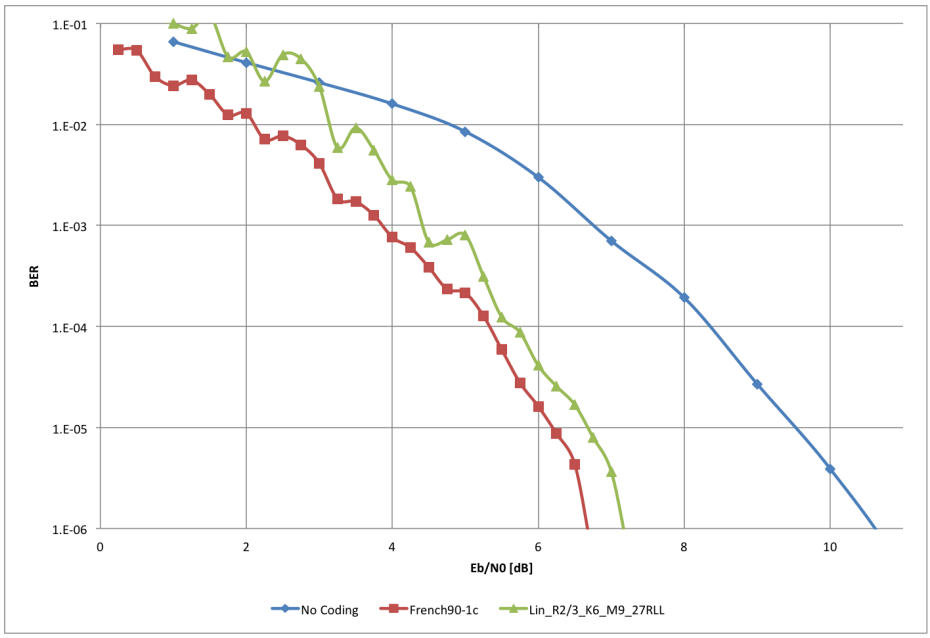

Figure 4.2: Bit error rate plots for combined ECC/RLL codes.

Chapter 5

SNUPI SENSOR NODE

This chapter describes the hardware design of the SNUPI sensor node. In particular, it will describe each of the major functional blocks of the hardware: power supply, microcontroller, transmitter, and antenna. In this chapter, each section will focus on a specific functional block of the SNUPI node, and present application-specific optimizations that can be made for each block. Finally, in Section 5.5, a comparison is made between SNUPI and existing wireless sensor nodes. Figure 5.1 (*left*) shows a picture and Figure 5.1 (*right*) shows a block diagram of the the first version of SNUPI from 2010 [44]. Although aspects of this implementation are described in this section, the goal is not to describe any one implementation, but to discuss the options a designer has when implementing a SNUPI sensor node. Throughout this chapter, a specific SNUPI implementations may be mentioned; however, there were over 5 completely different SNUPI implementations who's details comprise the content in this chapter. Figure 5.2 shows a picture of the first commercial implementation of a SNUPI sensor network. The WallyHome product launched in February 2014 by SNUPI Technologies[1], a company which I co-founded in 2012.

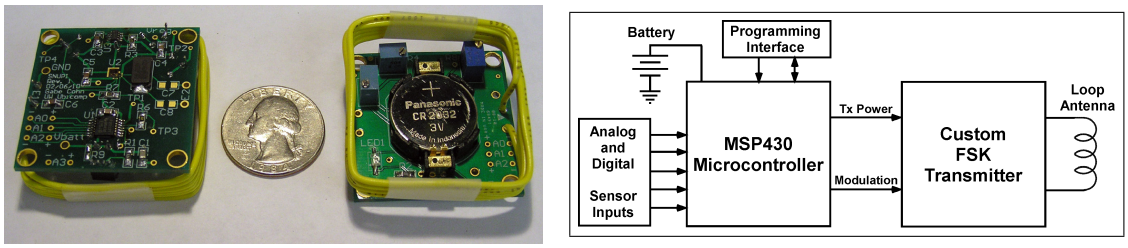

Figure 5.1: Picture (*left*) and block diagram (*right*) of 2010 SNUPI node.

[1] http://www.wallyhome.com

Figure 5.2: Picture of the 2014 commercial SNUPI node; part of the WallyHome sensor network.

5.1 Power Supply

In designing ultra-low-power sensor nodes, the power supply is one of the most important parts in determining the lifetime of the sensor node. The power supply needs to provide a stable source of energy over the long life (*e.g.*, 10 years) of the sensor node. In addition, it needs to be able to supply enough current for RF transmission, and be relatively stable in widely varying temperature conditions. At the same time, the power supply needs to be small, lightweight, and low-cost in order to make the sensor nodes practical. Although often overlooked, there are significant trade-offs in the choice of a sensor node's power supply. Although Section 5.1.5 will discuss power supplies that harvest energy from their environment, this chapter will focus primarily on batteries, as the most common power supply in WSNs.

When choosing a battery, it needs to meet several specifications of the sensor node:

1. Node lifetime requirement
2. Minimum and maximum voltage requirements
3. Peak current consumption of the node
4. Minimum and maximum temperature requirements

5. Physical size and weight constrains
6. Cost constrains

Finding a battery which meets all of these constraints is quite difficult and often takes a significant amount of experimentation. For this reason, most WSNs avoid finding an ideal battery, and instead choose a battery which meets some of the above constraints and deal with the consequences. The specifications which are often not met are the desired battery life, peak current consumption, temperature requirements, and physical size or weight of the battery. Choosing a battery which cannot supply the desired current or does not operate well outside of the desired temperature range has the most common side effect of decreased battery life. In order to achieve sufficient battery life, larger, heavier batteries are often used.

In this chapter, I hope to provide enough detail about the capabilities of sensor node batteries to allow the designer of a SNUPI sensor network to choose the appropriate battery for their specific application. For ultra-low-power sensor nodes, the most important parameter is most often the battery life. However, battery manufactures provide data which is not helpful for understanding how a battery will behave in the pulsed current operation of a sensor node. As a result, little is known about how much practical energy can be extracted from a battery in a sensor node. In Section 5.1.1, I describe an experiment designed to simulate the operation of a sensor node's battery over its lifetime in order to gain more insight for modeling battery life and to aid in choosing the proper battery. Section 5.1.2 discusses the results of this experiment, as well as different models for predicting the usable capacity, or energy stored within the battery. Although the analysis in Section 5.1.1 and Section 5.1.2 are limited to batteries with a standard 10 year shelf life, Section 5.1.3 discusses some batteries with a longer shelf-life. Section 5.1.4 discusses issues that arise from changes in the supply level, and how these can be mitigated. Finally, Section 5.1.5 moves away from traditional battery power supplies, and discusses some techniques for harvesting energy from the environment.

5.1.1 Battery Life Experiment

Battery manufacturers rarely give much more information than the rated capacity and nominal voltage of a battery. At times, battery datasheets will also include the voltage level over time while the battery is being discharged using a load on the order of 100 mA. Although it is useful to see how the voltage falls off over time, it's known that a battery behaves differently with different sized loads. In addition, it has been shown that the behavior is quite different when the battery is subjected to a pulsed load rather than a constant one [38, 39, 160]. As shown in Figure 5.3, the voltage decreases when a load is applied, but does not immediately return to its original unloaded voltage after the load is removed. If the load is not applied again for a certain amount of time, the voltage will slowly recover to a point nearly as high as it began. This "recovery" of the battery is one of the key differences between pulsed load operation and continuous load operation [38, 39]. It has been shown that the current draw during the pulse, the duration of the pulse, and the duration of the rest period between pulses is all relevant in the amount of charge that can be extracted from a battery.

The experiments described in this section were designed to simulate the current draw and timing of actual SNUPI nodes in order to yield accurate battery models. The basic idea behind the experiment is to slowly drain a battery with a fixed load to simulate the sleep current of the sensor node, and at the same time periodically apply a high-current load that simulates packet transmission. The test is to be run with a brand new battery and will continue until the battery is dead. Since SNUPI nodes often run at a duty cycle which allows them to last for a decade or more, it's obviously not practical to run the experiment at the exact same duty cycle. Instead the duty cycle is increased such that the battery can be drained in less than 2 months. However, a 2 month test is still very long and a number of different batteries under different conditions need to be tested. For this reason, the experiment runs 31 battery tests simultaneously.

Figure 5.4 shows a block diagram of the experimental setup. A National Instruments NI-DAQ PCI-6259 is used to collect data samples from all 32 channels (*i.e.*, 31 batteries and

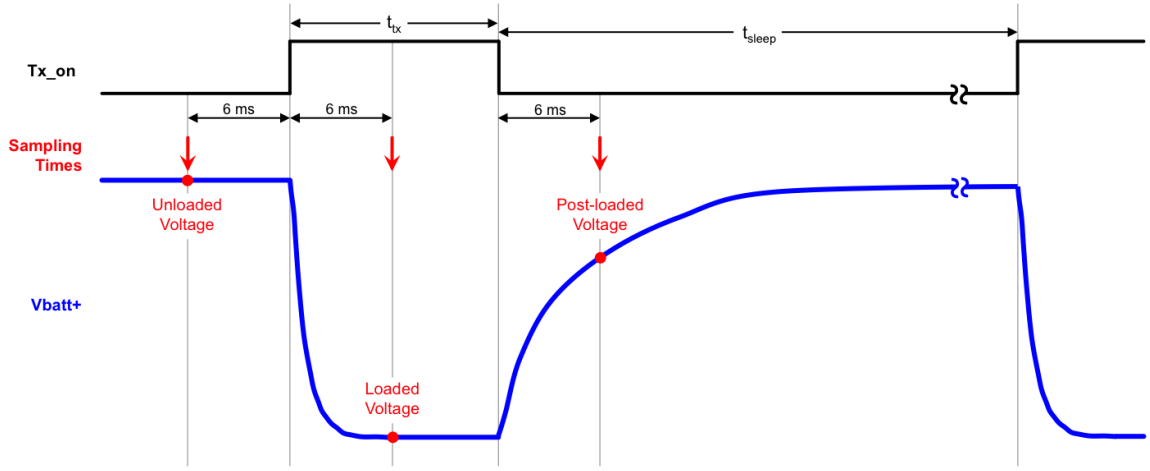

Figure 5.3: Timing diagram for the Battery Life Experiment, showing the three voltage samples (*i.e.*, *unloaded* voltage, *loaded* voltage, and *post-loaded* voltage).

1 test fixture). Since the system samples non-uniformly in time, an Arduino Duemilanove is used to control the timing, as shown in Figure 5.3. For a given test, the load (which represents the transmit current of a sensor node) is switched on for t_{tx} and then off for t_{sleep}. This process is repeated until the battery is dead. Every cycle, three voltage measurements are made. The *unloaded* voltage (V_{unload}) is measured 6 ms before the load is switched on and represents the unloaded battery voltage. The *loaded* voltage (V_{load}) is measured 6 ms after the load is switched on and represents the battery voltage during transmission. Finally, the *post-loaded* voltage (V_{post}) is measured 6 ms after the load is switched off in order to see how much the voltage recovers before the next cycle. In order to reduce noise in the measurement, the recorded values for each of these measurements is the average of 4 samples taken in quick succession. The load modulation times (t_{tx}) tested were 12, 24, and 48 ms to represent different packet lengths or different bit rates, as shown in Table 5.1. Sleep times (t_{SLEEP}) of 250, 500, and 1000 ms were used to test different duty cycles.

A 470 kΩ resistor, R_{SLEEP} is used to represent the current draw of the sensor node during sleep. The value of the resistor corresponds to roughly 6.4 µA from a 3.0 V supply. It should be noted, that the use of a fixed resistance rather than a current source means

t_{tx}	**Bits at 1 kb/s**	**Bits at 2 kb/s**	**Bits at 4 kb/s**	**Bits at 5 kb/s**	**Bits at 8 kb/s**	**Bits at 10 kb/s**
12 ms	12	24	48	60	96	120
24 ms	24	48	96	120	192	240
48 ms	48	96	192	240	384	480

Table 5.1: Packet times tested in the Battery Life Experiment, and their corresponding number of packet bits.

Load Current	R_{LOAD}
20 mA	150 Ω
40 mA	75 Ω
80 mA	37.4 Ω

Table 5.2: Test loads used in Battery Life Experiment. Note that a nominal battery voltage of 3.0 V is assumed.

that the exact current draw throughout the test is changes as the battery voltage drops. Despite this, a resistor, R_{LOAD} is also used to represent the current draw of the transmitter. The values of R_{LOAD} are chosen from Table 5.2, and correspond with transmit currents of approximately 20, 40, and 80 mA. The load resistor is switched into the circuit using the NTD5867 N-channel MOSFET whenever the Tx_on signal is high. To accurately represent the capacitive load between Vbatt+ and GND, 10 µF and 0.1 µF capacitors are put in parallel with the battery.

Since it is important to start with fresh batteries, and run them all the way until they are dead, care must be taken to ensure that the batteries don't discharge at all before the test begins. For this reason, the negative terminals of the batteries, Vbatt– were not connected directly to GND. A manual switch was used to connect them all to ground at once and to trigger the Arduino to start the test (see Figure 5.4). However, when running multiple month-long experiments, there are a number of issues that can arise which cause the test

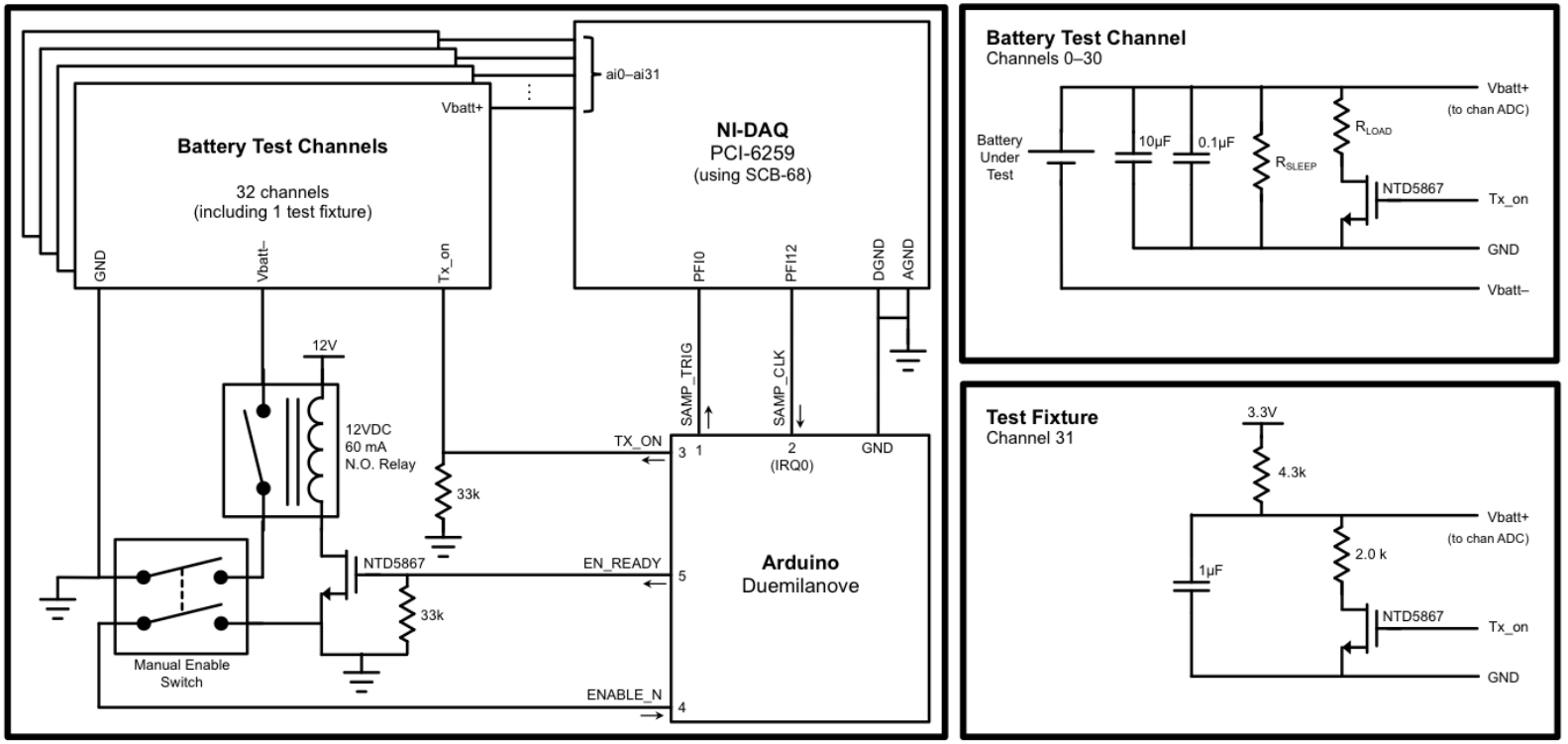

Figure 5.4: Block diagram of the Battery Life Experiment hardware, showing the data collection hardware on the left, and schematics of the battery test hardware and test fixture on the right.

to stop (*e.g.*, a power failure, software bug, or human intervention). As a result of this, a significant about of complexity has been added to the design to allow the tests to handle unknown errors, be stopped, and restarted without any loss of data. To stop the batteries from discharging when the test has been halted, a relay was put in series with the manual switch that connects Vbatt– to GND. This way, both the manual switch and the relay need to be on in order for the batteries to be connected the test circuitry. The relay is normally open, with a control signal coming from the Arduino, which will enable the relay during the test. If the Arduino loses power, or is reset, then the relay will open, and the test will be halted. In addition, there is communication between the Arduino system and NI-DAQ to ensure that both the timing circuitry and the sampling software stop if a problem is detected. The Arduino provides the sample trigger signal to the NI-DAQ, so data will only be collected if the Arudino is working properly and generating the trigger signal. In addition, the sample clock from the NI-DAQ is fed back into the Arduino. This way, if the sampling software stops running (*e.g.*, due to a software bug, or system restart), the NI-DAQ will no longer generate the sample clock, which will tell the Arduino to open the relay and stop the test. Opening the manual enable switch will also trigger the Arduino to stop the test. In this way, there are failsafes in place to stop the test due to (1) manual intervention, (2) hardware failure, or (3) software failure. Regardless of the cause, the batteries will be disconnected, data sampling will terminate, and an email will be sent alerting me of the issue.

Another issue that can arise in long tests, is corruption of the captured data stream due to dropped data buffers in the data collection PC. When the PC's CPU gets bogged down with other tasks, whole buffers full of sampled data can be lost, which results in the incorrect ordering of the pre-, during, and post-load measurements. Although a significant amount of buffering was added into the software, this issue can still arise and make it nearly impossible to sort out the data. As a result of this, one of the 32 channels contains a test fixture (schematic shown in Figure 5.4, bottom right) rather than battery test hardware. The data from the test fixture looks similar to that of a battery under test; however, the values of the resistors and capacitors were specifically chosen to guarantee the following conditions:

Manufacturer	Type	Chemistry	Nom. Voltage	Capacity	Shape	Diameter	Height	Weight
Panasonic	BR2325	Li-$(CF)_x$	3.0 V	165 mAh	Coin	23.0 mm	2.5 mm	3.13 g
Panasonic	CR2032	Li-MnO_2	3.0 V	225 mAh	Coin	20.0 mm	3.2 mm	2.85 g
Panasonic	CR2450	Li-MnO_2	3.0 V	620 mAh	Coin	24.5 mm	5.0 mm	6.27 g
Panasonic	1/2AA	Li-MnO_2	3.0 V	1000 mAh	Cylindrical	14.5 mm	25.2 mm	7.89 g
Energizer Ultimate	2xAAA	Li-FeS_2	3.0 V	1200 mAh	Cylindrical	10.5 mm	44.5 mm	8.55 g

Table 5.3: Battery data for those tested in Battery Life Experiment. Note that there are two Energizer Ultimate AAA batteries in series in order to achieve 3.0 V (*i.e.*, they are each 1.5 V). The dimensions and weight shown represent a single AAA battery.

1. $V_{\text{unload}} > V_{\text{load}}$
2. $V_{\text{unload}} > V_{\text{post}}$
3. $V_{\text{post}} > V_{\text{load}}$

Since these conditions are guaranteed, software in the data collection unit can check the incoming data to ensure that these conditions are met. If they are not met, then data buffers have been dropped, and the data stream can be repaired appropriately.

In this experiment, a number of different batteries were tested. Since SNUPI uses an MSP430 microcontroller, which operates between 1.8 and 3.6 V, only batteries which provide voltages in this range were tested. The only exception is the use of AAA cells, which are nominally 1.5 V. It is common in other WSNs to use two AAA batteries in order to provide long battery life and support high transmit power. As a result, we tested 2xAAA batteries, but considered this the limit in terms of battery size and weight. An important consideration is the shelf-life of the battery. Most alkaline batteries only have a shelf life of a few years, thus making them unsuitable for wireless sensor nodes which need to last a decade or more. Therefore, we only considered batteries which have a shelf life of at least 10 years. Most Lithium batteries operate at 3.0-3.6 V and have a shelf life of 10 years, and as a result this experiment exclusively uses Lithium batteries. Table 5.3 shows the specifications of the batteries chosen for this experiment. Other chemistries are available which can give batteries shelf-lives longer than 10 years, but their cost makes them prohibitive for

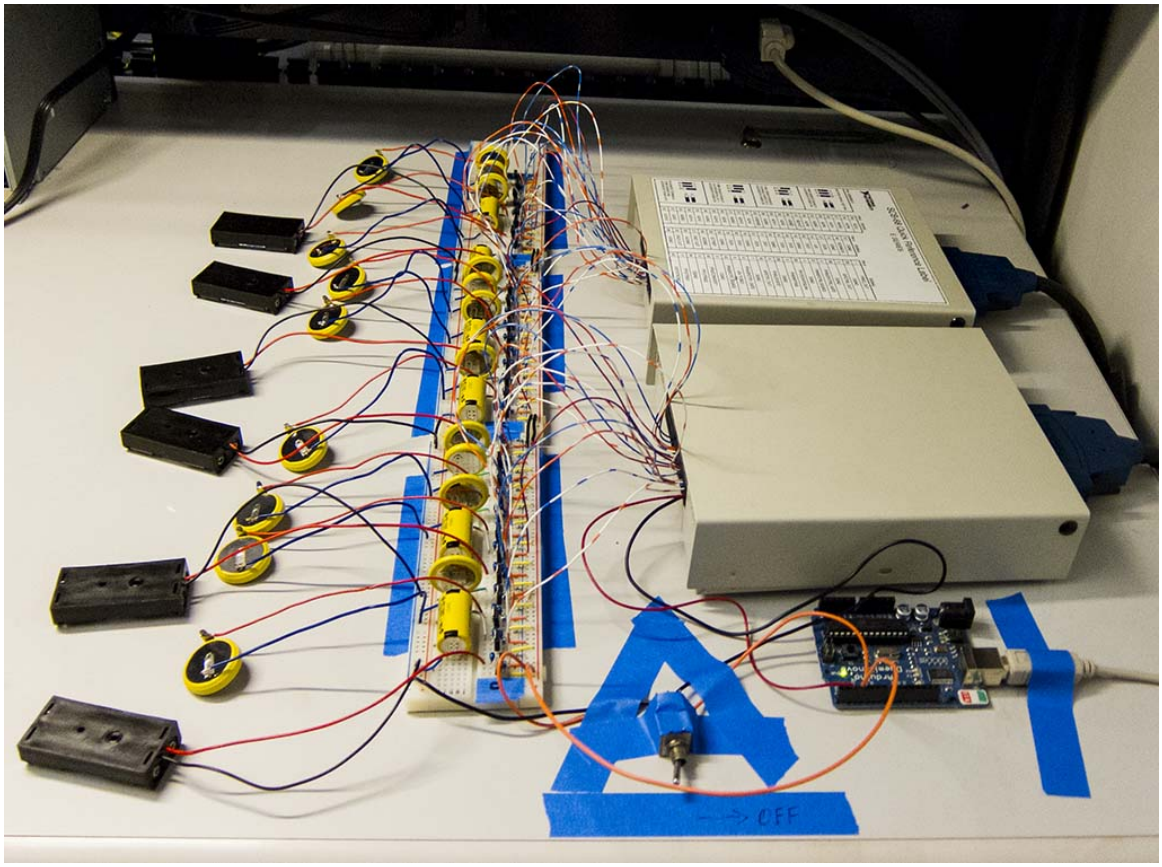

Figure 5.5: Picture of the Battery Life Experiment apparatus, showing all 31 batteries connected to the load modulation and data collection hardware.

most applications. However, these battery technologies are discussed in Section 5.1.3. Figure 5.5 shows a picture of the Battery Life Experiment apparatus with all batteries attached.

5.1.2 Battery Life Experiment Results

In the battery life experiment described in the previous section, the goal was to completely discharge a number of batteries under the same type of load expected during normal operation of a wireless sensor network. The experiment varied a number of parameters, including:

1. Battery: BR2325, CR2032, CR2450, 1/2AA, and 2xAAA

2. Transmit Current: 20 mA, 40 mA, 80 mA
3. Transmit Duration: 12 ms, 24 ms, 48 ms
4. Duty Cycle: 1.2%, 2.3%, 4.6%

To run a decent sized sample of batteries through each of the possible permutations of these variables would require over a year of testing. As a result, only a subset of these experiments were conducted. In addition, these experiments were unfortunately plagued by extremely bad luck. The first experiment, which was estimated to complete after 30 days, was halted on the 28th day by a direct lightening strike to a power sub-station, which took out the power in the University District for many hours. The second experiment, was also prematurely halted by a Windows Update. I had anticipated this being a problem, and therefore disabled Windows Update, and informed CSE support not to push any updates to the computer. However, someone in CSE support mistakenly changed the setting in their database and pushed an update to my computer, thus halting the second test. The third test was again halted by a power outage, but new experimental infrastructure (described in the previous section) allowed me to restart that test without any data loss as soon as power was restored. Finally, the fourth test was interrupted when another student needed to use the data collection hardware. Although this test has just been halted, and can be restarted whenever the data collection hardware is returned, it still has not been returned and may now be missing. In a total of over 4 continuous months of testing, only a single test completed, although much data was recovered from the prematurely halted tests. This has been a very good lesson in how to build robust, long-term experiments, and how to protect them from natural disasters, human error, and other students.

This section describes the analysis and conclusions that can be drawn from the available data extracted from these 4 months of testing. Table 5.4 lists the all of the completed battery tests that were available for analysis. The analysis in this section focuses on only the CR2450, 1/2AA, and 2xAAA batteries, because there was limited data available using the BR2325 and the CR2032. In addition, these smaller batteries are not well suited for long-life operation with transmit currents in the 10s of mA, as demonstrated by very short

Battery	Tx Duration	Sleep Duration	Duty Cycle	Tx Current	N	Lifetime (h)
BR2325	24 ms	500 ms	4.6 %	20 mA	3	210 ± 2
BR2325	24 ms	500 ms	4.6 %	40 mA	3	91 ± 2
BR2325	24 ms	500 ms	4.6 %	80 mA	2	0.016 ± 0.009
CR2032	24 ms	500 ms	4.6 %	20 mA	3	283 ± 7
CR2450	12 ms	250 ms	4.6 %	20 mA	7	658 ± 23
CR2450	12 ms	250 ms	4.6 %	40 mA	3	308 ± 18
CR2450	12 ms	250 ms	4.6 %	80 mA	3	140 ± 3
CR2450	24 ms	500 ms	4.6 %	40 mA	3	336 ± 3
CR2450	24 ms	500 ms	4.6 %	80 mA	3	136 ± 4
CR2450	24 ms	1000 ms	2.3 %	40 mA	3	632 ± 30
CR2450	24 ms	1000 ms	2.3 %	80 mA	3	305 ± 8
1/2AA	12 ms	250 ms	4.6 %	40 mA	6	650 ± 8
1/2AA	12 ms	250 ms	4.6 %	80 mA	3	336 ± 5
1/2AA	24 ms	500 ms	4.6 %	80 mA	3	324 ± 7
1/2AA	24 ms	1000 ms	2.3 %	80 mA	3	670 ± 24
2xAAA	12 ms	250 ms	4.6 %	40 mA	6	720 ± 9
2xAAA	12 ms	250 ms	4.6 %	80 mA	3	368 ± 4
2xAAA	24 ms	500 ms	4.6 %	80 mA	3	357 ± 5
2xAAA	24 ms	1000 ms	2.3 %	80 mA	3	717 ± 18

Table 5.4: Battery tests completed for analysis, where N is the number of battery tests completed under each set of parameters: battery, transmit duration (t_{tx}), sleep duration (t_{sleep}), duty cycle, and transmit current (I_{tx}). The mean lifetime is also shown, where the uncertainty represents one standard deviation.

lifetimes. The analysis in this section attempts to achieve three main goals:

1. To understand the discharge behavior of the batteries tested, and the effects of each of the variable parameters
2. To determine the best model for predicting the lifetime of a battery given each of the variable parameters
3. To determine an accurate estimate of the remaining battery life based on measurements taken from the battery at an unknown point in its discharge curve

This section is therefore divided into three main sections, one for each of these goals.

Battery Discharge Behavior

The first thing to look for are differences in the discharge curves amongst different batteries. Since every battery tested has a different lifespan, it is difficult to compare the discharge curves without normalizing the time scale. For this reason, all of the plots in this section will have "Percent Battery Life Remaining" on the x-axis. The lifetime of each battery is defined as the point at which any of the voltages measured drops below 1.8 V. This threshold of 1.8 V was chosen because it is the cutoff threshold for the MSP430 microcontroller that SNUPI nodes use. Note that practically, it is always the loaded voltage that crosses this threshold first. The percent battery life remaining is computed as the percent of time remaining until the point at which the loaded voltage drops below 1.8 V. As shown in Figure 5.6, all three battery types tested show quite different discharge curves for both the unloaded and loaded voltages. The discharge curves exhibited by the 1/2AA and 2xAAA batteries are preferred to that of the CR2450. This is because they remain relatively stable until the last 10% of life. On the other hand, the loaded voltage of the CR2450 gradually slopes downward throughout the life of the battery. This means that the available power and performance gradually degrade over time. It should also be noted that the unloaded voltage of the CR2450 remains relatively stable through the end of life, and also remains rather high in voltage. This suggests that the battery still has considerable charge remaining which is not useful because the loaded voltage has dropped too low for the circuitry. In other words,

the load current (*i.e.*, 40 or 80 mA) is too high for this battery, so its lifetime has been shortened.

The next trend to explore is the effect of the transmit and sleep duration, as well as the duty cycle. Past literature has claimed that using pulsed loads rather than continuous loads can improve battery life by giving the battery a period of time to recover [38, 39]. During this recovery time, the unloaded voltage increases, and more charge can therefore be pulled out of the battery before it is dead. Figure 5.7 (*top, left*) also shows this behavior of an increased unloaded voltage when the battery is given more time to recover. However, this effect is not seen in the loaded voltage, which remains the same across different recovery times and duty cycles, as shown in Figure 5.7 (*top, right*). Although Figure 5.7 (*top*) only shows data for the CR2450 with an 80 mA load, the same trends exist with a 40 mA load. In addition, the same trend can be seen for the 1/2AA battery, as shown in Figure 5.7 (*middle*). Again, the figure only shows data using an 80 mA load, but the same holds true with a 40 mA load. The 2xAAA battery however does not follow this trend, as shown in Figure 5.7 (*bottom*).

As one would expect, the size of the transmit load can have a significant impact on the discharge curves. As shown in Figure 5.8 (*top, right*), the loaded voltage discharge curves for the CR2450 can be clearly grouped by transmit current, with higher currents pulling the voltage lower. It can be noted here that these groupings are independent of the transmit and sleep times (and duty cycle). Note however that for the unloaded voltage (see Figure 5.8 (*top, left*)), the transmit and sleep times do come into play and it is not possible to group the discharge curves purely based on transmit current. The same trend can be seen in Figure 5.8 (*middle*) for the 1/2AA, and in Figure 5.8 (*bottom*) for 2xAAA batteries.

Battery Lifetime Modeling

It is important to be able to accurately model the expected lifetime of a battery in order to know how long a sensor node will live in the wild. Since a perfectly accurate model is not possible, it's also very important to know how much margin needs to be put into a battery

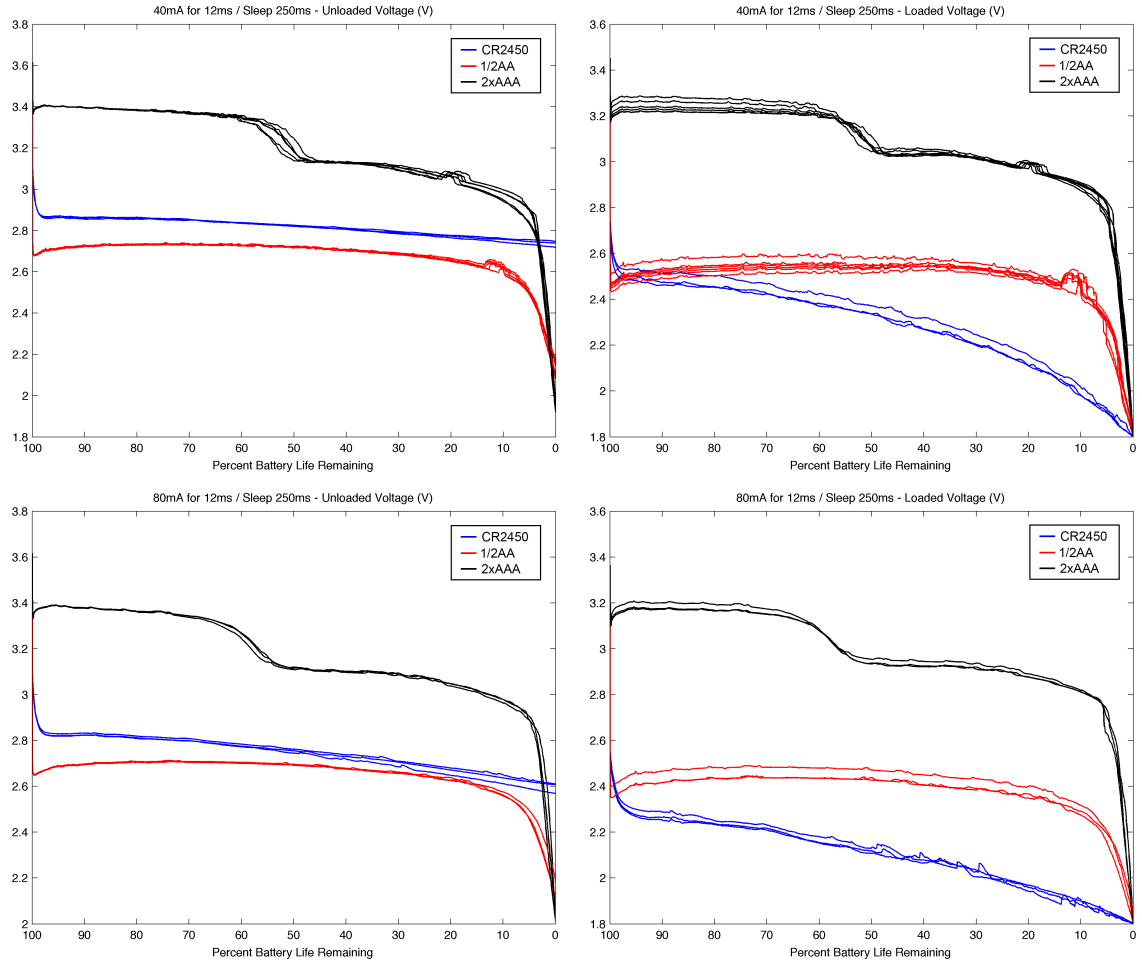

Figure 5.6: Battery discharge characteristics for different battery types. The plots on the left side show the unloaded voltage, and the right shows the loaded voltage. The top plots use a 40 mA load, while the bottom plots use an 80 mA load. Note that the transmit duration (t_{tx}) and sleep duration (t_{sleep}) are fixed at 12 ms and 250 ms, respectively.

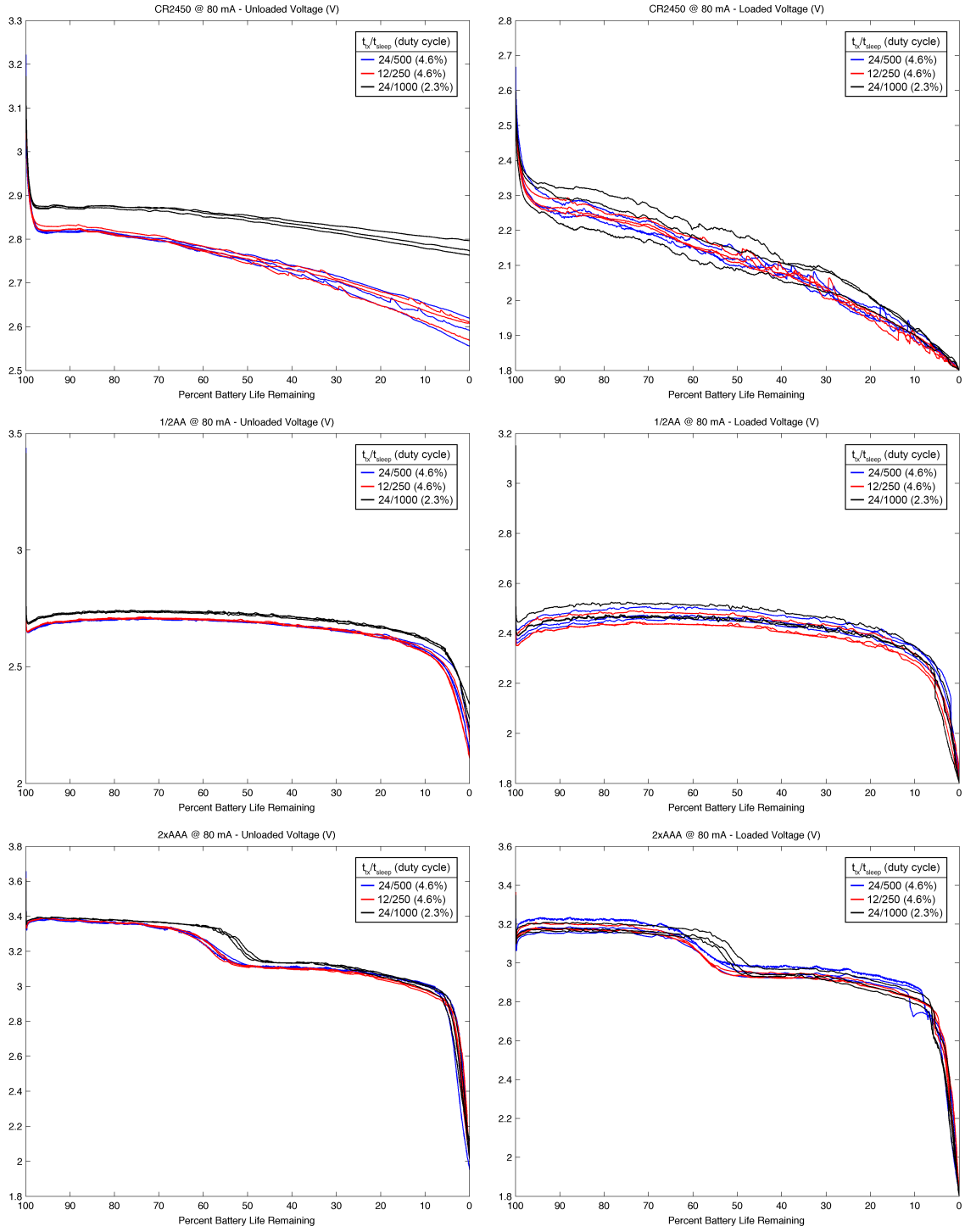

Figure 5.7: Battery discharge characteristics for different transmit and sleep durations. The plots on the left side show the unloaded voltage, and the right shows the loaded voltage. The top plots use CR2450 batteries, the middle plot use 1/2AA batteries, and the bottom plots use 2xAAA batteries. All batteries in these plots use 80 mA loads, although the same trends can be seen when using a 40 mA load.

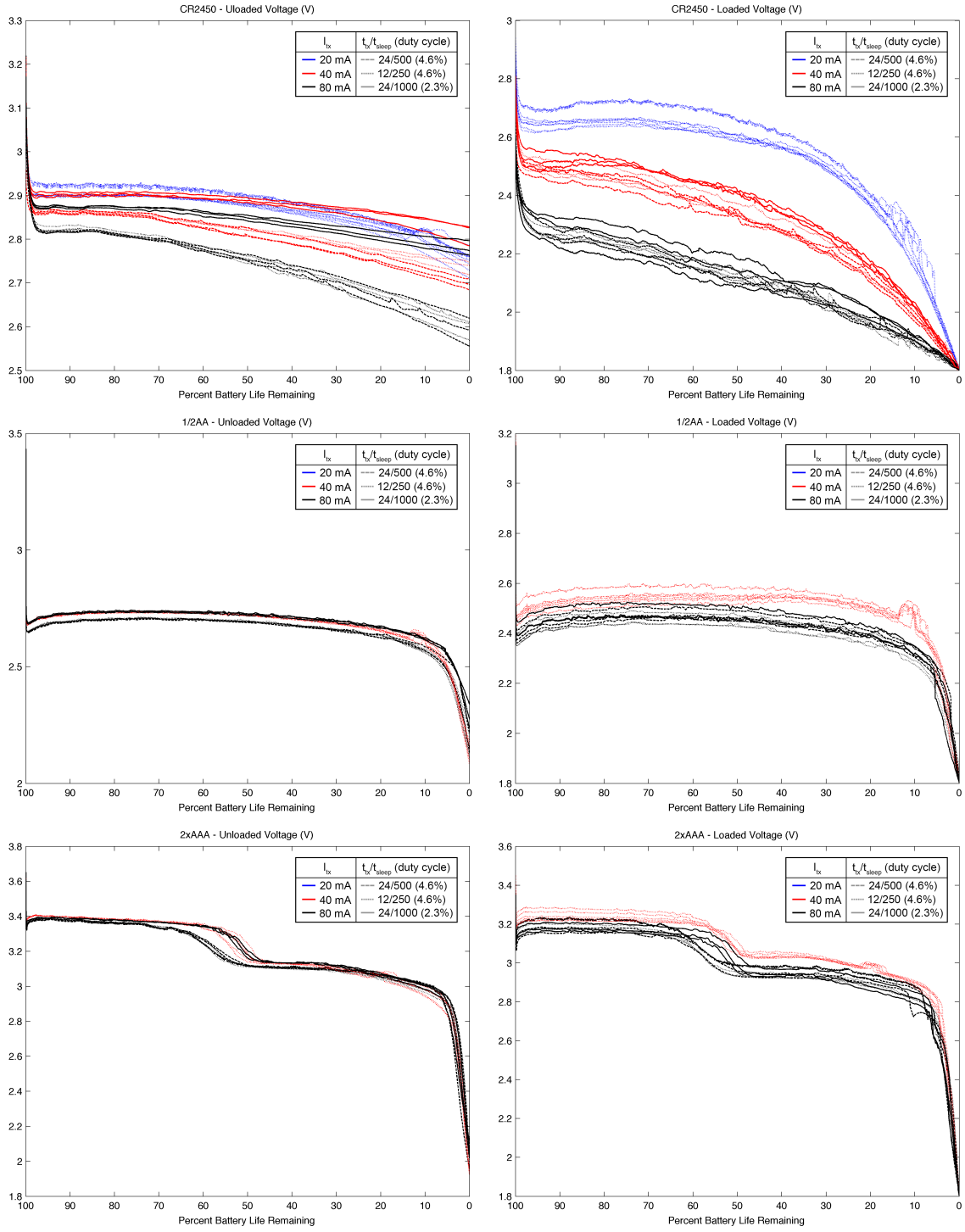

Figure 5.8: Battery discharge characteristics for different transmit loads. The plots on the left side show the unloaded voltage, and the right shows the loaded voltage. The top plots use CR2450 batteries, the middle plot use 1/2AA batteries, and the bottom plots use 2xAAA batteries. The color indicates the transmit load used, and the line style indicates the transmit and sleep durations.

model to ensure that a sensor node meets it's lifetime specification. In practice, most sensor nodes do not live anywhere near as long as they were predicted, and these errors are almost always in the modeling of the battery life. This section therefore proposes several different models for predicting battery lifetime, and then tests these models using the experimental data collected in the battery life experiment described in Section 5.1.1.

A full list of the battery life models attempted can be found in Table 5.5. First, they can be divided into two classes. The first class computes the lifetime L in terms of the battery capacity Q (*i.e.*, the total charge stored in the battery), and the average current I_{avg}. Using these two parameters, the battery life can be computed using:

$$L = \frac{Q}{I_{\text{avg}}}$$

The first 6 models shown in Table 5.5 use this method for computing battery life. Either the nominal battery capacity can be used (Q_{nom}), or a measured value of the battery capacity (Q_{meas}) can be used. The nominal capacity is listed on the battery's datasheet, and was used in models 1–3 in Table 5.5. For this analysis, the measured capacity was determined by summing the charge extracted from the battery throughout its lifetime:

$$Q_{\text{meas}} = \sum_{\text{i=1}}^{N} \left(\frac{V_{\text{load},i}}{R_{\text{load}}} t_{\text{tx}} + \frac{V_{\text{unload},i}}{R_{\text{sleep}}} t_{\text{sleep}} \right)$$

The measured capacity was used in models 4–6 in Table 5.5. Models 1–6 also rely upon an estimate of the average current, I_{avg}, which is most generally computed using two different states of the system: (1) transmitting, when the load is in the circuit, and (2) sleep, when only a small sleep current is flowing. The general equation for the average current is

$$I_{\text{avg}}(V_{\text{load}}, V_{\text{unload}}) = \frac{V_{\text{load}}}{R_{\text{load}}} \left(\frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{sleep}}} \right) + \frac{V_{\text{unload}}}{R_{\text{sleep}}} \left(\frac{t_{\text{sleep}}}{t_{\text{tx}} + t_{\text{sleep}}} \right)$$

In this equation, V_{load} is the average battery voltage under load, and V_{unload} is the average battery voltage in sleep (*i.e.*, unloaded). The parameter R_{load} represents the constant resis-

tance load that used to simulate the transmit current, and R_{sleep} is the large sleep resistor that was used to simulate the sleep current. Since small changes in sleep current could result in large errors for such long, low duty cycle experiments, the sleep current was accurately measured per channel of the test hardware and this value was used to compute an equivalent value of R_{sleep}. The times, t_{tx} and t_{sleep} are the transmit and sleep durations, respectively. For models 1 and 4, the nominal voltage V_{nom} is used for both V_{load} and V_{unload} to estimate I_{avg}. Models 2 and 5, use a computed value which is the average value of the battery voltage through the duration of the test. This value is calculated using the average values $\langle V_{\text{load}} \rangle$ and $\langle V_{\text{unload}} \rangle$ that were measured during the test:

$$\langle V_{\text{load}} \rangle = \frac{1}{N} \sum_{\text{i=1}}^{N} V_{\text{load},i}$$

$$\langle V_{\text{unload}} \rangle = \frac{1}{N} \sum_{\text{i=1}}^{N} V_{\text{unload},i}$$

$$V_{\text{avg}} = \langle V_{\text{load}} \rangle \left(\frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{sleep}}} \right) + \langle V_{\text{unload}} \rangle \left(\frac{t_{\text{sleep}}}{t_{\text{tx}} + t_{\text{sleep}}} \right)$$

Finally, models 3 and 6 used the average values of V_{load} and V_{unload} directly in the model, and should represent the most accurate model of the lifetime.

A second set of models, numbered 7–9 in Table 5.5, use the total energy stored in the battery rather than its capacity. Since this is not a value provided on the datasheet, we can only use a measured value from our experiments. This value, E_{meas} is computed similar to that of Q_{meas}:

$$E_{\text{meas}} = \sum_{\text{i=1}}^{N} \left(\frac{V_{\text{load},i}^2}{R_{\text{load}}} t_{\text{tx}} + \frac{V_{\text{unload},i}^2}{R_{\text{sleep}}} t_{\text{sleep}} \right)$$

In order to compute the battery life using the energy stored in the battery, the average power needs to be used instead of the average current:

$$P_{\text{avg}}(V_{\text{load}}, V_{\text{unload}}) = \frac{V_{\text{load}}^2}{R_{\text{load}}} \left(\frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{sleep}}} \right) + \frac{V_{\text{unload}}^2}{R_{\text{sleep}}} \left(\frac{t_{\text{sleep}}}{t_{\text{tx}} + t_{\text{sleep}}} \right)$$

#	Lifetime Model	Equations	
1	$Q_{\text{nom}} \mid V_{\text{nom}}$	$L = \frac{Q_{\text{nom}}}{I_{\text{avg}}(V_{\text{nom}})}$	$I_{\text{avg}}(V_{\text{nom}}) = \frac{V_{\text{nom}}}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{nom}}}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
2	$Q_{\text{nom}} \mid V_{\text{avg}}$	$L = \frac{Q_{\text{nom}}}{I_{\text{avg}}(V_{\text{avg}})}$	$I_{\text{avg}}(V_{\text{avg}}) = \frac{V_{\text{avg}}}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{avg}}}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
3	$Q_{\text{nom}} \mid V_{\text{states}}$	$L = \frac{Q_{\text{nom}}}{I_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle)}$	$I_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle) = \frac{\langle V_{\text{load}}\rangle}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{\langle V_{\text{unload}}\rangle}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
4	$Q_{\text{meas}} \mid V_{\text{nom}}$	$L = \frac{Q_{\text{meas}}}{I_{\text{avg}}(V_{\text{nom}})}$	$I_{\text{avg}}(V_{\text{nom}}) = \frac{V_{\text{nom}}}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{nom}}}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
5	$Q_{\text{meas}} \mid V_{\text{avg}}$	$L = \frac{Q_{\text{meas}}}{I_{\text{avg}}(V_{\text{avg}})}$	$I_{\text{avg}}(V_{\text{avg}}) = \frac{V_{\text{avg}}}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{avg}}}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
6	$Q_{\text{meas}} \mid V_{\text{states}}$	$L = \frac{Q_{\text{meas}}}{I_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle)}$	$I_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle) = \frac{\langle V_{\text{load}}\rangle}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{\langle V_{\text{unload}}\rangle}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
7	$E_{\text{meas}} \mid V_{\text{nom}}$	$L = \frac{E_{\text{meas}}}{P_{\text{avg}}(V_{\text{nom}})}$	$P_{\text{avg}}(V_{\text{nom}}) = \frac{V_{\text{nom}}^2}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{nom}}^2}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
8	$E_{\text{meas}} \mid V_{\text{avg}}$	$L = \frac{E_{\text{meas}}}{P_{\text{avg}}(V_{\text{avg}})}$	$P_{\text{avg}}(V_{\text{avg}}) = \frac{V_{\text{avg}}^2}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{V_{\text{avg}}^2}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$
9	$E_{\text{meas}} \mid V_{\text{states}}$	$L = \frac{E_{\text{meas}}}{P_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle)}$	$P_{\text{avg}}(\langle V_{\text{load}}\rangle, \langle V_{\text{unload}}\rangle) = \frac{\langle V_{\text{load}}\rangle^2}{R_{\text{load}}}\left(\frac{t_{\text{tx}}}{t_{\text{tx}}+t_{\text{sleep}}}\right) + \frac{\langle V_{\text{unload}}\rangle^2}{R_{\text{sleep}}}\left(\frac{t_{\text{sleep}}}{t_{\text{tx}}+t_{\text{sleep}}}\right)$

Table 5.5: Equations and parameters used for all battery lifetime models.

In order to compute the lifetime using E_{meas} and P_{avg}:

$$L = \frac{E_{\text{meas}}}{P_{\text{avg}}}$$

Just as with the charge/current based models, there are three different sets of voltages that can be used to estimate the average power: V_{nom}, V_{avg}, and finally both $\langle V_{\text{load}}\rangle$ and $\langle V_{\text{unload}}\rangle$.

Table 5.6 contains all of the nominal and measured parameters from the tested batteries, including Q_{nom}, V_{nom}, V_{avg}, $\langle V_{\text{unload}}\rangle$, $\langle V_{\text{load}}\rangle$, Q_{meas}, E_{mean}, and the measured lifetime L_{meas}. All values with uncertainties are measured values, and the values reported are the mean value over the N batteries tested, while the uncertainty represents the standard deviation. Table 5.7 shows the results of the lifetime model comparison. Each column represents a different model from Table 5.5 for estimating the battery life. The values in the table represent the mean percent error of the predicted model compared to the measured battery lifetime shown in Table 5.6, and the uncertainty represents one standard deviation. The last row contains the aggregate absolute mean percent error across all batteries tested for each model, and serves as a metric for comparing the accuracy of each model.

Battery	$t_{tx} \mid t_{sleep}$ (ms)	Duty Cycle	I_{tx} (mA)	N	Q_{nom} (mAh)	V_{nom} (V)	V_{avg} (V)	$\langle V_{unload} \rangle$ (V)	$\langle V_{load} \rangle$ (V)	Q_{meas} (mAh)	E_{meas} (Wh)	L_{meas} (h)
CR2450	12 | 250	4.6 %	20	7	620	3.0	2.9 ± 0.009	2.9 ± 0.009	2.5 ± 0.021	509 ± 17	1.29 ± 0.044	658 ± 22.6
CR2450	12 | 250	4.6 %	40	3	620	3.0	2.8 ± 0.003	2.8 ± 0.002	2.3 ± 0.022	433 ± 25	1.00 ± 0.060	308 ± 18.1
CR2450	12 | 250	4.6 %	80	3	620	3.0	2.7 ± 0.009	2.7 ± 0.010	2.1 ± 0.010	361 ± 7	0.76 ± 0.015	140 ± 2.6
CR2450	24 | 500	4.6 %	40	3	620	3.0	2.8 ± 0.006	2.8 ± 0.007	2.3 ± 0.008	470 ± 6	1.08 ± 0.017	336 ± 3.5
CR2450	24 | 500	4.6 %	80	3	620	3.0	2.7 ± 0.009	2.7 ± 0.010	2.1 ± 0.015	350 ± 13	0.74 ± 0.033	136 ± 4.3
CR2450	24 | 1000	2.3 %	40	3	620	3.0	2.9 ± 0.007	2.9 ± 0.007	2.3 ± 0.002	466 ± 22	1.10 ± 0.051	632 ± 29.8
CR2450	24 | 1000	2.3 %	80	3	620	3.0	2.8 ± 0.006	2.8 ± 0.007	2.1 ± 0.038	408 ± 18	0.87 ± 0.053	305 ± 8.1
1/2AA	12 | 250	4.6 %	40	6	1000	3.0	2.7 ± 0.003	2.7 ± 0.003	2.5 ± 0.021	995 ± 12	2.49 ± 0.042	650 ± 8.4
1/2AA	12 | 250	4.6 %	80	3	1000	3.0	2.6 ± 0.005	2.7 ± 0.005	2.4 ± 0.025	987 ± 20	2.37 ± 0.067	336 ± 5.3
1/2AA	24 | 500	4.6 %	80	3	1000	3.0	2.6 ± 0.004	2.7 ± 0.005	2.4 ± 0.022	962 ± 12	2.33 ± 0.010	324 ± 6.8
1/2AA	24 | 1000	2.3 %	80	3	1000	3.0	2.7 ± 0.003	2.7 ± 0.003	2.4 ± 0.029	1020 ± 25	2.48 ± 0.036	670 ± 24.1
2xAAA	12 | 250	4.6 %	40	6	1200	3.0	3.2 ± 0.007	3.2 ± 0.008	3.1 ± 0.012	1355 ± 19	4.18 ± 0.071	720 ± 9.0
2xAAA	12 | 250	4.6 %	80	3	1200	3.0	3.2 ± 0.001	3.2 ± 0.002	3.0 ± 0.013	1346 ± 10	4.04 ± 0.024	368 ± 3.9
2xAAA	24 | 500	4.6 %	80	3	1200	3.0	3.2 ± 0.005	3.2 ± 0.006	3.0 ± 0.024	1312 ± 11	3.95 ± 0.034	357 ± 5.2
2xAAA	24 | 1000	2.3 %	80	3	1200	3.0	3.2 ± 0.007	3.2 ± 0.006	3.0 ± 0.024	1352 ± 25	4.07 ± 0.049	717 ± 18.4

Table 5.6: Nominal and measured parameters for tested batteries. All values with uncertainties are measured values, and the values reported are the mean value over the N batteries tested, while the uncertainty represents the standard deviation. These parameters are used in computing the battery lifetime models, which are compared in Table 5.7.

Battery Parameters				Mean Percent Error for Each Model Compared to Measured Lifetime ± One Standard Deviation								
Battery	**t_{tx} | t_{sleep} (ms)**	**Duty Cycle**	**I_{tx} (mA)**	**Q_{nom} | V_{nom}**	**Q_{nom} | V_{avg}**	**Q_{nom} | V_{states}**	**Q_{meas} | V_{nom}**	**Q_{meas} | V_{avg}**	**Q_{meas} | V_{states}**	**E_{meas} | V_{nom}**	**E_{meas} | V_{avg}**	**E_{meas} | V_{states}**
CR2450	12 | 250	4.6 %	20	$+2.2 \pm 3.5$	$+7.6 \pm 3.5$	$+21.9 \pm 4.1$	-16.1 ± 0.7	-11.7 ± 0.5	$-1.2 \times 10^{-12} \pm 2.5 \times 10^{-12}$	-29.2 ± 1.2	-21.6 ± 0.9	$+0.7 \pm 0.04$
CR2450	12 | 250	4.6 %	40	$+9.9 \pm 6.5$	$+18.0 \pm 6.9$	$+43.6 \pm 8.6$	-23.4 ± 0.7	-17.8 ± 0.7	$+6.7 \times 10^{-14} \pm 3.8 \times 10^{-12}$	-41.0 ± 1.1	-31.9 ± 1.2	$+0.7 \pm 0.03$
CR2450	12 | 250	4.6 %	80	$+20.5 \pm 2.3$	$+33.2 \pm 2.6$	$+71.7 \pm 3.1$	-29.8 ± 0.3	-22.4 ± 0.6	$+9.2 \times 10^{-14} \pm 1.2 \times 10^{-12}$	-50.5 ± 0.5	-39.5 ± 1.0	$+0.5 \pm 0.04$
CR2450	24 | 500	4.6 %	40	$+0.5 \pm 1.0$	$+8.5 \pm 0.9$	$+32.0 \pm 1.7$	-23.9 ± 0.3	-17.8 ± 0.4	$+6.2 \times 10^{-12} \pm 5.9 \times 10^{-12}$	-41.6 ± 0.4	-32.0 ± 0.8	$+0.7 \pm 0.09$
CR2450	24 | 500	4.6 %	80	$+24.3 \pm 3.9$	$+37.5 \pm 4.2$	$+77.2 \pm 6.7$	-29.8 ± 0.5	-22.4 ± 0.8	$+2.0 \times 10^{-12} \pm 9.3 \times 10^{-13}$	-50.5 ± 0.7	-39.4 ± 1.2	$+0.4 \pm 0.02$
CR2450	24 | 1000	2.3 %	40	$+4.0 \pm 4.8$	$+8.8 \pm 4.8$	$+33.3 \pm 6.1$	-22.0 ± 0.1	-18.3 ± 0.2	$+7.3 \times 10^{-12} \pm 6.2 \times 10^{-12}$	-38.6 ± 0.1	-32.8 ± 0.4	$+0.7 \pm 0.04$
CR2450	24 | 1000	2.3 %	80	$+7.8 \pm 2.9$	$+14.4 \pm 2.9$	$+52.1 \pm 6.8$	-29.1 ± 1.3	-24.7 ± 1.5	$+2.4 \times 10^{-13} \pm 3.3 \times 10^{-12}$	-49.5 ± 1.8	-43.1 ± 2.3	$+0.4 \pm 0.08$
1/2AA	12 | 250	4.6 %	40	-16.3 ± 1.1	-6.0 ± 1.3	$+0.5 \pm 1.3$	-16.7 ± 0.7	-6.4 ± 0.8	$+3.3 \times 10^{-12} \pm 6.1 \times 10^{-12}$	-30.4 ± 1.1	-12.2 ± 1.4	$+0.2 \pm 0.02$
1/2AA	12 | 250	4.6 %	80	-19.2 ± 1.3	-8.2 ± 1.6	$+1.3 \pm 2.1$	-20.2 ± 0.8	-9.4 ± 1.0	$+4.7 \times 10^{-12} \pm 6.7 \times 10^{-13}$	-36.3 ± 1.3	-17.7 ± 1.8	$+0.2 \pm 0.03$
1/2AA	24 | 500	4.6 %	80	-16.2 ± 1.7	-4.9 ± 2.0	$+4.0 \pm 1.3$	-19.4 ± 0.7	-8.5 ± 0.9	$-2.6 \times 10^{-12} \pm 1.0 \times 10^{-11}$	-35.0 ± 1.2	-16.2 ± 1.6	$+0.2 \pm 0.02$
1/2AA	24 | 1000	2.3 %	80	-20.8 ± 2.9	-11.5 ± 3.3	-2.0 ± 2.5	-19.2 ± 1.0	-9.8 ± 1.1	$-1.3 \times 10^{-12} \pm 6.7 \times 10^{-12}$	-34.6 ± 1.5	-18.4 ± 1.9	$+0.2 \pm 0.02$
2xAAA	12 | 250	4.6 %	40	-9.3 ± 1.1	-14.6 ± 0.9	-11.5 ± 1.3	$+2.4 \pm 0.4$	-3.5 ± 0.5	$+4.1 \times 10^{-12} \pm 4.7 \times 10^{-12}$	$+5.4 \pm 0.8$	-6.5 ± 1.1	$+0.5 \pm 0.01$
2xAAA	12 | 250	4.6 %	80	-11.3 ± 0.9	-15.7 ± 0.9	-10.9 ± 0.7	-0.5 ± 0.4	-5.4 ± 0.4	$+2.6 \times 10^{-12} \pm 4.1 \times 10^{-12}$	-0.6 ± 0.8	-10.1 ± 0.8	$+0.5 \pm 0.03$
2xAAA	24 | 500	4.6 %	80	-8.6 ± 1.3	-13.2 ± 1.3	-8.5 ± 0.8	-0.1 ± 0.8	-5.1 ± 0.9	$+2.2 \times 10^{-12} \pm 2.4 \times 10^{-12}$	$+0.3 \pm 1.7$	-9.5 ± 1.8	$+0.5 \pm 0.08$
2xAAA	24 | 1000	2.3 %	80	-11.3 ± 2.2	-16.6 ± 1.9	-11.2 ± 1.6	-0.1 ± 0.8	-6.0 ± 0.6	$+3.9 \times 10^{-12} \pm 3.0 \times 10^{-12}$	$+0.4 \pm 1.6$	-11.3 ± 1.1	$+0.5 \pm 0.05$
			Aggregate Errors:	11.7 ± 7.1	13.6 ± 9.0	23.1 ± 23.5	16.1 ± 10.1	11.7 ± 6.9	$4.2 \times 10^{-12} \pm 3.1 \times 10^{-12}$	28.4 ± 17.2	21.3 ± 11.9	0.5 ± 0.19

Table 5.7: Battery lifetime model comparison. Each column labeled $X \mid Y$ represents a different model from Table 5.5 for estimating the battery life using parameters X and Y. The values in the table represent the mean percent error of the predicted model compared to the measured battery lifetime shown in Table 5.6. The uncertainty represents one standard deviation. The last row contains the aggregate absolute mean percent error across all batteries tested for each model, and serves as a metric for comparing the accuracy of each model.

The first conclusion to draw from these results is that models which use a multi-state model of the voltage preform much better than models that do not. This is not surprising as the loaded and unloaded voltages are quite different. Luckily, it is not difficult to simply measure the loaded and unloaded voltages without running a full discharge test. The next conclusion to draw is that it is important to measure or model the capacity Q, as the nominal value can be quite inaccurate. For example, in the case of the CR2450, the nominal value was quite a bit higher than the measured value for all conditions tested. An the other hand, the nominal value was rather low for the 2xAAA, and was just about right for the 1/2AA. It's therefore important to either measure the capacity, which requires a full discharge, or model it in some way.

Modeling the capacity is also needed when the available capacity is a strong function of the discharge parameters (*i.e.*, load duration, duty cycle, and load current). The best way to identify which parameters effect the model in an unanticipated way is look at the Q_{meas} values for a single battery type. For each battery type, the value of Q should be the same, and any deviation is due to unmodeled effects. For example, it is clear from the data that the transmit duration has an effect on the available capacity and therefore the lifetime of the battery even when the load current and duty cycle are the same. In other words, there are damaging effects from the high current output that are not currently being modeled by purely looking at the charge or energy being delivered. Unfortunately, we don't currently have enough data to derive these models, but in the future I would like to collect more data to try to build a model for each battery that estimates the available capacity as a function of transmit duration, duty cycle, and transmit current.

It can also be seen from the data that the models which use the battery capacity are more accurate than those which use the energy stored in the battery. Since the same discharge test is needed to measure either value, it's best to use the capacity based model. It should also be noted that the accuracy of the $Q_{\text{meas}} \mid V_{\text{states}}$ model is not nearly as good as shown in the table. The incredible accuracy in the table comes from the fact that the method used to calculate Q_{meas} is very similar to the method used to compute the lifetime model. The

accuracy of the model in practice will be limited by the accuracy of the estimated Q, and the variation in the voltages over the battery's lifetime.

Another important note to make is that all of these models and the discharge tests performed assume a constant resistance load. This may not be the case; in fact, many RF radios consume constant current rather than presenting a constant resistance. In the future, I'd also like to explore models for constant current loads. Another important limitation of this work is that there is no control or modeling for ambient temperature. It is well known that battery voltage and lifetimes vary significantly as a function of temperature. In fact, I believe that some of the anomalous jumps in battery voltage that were seen in the discharge experiments were likely due to changes in ambient temperature. In the future, I'd also like to further explore the effect of temperature on the battery lifetime as well.

Estimating Remaining Battery Life

In addition to modeling the expected lifetime of a battery is also important to develop methods to allow sensor nodes to make measurements on their own batteries in order to estimate how much battery life is remaining. This is useful for example so that sensor nodes can send packets alerting the network before their batteries die. By looking at Figure 5.8, it may seem reasonable to simply set a threshold on either the loaded or unloaded voltage to determine the remaining battery life. This approach may work fairly well for the 1/2AA or 2xAAA batteries who's discharge curves don't vary much as a function of the discharge parameters, and who's end of life voltage dropout is very sudden. However, looking at the discharge curve for the CR2450, it's clear that it is not always possible to simply set a threshold on either the loaded or unloaded voltage to determine the battery life because the discharge curves vary greatly as a function of the discharge conditions.

Another potential metric would be to measure the loaded voltage drop, V_{drop} which gives an indication of the internal resistance of the battery.

$$V_{\text{drop}} = V_{\text{unload}} - V_{\text{load}}$$

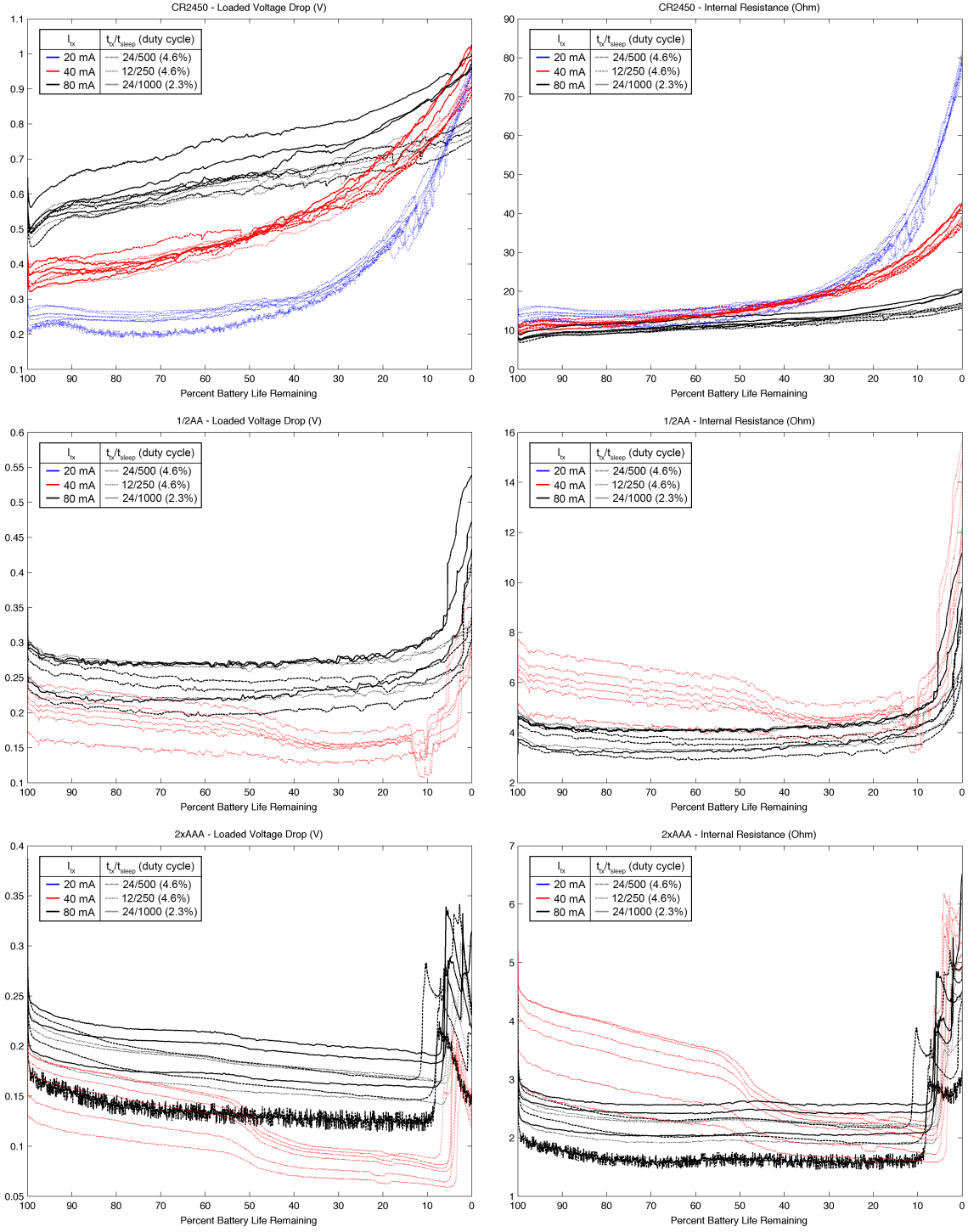

Figure 5.9: Internal resistance characteristics for different transmit loads. The plots on the left side show the loaded voltage drop, and the right shows the internal resistance. The top plots use CR2450 batteries, the middle plot use 1/2AA batteries, and the bottom plots use 2xAAA batteries. The color indicates the transmit load used, and the line style indicates the transmit and sleep durations.

The theory is that the internal resistance will increase greatly near the end of life, and therefore it would be easier to set a threshold to determine the end of life. As shown in Figure 5.9 (*left*), using the loaded voltage drop is really no better than simply using the loaded voltage. In fact, they are quite a bit worse as there is more variation from transmit time and duty cycle which effect the loaded voltage very little.

Since the idea behind using the voltage drop was to measure the internal resistance, it makes more sense to actually compute the internal resistance R_s since the current is known.

$$R_s = (V_{\text{unload}} - V_{\text{load}}) \frac{R_{\text{load}}}{V_{\text{load}}}$$

As shown in Figure 5.9 (*right*), this metric works much better than the loaded voltage drop, but still needs to be aware of the transmit current, and perhaps the duty cycle and transmit duration.

In conclusion, the best way to estimate the remaining battery life is to use the loaded voltage, as shown in Figure 5.8 (*right*).

5.1.3 Extended Shelf-life Batteries

All of the batteries tested in the above sections have a shelf-life of 10–15 years, and as a result they are an ideal choice for most ultra-low-power sensor nodes. However, in cases where more battery life is desired, there are batteries available that have even lower self-discharge rates. Most of these extended shelf-life batteries are made by Tadiran[2]. The most common extended shelf-life batteries are made of Li-MNO_2, and have a typical shelf life of 25 years. These are often used in consumer electronics devices that need to run or have battery backup for more than 10 years. There are also Li-$SOCl_2$ batteries which boast a 40 year shelf-life!

---

[2] `http://www.tadiranbat.com`

5.1.4 Supply Level Variation

The ideal battery discharge curve would be perfectly flat, and then fall off a cliff at the end of life. Unfortunately, as can be seen from Figure 5.8, this is rarely the case. The battery voltage tends to slowly decline over time, particular when using higher current loads. This steady reduction of the supply voltage corresponds to a steady reduction in performance. The power delivered to the antenna is usually a square function of the supply voltage, so as the voltage reduces there is a squared reduction in transmit power. This unfortunately results in sensor nodes losing network connection long before the batteries actually die.

In order to keep the transmitter performance constant throughout the life of the sensor node, engineers typically use a voltage regulator to keep the supply voltage constant. This is particularly important when there is supply voltage droop during a single packet transmission. This is common when pulling a high transmit current out of a coin-cell battery like the CR2450. Using a regulated supply however has several downsides. First, there is some drop-out voltage in the regulator, which essentially increases the voltage threshold at which the sensor node dies; effectively reducing the battery life. For this reason, you want to set your regulated voltage as close as possible to the minimum operating voltage of your transmitter. However, this ensures that your node will always operate at minimum transmit power. Finally, the regulator has some losses in the form of quiescent current. Although this loss is likely insignificant when compared to the load current of the transmitter, it can dominate the current consumption when in sleep. As a result, the regulator should ideally be duty cycled so that it is only on when transmitting, and off when in sleep. This however introduces significant energy losses in charging and discharging supply line capacitors in addition to excess power consumed during the startup stabilization of the regulated supply. As future work, I'd like to design a hybrid regulated/unregulated supply that optimally balances these constraints.

5.1.5 Harvesting Power Supplies

In order to achieve even longer lifetime, energy can be harvested from the environment either to prolong the life of an on-board battery, or to allow battery-free operation of a sensor node. With the extremely low-power consumption of SNUPI nodes, the potential for entirely battery-free sensor nodes that can send several packets per day of sensor data is becoming a reality. The end of Section 2.1.4 describes some of the related work on power harvesting for wireless sensor nodes including the use of backscatter communications. To date, there have been no implementations of SNUPI which use a harvested energy source, but I think this is a very exciting direction of future research!

5.2 Microcontroller

The microcontroller is the brain of the sensor node, and is crucial for timing control, computation, and the capture of analog sensor signals. It controls powering the sensors, sampling data, and both powering and modulating the transmitter. The RF transmitter is often powered directly from a digital output pin on the microcontroller so that the transmitter can be completely powered down during the sleep phase. In addition, the microcontroller can be used as a general computation platform which is often useful for channel coding as well as encryption. Performing computations on the data is also very useful in order to reduce the quantity of data to be sent over the network. Because computation power is typically less expensive that communications power, it is generally best to compress the data as much as possible before sending.

For ultra-low-power electronics, the most common microcontroller used today is the 16-bit Texas Instruments MSP430[3]. All of the SNUPI implementations to date have been built around the MSP430. Most of these designs use the MSP430F2xx series microcontroller, which boasts around 220 μA/MHz current consumption in active mode while only 0.7 μA in standby. The microcontroller can operate between 1.8 and 3.6 V, and up to 16 MHz. This is typically the best microcontroller to use when sleep current is most important, as is the

[3] http://www.msp430.com

case for long-life sensor nodes which spend the vast majority of the time in sleep.

The newer MSP430F5xx series is useful when either more computational power or more peripherals are needed. It contains a hardware multiplier, DMA, internal regulator, more timers, more serial interfaces, and can operate up to 25 MHz. The MSP430F5xx series also has lower active power than the MSP430F2xx series, but significantly higher standby power at 2.5 µA. It is because of this standby power that the MSP430F5xx series microcontrollers are less ideally suited for long-lived sensor nodes.

The newest line in the MSP430 family is the FRAM-based MSP430FR5xx series, which boasts both faster and lower power memory operations and RAM retention. The active current is also quite a bit lower at less than 100 µA/MHz. However, like the MSP430F5xx series, it suffers from a high standby current of 6.3 µA. In addition, it requires at least 2.0 V to operate, which effectively reduces the node's battery life. TI also has a low-voltage microcontroller, the MSP430L092, which operates between 0.9 and 1.65 V, making it optimal for running off of a single 1.5 V alkaline cell. It also has an extremely low power active mode at only 45 µA/MHz, yet suffers from a high 6 µA standby current. Unlike the other MSP430 microcontrollers, the MSP430L092 does not have a programmable ROM, and therefore program memory either needs to live in the 2 KB volatile SRAM for the entire life of the device, or be loaded into SRAM on power-up from an external ROM device.

The 8-bit Atmel AVR[4] is also commonly used on wireless sensor nodes, and has an active current comparable to the MSP430 at around 200 µA/MHz. However, the sleep power of the AVR is almost 10 times that of the MSP430 making it unsuitable for long-lived sensor nodes. However, the AVR is typically much easier to use, can operate at higher clock speeds, and has more peripherals. Microchip[5] also has a competitive product in the 8-bit PIC16LF1503, which like the MSP430 operates between 1.8 and 3.6 V at up to 20 MHz. It boasts an incredible 30 µA/MHz active current, and less than 0.03 µA in standby!

However, care should be taken in comparing the power numbers from datasheets at face

[4] http://www.atmel.com/products/microcontrollers/avr/

[5] http://www.microchip.com

value, even for multiple parts from the same manufacturer. Different datasheets use different terminology, different operating assumptions and different measurement techniques to obtain these values. In order to compare the power consumption of different microcontrollers, it is best to run controlled experiments using several different benchmark tasks.

5.3 Transmitter

The transmitter needs to produce the analog signals to deliver to the antenna in order to send the wireless data. This section describes several different hardware implementations. Section 5.3.1 describes a full-custom analog integrated circuit implementation of the transmitter which represents a lower bound on the power consumption. However, it comes at the cost of requiring a full custom IC. In addition to the cost, it increases the system complexity, and the prototype IC is very sensitive to static and other environmental conditions making it very fragile and therefore practically very difficult to test in the field without destroying it. Section 5.3.2 describes discrete implementations using both analog and digital commercial off-the-shelf parts. Although these implementations have a higher power consumption, they are much cheaper to prototype and build in low volumes, and can be very flexible for prototyping. Section 5.3.3 describes an implementation which relies on the hardware already inside the MSP430 microcontroller, and requires minimal additional components. Although this implementation dramatically reduces system complexity, component count, and cost, it consumes significantly more power than any of the alternatives.

Finally, Section 5.3.4 describes the software algorithms required to generate the modulation signal that drives the transmitter hardware. I present several methods for reducing the power consumption of generating the modulation signal. One of these methods is able to control the modulation bit during transmission without running the CPU at all.

5.3.1 Custom Analog IC Implementation

In order to minimize power consumption of the transmitter, I partnered with Jagdish Pandey and Brian Otis, who are experts in ultra-low-power analog integrated circuit design, to build

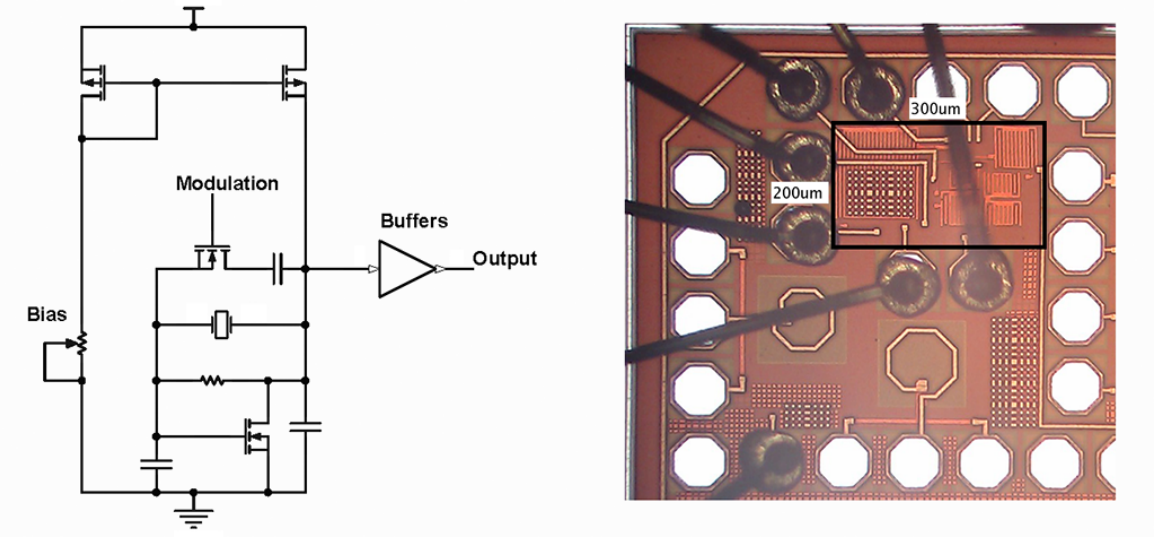

Figure 5.10: (*left*) Schematics of the custom FSK transmitter. (*right*) 20X microscope image of CMOS silicon die containing the custom transmitter. The transmitter circuitry is within the black rectangle.

a custom CMOS transmitter for the first SNUPI implementation in 2010. Our goal was to design a 27 MHz transmitter using minimum power. We implemented a binary frequency shift keying (BFSK) transmitter using a Pierce oscillator with a 27.0 MHz crystal resonator (schematics shown in Figure 5.10). To modulate the transmitter, a small 4 pF on-chip load capacitance across the crystal resonator is switched to cause a 10 kHz frequency shift. The crystal oscillator has a relatively slow startup time, between 1 and 4 ms, which varies as a function of the oscillator bias current such that the startup time is longer when the oscillator consumes less power.

A digital buffer chain isolates the oscillator from the low impedance ($\sim$ 200 Ω) loop antenna. In order to save power, the buffers use a very low (*i.e.*, sometimes even sub-threshold) supply voltage. By adjusting this supply voltage, the output power of the antenna can be varied by 18 dB. At the minimum output power, the radio consumes only 35 µW, and at the maximum output power the radio's power consumption is 190 µW. During modulation, the CPU also needs to be on in order to control the modulation input of the transmitter.

Including the power consumption of the microcontroller, the total node power consumption is between 900 μW and 1.5 mW, depending on the supply voltage for the output buffer. This power range is specified by the component values used in our prototype, and therefore the power range could be shifted or extended using different sized components.

This transmitter design can be made very low-power as long as the stray capacitance is not too large. Our implementation of the oscillator on a single silicon die using a 0.13 μm CMOS process resulted in only 65 μW of power for whole-home range. Figure 5.10 shows a microscope image of the transmitter die, which was wire-bonded to the custom SNUPI printed circuit board (PCB). The custom transmitter IC contains the transmit oscillator, FSK modulator, and transmit amplifier. Additional external components are required to control the power supply for the transmitter IC, impedance match to the loop antenna, and control the power consumption and power output of the transmitter. Although the use of this custom transmitter IC produces extremely low power consumption, it results in a high system complexity. In addition, the prototype custom transmitter IC is very sensitive to static and other environmental conditions making it very fragile and therefore practically very difficult to test in the field without destroying it.

5.3.2 Discrete Implementation

Although the custom analog IC implementation of the SNUPI transmitter pushes the boundary of ultra-low-power transmitter design, it requires custom silicon to be produced, and is therefore impractical if only small quantities are required. Using a similar Pierce oscillator design as shown in Figure 5.10, the oscillator and power amplifier can be implemented using discrete analog components. Figure 5.11 shown the schematic diagram of such a design using bipolar junction transistors (BJT). This implementation operated at about 120 μW using 2N3905s for Q_1 and Q_2, a BC547BG for Q_3, a PN3640 for Q_4, 9.96 kΩ R_1, 338.4 kΩ R_2, 1 MΩ R_3, and 100 kΩ R_4.

Although the transmitter is an analog circuit, it can also be implemented using commercial off-the-shelf (COTS) *digital* logic ICs. Although the power consumption will be

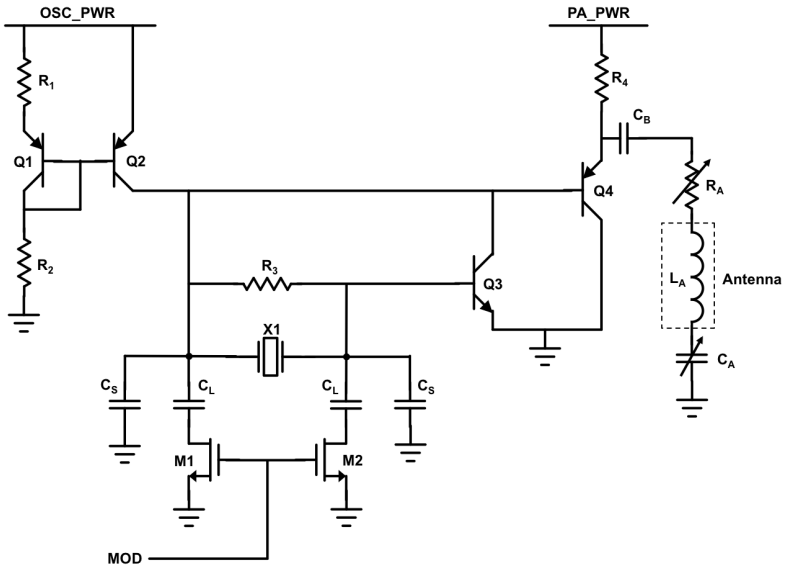

Figure 5.11: Discrete analog transistor implementation of transmitter.

greater than using an analog circuit design, these ICs are mass produced and are therefore very cheap. In addition, all of the transistors for both the oscillator and the power amplifier can be found in the same chip to reduce the part count. However, some external components are needed to make the digital logic behave as an oscillator or power amplifier. Since these ICs were produced for digital logic and not as efficient oscillators or amplifiers, they will consume significantly more power than is needed.

Figure 5.12 shows a schematic of an implementation of the transmitter in which both the oscillator and power amplifier use only 74AUP00 digital NAND gates (U1). The oscillator implemented in Figure 5.12 is also a Pierce oscillator, where the NAND gates are simply acting as inverting amplifiers. Using the second pin of the NAND gates as a control signal allows both the oscillator and the power amplifier to be turned on and off digitally. The crystal X1 can be at any frequency, although the existing SNUPI implementations all use 13.56 and 27.00 MHz. R_1 is typically 1 MΩ, and R_2 is typically 240 Ω. The load capacitors C_L are switched in and out of the circuit by the BF1212 low-capacitance dual-gate n-channel MOSFETS, M1 and M2. The load capacitors C_L can be up to about 47 pF and

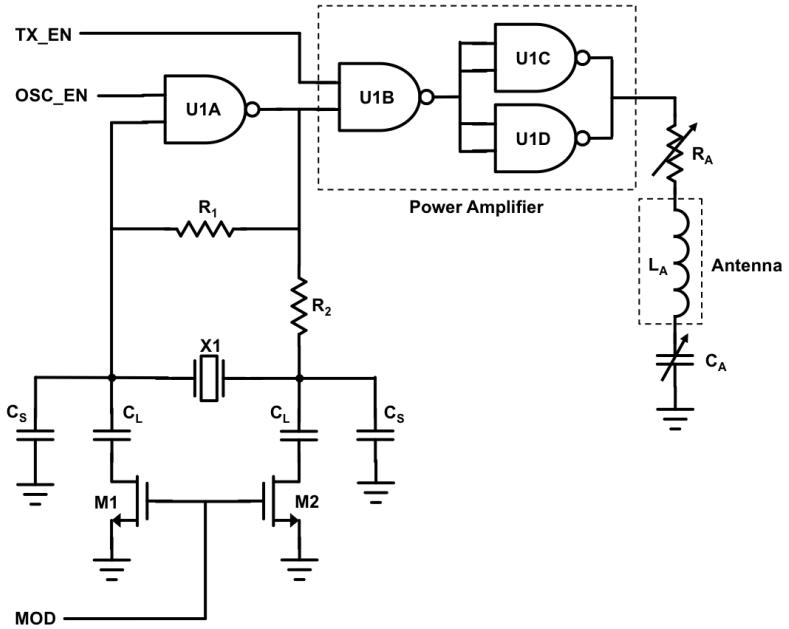

Figure 5.12: Discrete digital IC implementation of transmitter using 7400 NAND gates for both the oscillator and the power amplifier.

cause a frequency shift of up to 15 kHz. In addition, static load capacitors C_S can used to shift the center frequency slightly. Doing so however reduces the amount of frequency shift when modulated.

The NAND output of U1C and U1D would normally swing rail to rail unless a resonant tank is present at the output. To resonate the output, a series capacitor (C_A) is used along with the loop antenna, which is mostly inductive (L_A). A variable capacitor is used so that it can be manually tune the antenna into resonance at the desired center frequency. The transmit power can be varied by changing the value of the resistor in series with the loop antenna (R_A). This will limit the current into the antenna, but will also reduce the quality factor (Q) of the antenna.

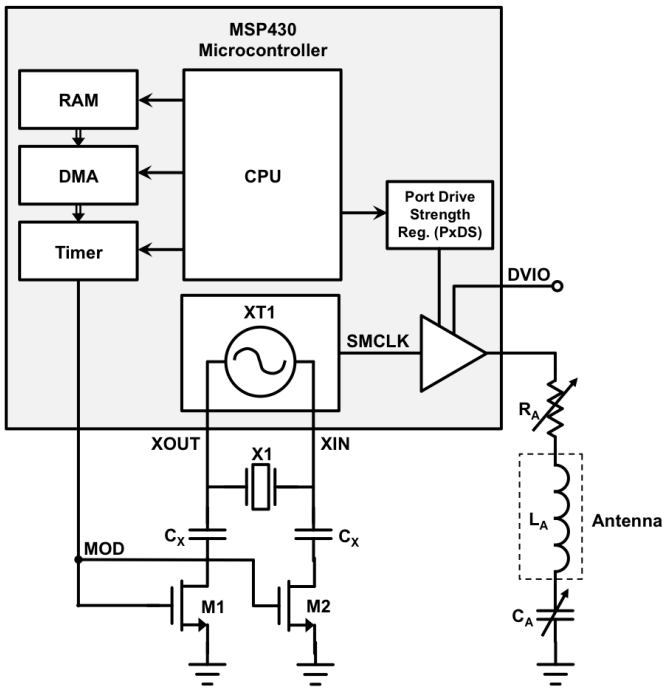

Figure 5.13: Block diagram of MSP430-based transmitter hardware.

5.3.3 MSP430-based Implementation

This section describes a low-power implementation of the radio transmitter using *only* the TI MSP430 microcontroller with minimal external hardware. This implementation dramatically reduces system complexity, component count, and cost. In addition, this design allows for more control of the transmitter's power output and modulation depth (which relates to bandwidth). This design is also much more robust than designs requiring additional components, and therefore can be easily deployed in the field with much less worry about damaging the circuitry. The simplicity, flexibility, and robustness of this design come at the cost of increased power consumption. Figure 5.13 shows the block diagram of the transmitter.

As shown in Figure 5.13, the internal digital clock oscillator (XT1) in the MSP430F5172 is used along with an external 27 MHz crystal resonator (X1) as the transmit oscillator. The

data is modulated by shifting the transmit oscillator frequency by switching load capacitors (C_X) into the oscillator's resonant tank. The capacitors are switched using very low capacitance n-channel MOSFETs (M1 and M2). In the current implementation, the C_X load capacitors are 47 pF, and the switching transistors, M1 and M2, are the BF1212 dual-gate n-channel MOSFETs. Using these components on our custom printed circuit board (PCB), the frequency shift is approximately 15 kHz. The modulation of the frequency is controlled using the MOD output of the microcontroller, which is connected to the gates of the switching transistors.

The modulation depth, or the amount of frequency shift during modulation, can be changed by using different size load capacitors C_X. In addition, a higher frequency deviation can be obtained by adding an inductor in series with the crystal.

The power consumption of the XT1 oscillator at 27 MHz is 920 μA at 3.0 V. This is the power consumption of the oscillator and SMCLK lines only, not the I/O buffer. This therefore represents the minimum transmit power consumption of this implementation.

Inside the MSP430, the XT1 transmit oscillator output is configured to drive the internal SMCLK clock line, and is routed inside the MSP430 to an individual I/O pin driver. This I/O buffer is used as the transmit amplifier which drives the antenna. The following section describes how the power delivered to the antenna can be varied.

The I/O buffer on the MSP430 which is designed to drive digital signals has a high gain push-pull output and therefore will produce a square wave (rail-to-rail) unless a resonant tank is present at the output. To resonate the output, a series capacitor (C_A) is used along with the loop antenna, which is mostly inductive (L_A). A variable capacitor is used so that it can be manually tuned to resonate the antenna at the desired center frequency.

It is often very desirable to adjust the power output delivered to the antenna in order to balance the power consumption of the transmitter with the signal-to-noise ratio seen at the receiver. Using the implementation of the transmitter described in this section, there are a number of ways in which this output power can be varied. The maximum current that can be delivered to the antenna is 20 mA. If more current is needed, an external amplifier will

be required.

The simplest way to vary the output power is to change the value of the resistor in series with the loop antenna (R_A). This will limit the current into the antenna, but will also reduce the quality factor (Q) of the antenna.

The output power can also be varied by changing the supply voltage of the I/O buffer (DVIO) which drives the antenna. The I/O buffer supply can be set independent of the MSP430s supply voltage if the antenna is connected to one of a few pins which are powered from the DVIO supply rather than DVCC. This is a feature only available on select MSP430 microcontrollers, including the MSP430F5172 used in this design. The DVIO supply can be between 1.8 and 5.5 V. The DVIO pin can be connected to an external supply voltage, a separate regulated voltage, or simply through a series resistor to DVCC. In the simplest implementation, a series potentiometer is used to control the DVIO supply voltage. This simple resistor works quite well since the current consumption of the buffer is relatively constant. This is also desirable over a regulator, because no current is consumed when the transmitter is off.

The third method of controlling the output power is to change the drive strength of the I/O buffer. On the MSP430F5172, the I/O buffers have a digitally controlled drive strength. When in the full drive strength mode the maximum current is 20 mA (at 5.0 V), while in the low drive strength mode the maximum current is 12 mA (at 5.0 V).

All three methods to vary the transmit power can be combined to give fine grain control of the output power and antenna Q. Since the power consumption of the transmit buffer can be varied between 0 and 100 mW (20 mA at 5.0 V), and the power consumption with a very high impedance load is 4.74 mW (1.58 mA at 3.0 V), then the power consumption of the whole node during transmission can vary between 4.74 mW and 100 mW.

5.3.4 Packet Modulation

The previous section describes the hardware required for producing the RF signal which is transmitted over the antenna. Each of these designs have a digital input to modulate the data

using binary FSK. This modulation signal is generated by the sensor node's microcontroller which also needs to consume power to produce this signal. In the original SNUPI paper, this was done by running the CPU continuously during transmission [44]. In that original paper, the power consumption during transmission was actually dominated by the CPU, and therefore reducing the power needed to control the modulation bit would result in significant power savings. In this subsection, I present several methods for reducing the power consumption of generating the modulation signal. One of these methods controls the modulation bit during transmission without running the CPU at all.

In this section, I will refer to specific timer registers present in the MSP430F5xx series. The block diagram for Timer_A can be found in the MSP430F5xx User's Guide SLAU208[6] and shows the structure of the timers. Note that there are 7 CCRx blocks on each timer, and the CCRx block shown in the block diagram is CCR6, but we are actually using CCR1, so I'll refer to names on CCR1 so the numbers on some signals and registers won't match the diagram exactly. For example, we are using OUT1, but it is labeled OUT6 in the diagram.

In order to modulate the packet data during transmission, the modulation bit must toggle at the correct data rate and according to the data to be transmitted. Therefore, some control logic is needed to control when the modulation bit toggles. The simplest, but most power hungry method is to run the CPU continuously. The simplest optimization of this is to put the CPU to sleep during each bit period, so that it is on only at the bit edges. Doing so requires the use of an on-chip timer, like Timer_A from the MSP430F5xx series. The flow chart of this minimally optimized algorithm is shown in Figure 5.14 (*left*), and this is the most common algorithm used in wireless sensor nodes today. The hardware timer is used to generate an interrupt at every bit edge so that the CPU can change the state of the modulation pin corresponding to the next bit value to send.

The first optimization to this basic, timer-based modulation algorithm is to use pulse widths rather than the bit stream. In this optimization, we are still using the timer to wake

---

[6] `www.ti.com/lit/pdf/SLAU208`, Figure 17-1 (page 460) of Rev. N (May 2014)

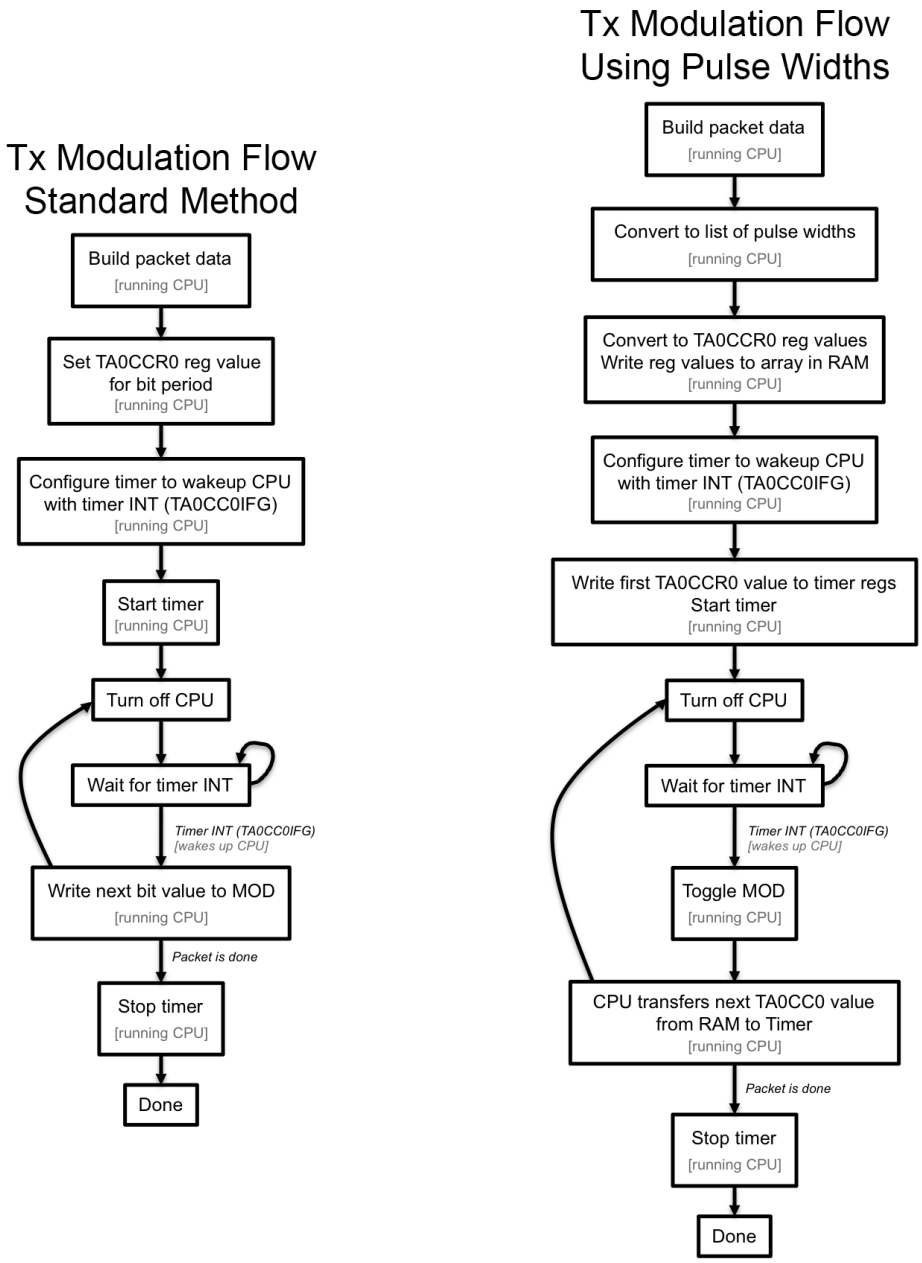

Figure 5.14: Flow diagrams for CPU controlled modulation techniques. The algorithm on the left uses the timer to wake up the CPU on every bit edge so that the CPU can change the state of the modulation pin. The algorithm on the right uses the timer to wake up the CPU only when the data value needs to change, thus eliminating wasted CPU activity when no action needs to be taken.

up the CPU, but instead of waking up every bit period to set the next bit value on the pin, we pre-encode the packet as a list of pulse widths and therefore only wake up the CPU when the state of the modulation pin needs to be changed. The flow chart of this algorithm is illustrated in Figure 5.14 (*right*). Although this saves power during transmission, more power needs to be consumed to pre-encode the packet data into pulse widths.

The next optimization is to free the CPU from the task of changing the modulation pin. Instead, we drive the modulation pin using the timer output OUT1. The timer is configured to set OUT1 high then TA0R == TA0CCR1, and set OUT1 low when TA0R == TA0CCR0. So, basically once all of the other timer registers are setup (*i.e.*, to make it count at the right rate and generate the correct outputs and interrupts), the result on OUT1 is determined by the values of two registers: TA0CCR0 and TA0CCR1. Each cycle of the timer, it counts from 0 to TA0CCR0 and then resets. During this period one high pulse and one low pulse is generated on OUT1 (*i.e.*, the MOD signal). Therefore by running many cycles of the timer and changing the values of the two TA0CCRx registers on each cycle we can output any arbitrary data. In order to transmit the data of the whole packet, these timer control registers will need to be updated each timer cycle until the whole packet has been transmitted. This can be done using CPU interrupts. In this case, the timer is configured and then the CPU is turned off. At the end of each cycle, the timer turns the CPU back on through an interrupt request so that the CPU can reconfigure the timer's control registers for the next cycle. This method of modulation allows the CPU to be off for part of the time that the packet is being transmitted; however, the CPU is still required every timer cycle to update the control registers. Figure 5.15 (*left*) shows the program flow for this algorithm.

Finally, the need to turn on the CPU can be eliminated if the timer's control registers are updated using a direct memory access (DMA) transfer directly from RAM. In this case, the CPU can remain off during the entire packet transmission. For this implementation, the CPU configures the timer as before, fills arrays in RAM with all of the timer control register values that will be needed during the transmission, and configures the DMA controller to transfer data from the arrays in RAM directly to the timer control registers. The DMA

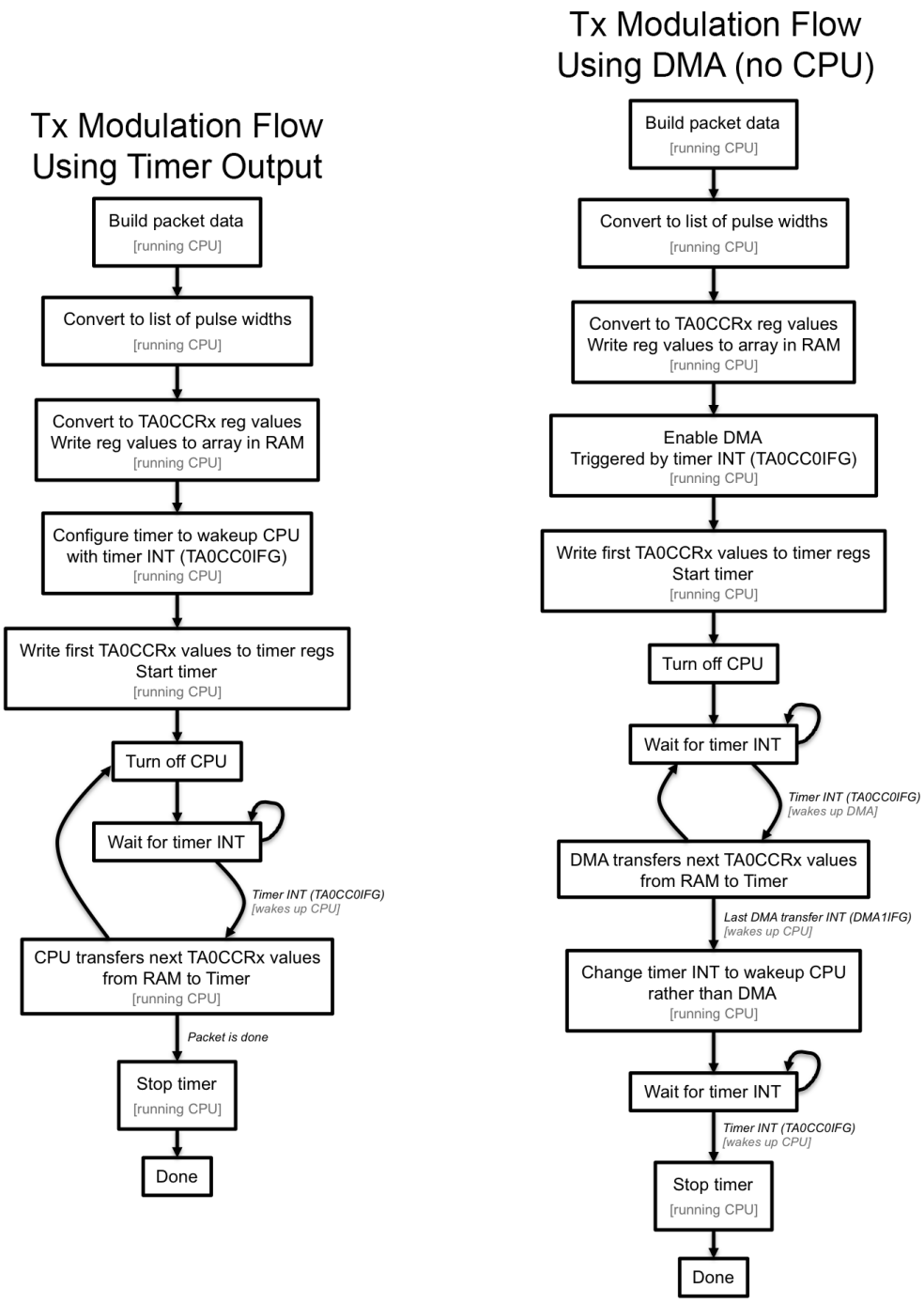

Figure 5.15: Flow diagrams for timer output controlled modulation techniques. The algorithm on the left uses the CPU to change the timer control registers every cycle of the timer. The algorithm on the right frees the CPU from this task by using the DMA to change the timer control registers every cycle of the timer so that the CPU can remain off during the entire packet transmission.

transfers are triggered each timer cycle by the timer overflow event, and the CPU is turned back on through an interrupt request once the last DMA transfer occurs. Once the CPU is turned back on, it stops the timer and turns off the transmitter since the packet transmission has been completed. Note that this method requires carefully choosing the clock frequencies for CPU, DMA, and TA0 in order to satisfy timing constraints. Figure 5.15 (*right*) shows the program flow for this algorithm.

Freeing up the CPU can have a major power savings as the CPU consumes much more power than the Timer or DMA peripherals. Alternatively, the CPU can be used to perform another task, which can also be considered a power savings since that task won't need to be done later and thus the whole system can be powered down sooner.

5.4 Antenna

The antenna is a critical part of the sensor node, as its job is to wirelessly couple the RF signals to the power lines as efficiently as possible. For most antenna designs, maximally efficient antennas are roughly ½ of a wavelength in size. Since SNUPI works best at the 13.56 and 27.12 MHz ISM bands, this corresponds to wavelengths of 22.12 and 11.06 m respectively. Clearly, it is not practical to make wireless sensor nodes with antennas that are tens of meters in size. As a result, we are forced to use electrically small, highly inefficient antennas. These inefficient antennas are sufficient however because the SNUPI node only needs to couple it's signal in the near-field to the nearest power line, which is at most a few meters away. In addition, the size of the power line infrastructure in the home is roughly the size of an efficient antenna at the SNUPI operating frequency, and therefore makes a fairly efficient receiving antenna. However, it is still very important to optimize the radiation efficiency of the transmit antenna as much as possible. This section describes methods for modeling, testing, and optimizing electrically small loop antennas for use in a SNUPI sensor network.

Experimentation has shown that RF signals can most efficiently be inductively coupled to the power lines. Therefore, loop antennas are ideal because they maximize the magnetic

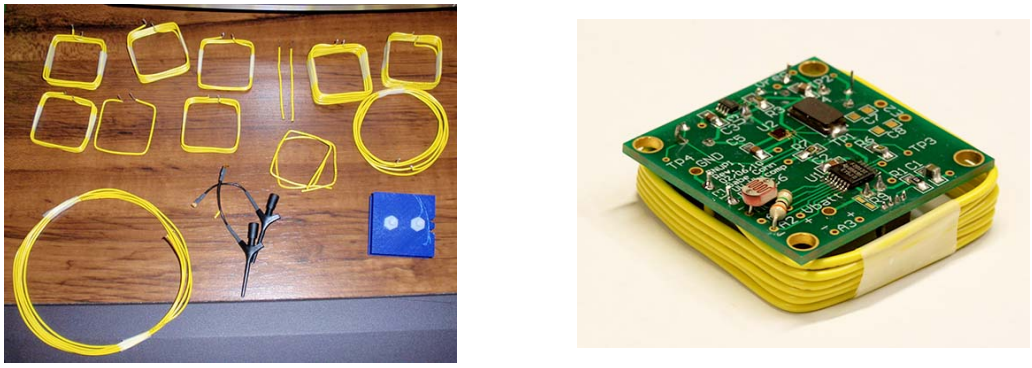

Figure 5.16: (*left*) Wire-wound loop antennas tested for original SNUPI node. (*right*) Original 2010 SNUPI node showing wire-wound loop antenna made from 6 turns of 22 AWG wire.

field radiated. For the original 2010 SNUPI node, a number of different wire-wound antennas were tested. Each antenna was attached to the SNUPI node which was programmed to emit a continuous tone at 27.0 MHz. The node's power consumption was measured along with the received amplitude of the carrier tone from a spectrum analyzer plugged into the power line. After testing over 20 different antennas, some shown in Figure 5.16 (*left*), the best was selected. The best antenna was made from 6 turns of 22 AWG wire wound around the 3.8 $\times$ 3.8 cm square perimeter for the 2010 SNUPI node, as shown in Figure 5.16 (*right*). Multiple turns were used to increase the impedance and improve the radiation efficiency by increasing the radiation resistance. These 6 turns of insulated wire also made the antenna about 1.5 cm in height. Heavier gauge wire is used to reduce the loss resistance of the antenna, which also improves the radiation efficiency [15].

Although the chosen antenna was the best of the 20 or more antennas tested, it was assumed to be far from ideal. In this section, I formalize the process of designing and selecting antennas for SNUPI sensor nodes. In Section 5.4.1, I list the set of antennas chosen for modeling and testing. Section 5.4.2 describes the methods for modeling the electrical properties and radiation efficiency of both wire-wound and PCB-etched antennas. Section 5.4.3 gives the modeling results for the selected antennas, and Sections 5.4.4 and 5.4.5 describe methods for measuring the antenna parameters and radiation efficiency of the

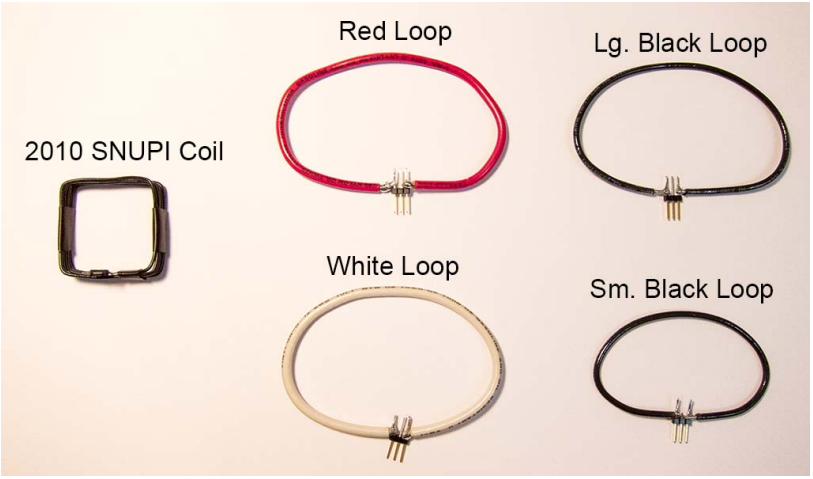

Figure 5.17: Wire-wound antennas evaluated. This includes the antenna design used on the original SNUPI node (square, black loop), and four single turn antennas with varied area, wire thickness, and solid-core vs. stranded wire.

antennas once they are built. Section 5.4.6 compares the results of the antenna models from their actual measurements as a method for validating the models. Finally, Section 5.4.7 describes a different approach of using the human body of a user as an antenna to couple SNUPI communications signals to the power lines for wearable sensor applications.

5.4.1 Antennas to Test

With the chosen wire antenna for the 2010 SNUPI node, 6 turns of wire were determined to be an optimal balance between resistance losses and improved radiation strength of increasing numbers of turns. However, another theory is that a single turn of very thick wire would make up for lost radiation strength with extremely low resistance losses. Such antennas were used widely in commercially available 27 MHz wireless computer mice. Therefore, in addition to the original 2010 SNUPI antenna, I will also model, measure, and evaluate 4 different antennas made from a single turn of very thick wire, as shown in Figure 5.17. The specifications of these antennas are listed in Table 5.8. These four single turn antennas varied in area, wire thickness, and the use of solid-core vs. stranded wire.

Although the 2010 SNUPI node used a wire-wound antenna, such antennas suffer from

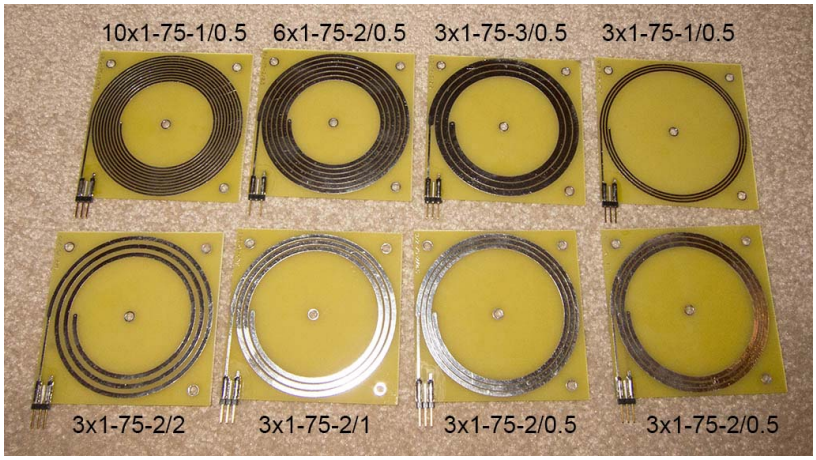

Figure 5.18: 75 mm round PCB antennas evaluated.

imperfections and parameter variation since they are hand wound. Antennas printed onto a PCB are thought to alleviate this concern and therefore allow for a higher Q, or quality factor. However, early testing with PCB antennas showed them to be inferior to wire-wound antennas due to high resistance losses. However, as shown in the chapter, efficient PCB antennas can be designed using the models described in Section 5.4.2. Initially, a number of round PCB antennas were made using different trace thickness, spacing, and number of turns, as shown in Figure 5.18. The specifications of these antennas are also shown in Table 5.8. All of these loops are round with a 75 mm outer diameter. They are labeled in the form NxS-D-T/s, where N is the number of turns on one side, S is the number of sides (1 or 2), D is the outer diameter in mm, T is the thickness of the wire trace of each turn in mm, and s is the spacing between turns in mm. For example, 3x1-75-2/0.5 is a 3 turn, 75 mm diameter loop with 2 mm thick traces that are spaced by 0.5 mm. A number of 2-sided 75 mm PCB antennas were made, but it was discovered that they had too much capacitance between the traces on the two sides of the board, resulting in a self-resonant frequency below the operating frequency. As a result, none of these 2-sided antennas are included in Table 5.8.

Based on the antenna models described in Section 5.4.2, 6 antennas were designed

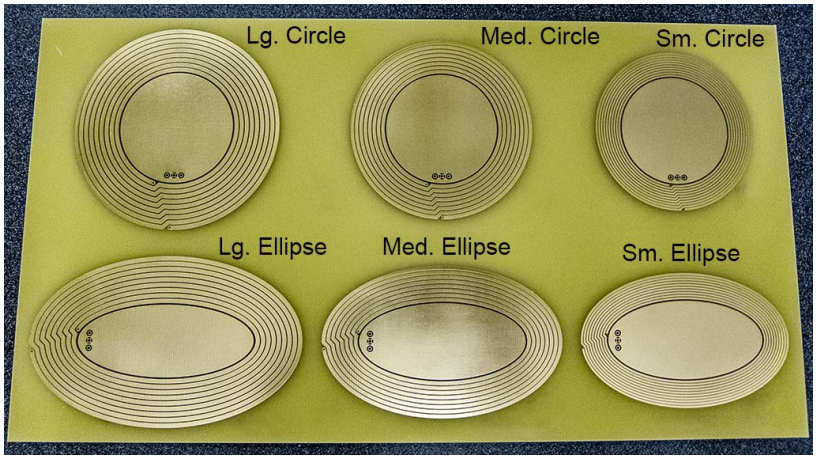

Figure 5.19: PCB antennas designed by model optimization.

based on the recommended optimal specifications from the model. These antennas are shown in Figure 5.19, and their specifications are listed in Table 5.8. Three of these antennas are circles, labeled "Lg. Circle," "Med. Circle," and "Sm. Circle" to indicate the large, medium, and small area circles. Similarly, the other three antennas have the same area as the large, medium, and small circles, but are ellipses, and are labeled appropriately. Note that these antennas also have an interior ground plane to represent the ground plane of the node's circuitry. This ground plane was not represented in the antenna model.

5.4.2 Antenna Modeling

The ultimate goal of the antenna modeling is to be able to computationally determine the dimensions of an optimal antenna given size, shape, and material constraints. To this end, we first need to build a model which is able to predict the antenna performance and impedance given the dimensions, shape, and materials of an antenna design. Once we have a model which accurately represents the physical antennas of the same specified dimensions as the model, we can use this model to find optimal antennas given the constraints of size, shape and materials. When designing a sensor node's antenna, the properties of greatest interest are the:

Antenna	Type	Shape	Area	Dimensions (mm)	N Turns	Wire Type	Wire Thickness	Wire Spacing
2010 SNUPI Coil	Wire Coil	Square	14 cm^2	$s = 38$	6	Solid Cu	#22 (0.644 mm)	–
Red Loop	Wire Loop	Oval	44 cm^2	$d_1 = 96, d_2 = 59$	1	Solid Cu	#10 (2.588 mm)	–
White Loop	Wire Loop	Oval	44 cm^2	$d_1 = 96, d_2 = 59$	1	Stranded Cu	#10 (2.588 mm)	–
Lg. Black Loop	Wire Loop	Oval	44 cm^2	$d_1 = 96, d_2 = 59$	1	Solid Cu	#14 (1.628 mm)	–
Sm. Black Loop	Wire Loop	Oval	28 cm^2	$d_1 = 77, d_2 = 47$	1	Solid Cu	#14 (1.628 mm)	–
3x1-75-2/0.5	PCB Spiral	Circle	44 cm^2	$d = 75$	3	1 oz Cu	2 mm	0.5 mm
3x1-75-2/1	PCB Spiral	Circle	44 cm^2	$d = 75$	3	1 oz Cu	2 mm	1 mm
3x1-75-2/2	PCB Spiral	Circle	44 cm^2	$d = 75$	3	1 oz Cu	2 mm	2 mm
3x1-75-1/0.5	PCB Spiral	Circle	44 cm^2	$d = 75$	3	1 oz Cu	1 mm	0.5 mm
3x1-75-3/0.5	PCB Spiral	Circle	44 cm^2	$d = 75$	3	1 oz Cu	3 mm	0.5 mm
6x1-75-2/0.5	PCB Spiral	Circle	44 cm^2	$d = 75$	6	1 oz Cu	2 mm	0.5 mm
10x1-75-1/0.5	PCB Spiral	Circle	44 cm^2	$d = 75$	10	1 oz Cu	1 mm	0.5 mm
Sm. Circle	PCB Coil	Circle	28 cm^2	$d = 60$	10	2 oz Cu	0.8 mm	0.15 mm
Med. Circle	PCB Coil	Circle	35 cm^2	$d = 66.8$	8	2 oz Cu	1.4 mm	0.2 mm
Lg. Circle	PCB Coil	Circle	44 cm^2	$d = 75$	9	2 oz Cu	1.7 mm	0.15 mm
Sm. Ellipse	PCB Coil	Ellipse	28 cm^2	$d_1 = 77, d_2 = 47$	10	2 oz Cu	0.75 mm	0.15 mm
Med. Ellipse	PCB Coil	Ellipse	35 cm^2	$d_1 = 86, d_2 = 52$	8	2 oz Cu	1.35 mm	0.15 mm
Lg. Ellipse	PCB Coil	Ellipse	44 cm^2	$d_1 = 96, d_2 = 59$	9	2 oz Cu	1.55 mm	0.15 mm

Table 5.8: Antenna specifications for all tested antennas.

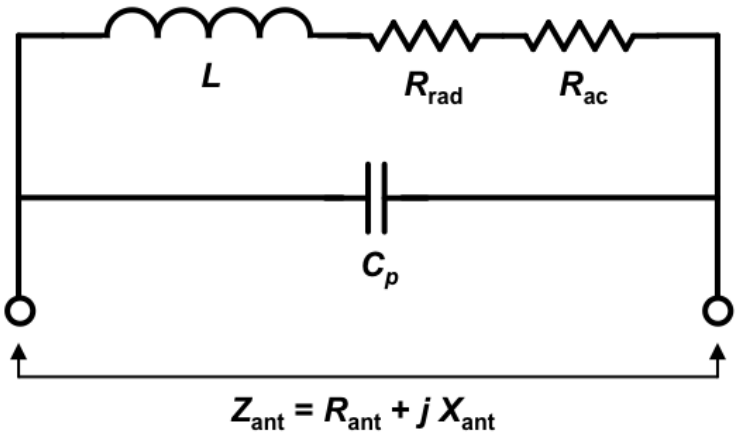

Figure 5.20: Circuit model of loop antenna.

1. radiation efficiency
2. self-resonance
3. tuning capacitance needed to resonate the antenna
4. Q, or quality factor

All of these properties of the antenna can be computed directly if four electrical parameters of the antenna are known. These parameters are shown in Figure 5.20. R_{ac} represents that AC loss resistance of the antenna, which models the power dissipated as heat. On the other hand, R_{rad} represents the radiation resistance of the antenna, which models the power radiated into the air. Since we are only concerned with loop antennas, the bulk of the reactance is inductive. L represents the inductance of the loop, while C_p represents the parasitic capacitance between the coils of the antenna.

Computing Antenna Selection Metrics

The next sub-section describes how to estimate each of the antenna parameters: R_{rad}, R_{ac}, L, and C_p. Here we will address computing the desired antenna properties given these parameters. First, it's useful to compute the impedance of the antenna using the antenna

model shown in Figure 5.20.

Impedance We will break the antenna impedance, Z_{ant} into it's resistive (R_{ant}) and reactive (X_{ant}) components such that $Z_{\text{ant}} = R_{\text{ant}} + jX_{\text{ant}}$. Before computing the impedance, let's first compute a few intermediates as a function of the operating frequency f in order to simplify the equations.

$$X_C = -\frac{1}{2\pi f C_p}$$

$$X_L = 2\pi f L$$

$$R_s = R_{\text{rad}} + R_{\text{ac}}$$

Now, the antenna impedance can by computed using the following equations:

$$R_{\text{ant}} = \begin{cases} R_s & C_p = 0 \\ \frac{R_s X_C^2}{R_s^2 + (X_C + X_L)^2} & C_p > 0 \end{cases}$$

$$X_{\text{ant}} = \begin{cases} X_L & C_p = 0 \\ \frac{R_s^2 X_C + X_C^2 X_L + X_C X_L^2}{R_s^2 + (X_C + X_L)^2} & C_p > 0 \end{cases}$$

$$Z_{\text{ant}} = R_{\text{ant}} + jX_{\text{ant}}$$

Radiation Efficiency With the impedance calculated, we can move on the compute the radiation efficiency, η_r which is the figure of merit to be maximized when designing an antenna. This represents the ratio of radiated power over the power input into the antenna,

and can be computed using:

$$\eta_r = \begin{cases} \frac{R_{\text{rad}}}{R_s} & C_p = 0 \\ \frac{R_{\text{rad}}\left(1+X_L^2+R_s^2\right)}{R_s^3+R_s(X_C+X_L)^2} & C_p > 0 \end{cases}$$

Self-Resonance Although the radiation efficiency is the most important property, as it determines the efficiency of the antenna, there are several other properties which impose practical constraints on the antenna design. The self-resonance, f_s is the frequency at which the antenna resonates without any external tuning capacitor. This is important because the antenna cannot be used above this frequency. The self-resonance can be computed using:

$$f_s = \frac{1}{2\pi\sqrt{LC_p}}$$

Tuning Capacitance Somewhat related to this is the size of the tuning capacitor C_t that is needed to tune the antenna into resonance. There are practical constraints on the range of values for this capacitor which can be physically realized. The tuning capacitance can be calculated using:

$$C_t = \frac{1}{2\pi f X_{\text{ant}}}$$

Quality Factor Finally, antennas are often compared by their Q, or quality factor, which represents how sharp or sensitive the tuning is. The Q can also be computed from the antenna impedance as:

$$Q = \frac{|X_{\text{ant}}|}{R_{\text{ant}}}$$

Computing Antenna Parameters

This sub-section describes the calculation of the antenna parameters: R_{rad}, R_{ac}, L, and C_p, which can be used to compute the metrics used for selection that are described in the previous sub-section. This model currently handles both wire-wound and PCB antennas in

both circular coils, and circular planar spirals. A coil is defined as loop with one or more turns that are all the same diameter, and thus stack on top of each other in the z-axis. On the other hand, a spiral is flat, and it's turns continually decrease in radius. This section is organized by the computation of each each parameter, but there are often different equations depending on the shape and substrate (*i.e.*, PCB or wire).

Generic Constants First, we'll define some generic constants that will be used throughout this sub-section:

$$
\begin{aligned}
c &= 2.99792458 \times 10^8 \quad \left[\tfrac{\mathrm{m}}{\mathrm{s}}\right] && \text{speed of light in free space} \\
\mu_0 &= 4\pi \times 10^{-7} \quad \left[\tfrac{\mathrm{H}}{\mathrm{m}}\right] && \text{magnetic permeability of free space} \\
\varepsilon_0 &= \tfrac{1}{\mu_0 c^2} \quad \left[\tfrac{\mathrm{F}}{\mathrm{m}}\right] && \text{permittivity of free space}
\end{aligned}
$$

Material Properties We also need to understand the material properties of the conductor that the antenna is made from. We'll assume that this is always Copper (Cu):

$$
\begin{aligned}
\chi_{\mathrm{Cu}} &= -0.98 \times 10^{-5} && && \text{magnetic susceptibility of Cu} \\
\mu_{\mathrm{Cu}} &= \mu_0(1+\chi_{\mathrm{Cu}}) && \left[\tfrac{\mathrm{H}}{\mathrm{m}}\right] && \text{magnetic permeability of Cu} \\
\sigma_{\mathrm{Cu}} &= 5.96 \times 10^7 && \left[\tfrac{\mathrm{S}}{\mathrm{m}}\right] && \text{conductivity of Cu}
\end{aligned}
$$

In addition, we need to convert the thickness of the wires and height of the PCB traces into meters. Wires in the U.S. are typically sized in AWG, and the height of the copper clad on a PCB is typically specified in ounces per square foot. The following equations can be used either to convert a wire size in AWG (G) to a radius in m (r_{wire}), or a PCB trace height in ounces per square foot (w_{Cu}) into a height in m (z_{wire}):

$$
r_{\mathrm{wire}}(G) = \frac{0.127 \times 10^{-3}}{2} \times 92^{(36-G)/39}
$$

$$z_{\text{wire}}(w_{\text{Cu}}) = w_{\text{Cu}} \times 34.057 \times 10^{-6}$$

Dimension Parameters For wire-wound coils, the dimensions are specified as:

r	[m]	radius of coil
G_{wire}	[AWG]	wire gauge
N		number of turns
s	[m]	spacing between turns in the z-axis, if $N > 1$

For wire-wound spirals, the dimensions are specified as:

r_{max}	[m]	outer radius of spiral
G_{wire}	[AWG]	wire gauge
N		number of turns
s	[m]	planar spacing between turns, if $N > 1$

For PCB spirals, the dimensions are specified as:

r_{max}	[m]	outer radius of spiral
w_{Cu}	$\left[\frac{\text{oz}}{\text{ft}^2}\right]$	copper clad thickness (height)
N		number of turns
T	[m]	trace thickness
s	[m]	spacing between traces, if $N > 1$

Radiation Resistance Generically, we can compute the radiation resistance of an air-core loop antenna as a function of its effective area (A_{eff}) and the wavelength (λ):

$$\lambda = \frac{c}{f}$$

$$R_{\text{rad}} = 320\pi^4 \left[\frac{A_{\text{eff}}}{\lambda^2}\right]^2$$

In the simple case of a coil, the effective area is simply the area of each turn (A_t), times the number of turns (N):

$$A_t = \pi r^2$$

$$A_{\text{eff}} = NA_t$$

In the case of a spiral, the effective area is not as straight forward. The area of an arc is defines below, where Ω is an arc segment of a circle, normalized such that $\Omega = 1$ is a complete circle.

$$A_{\text{arc}} = \int \pi \, r(\Omega)^2 \, \mathrm{d}\Omega$$

For a simple circular loop,

$$r(\Omega) = r \quad \forall \Omega$$

However, for a spiral we can define $r(\Omega)$ in terms of the pitch (p), or the center-to-center spacing between the turns:

$$p = T + s$$

$$r(\Omega) = r_{\text{max}} - p\Omega$$

Therefore, the effective area is:

$$A_{\text{eff}} = \int_0^N \pi \, r(\Omega)^2 \, \mathrm{d}\Omega = \int_0^N \pi (r_{\text{max}} - p\Omega)^2 \, \mathrm{d}\Omega$$

$$A_{\text{eff}} = N\pi\left(r_{\text{max}}^2 - r_{\text{max}}Np + \frac{1}{3}N^2p^2\right)$$

Thus, the radiation resistance of an air-core spiral is:

$$R_{\text{rad,spiral}} = 320\pi^4\left[\frac{N\pi\left(r_{\text{max}}^2 - r_{\text{max}}Np + \frac{1}{3}N^2p^2\right)}{\lambda^2}\right]^2$$

While the radiation resistance of an air-core coil is:

$$R_{\text{rad,coil}} = 320\pi^4\left[\frac{N\pi r^2}{\lambda^2}\right]^2$$

DC Resistance The resistance (R) of the wire, is simply a function of the length of the wire (l), its cross-sectional area (A_c), and its conductivity (σ):

$$R = \frac{l}{\sigma A_c}$$

Thus we need to compute the cross-sectional area (A_c) and length of the wire l. In the case of a wire-wound antenna, the wire is round, and thus the cross-sectional area is a function of the wire radius (r_{wire}):

$$A_{c,\text{wire}} = \pi r_{\text{wire}}^2$$

However, in the case of a PCB antenna, the conductors are rectangular, and thus the cross-sectional area is a function of the height (z_{wire}) and the width (T):

$$A_{c,\text{pcb}} = T z_{\text{wire}}$$

For a coil, the length of the wire (l) can easily be computed as N times the circumference of one turn:

$$l_{\text{coil}} = 2\pi rN$$

However, the length of the wire in a spiral antenna is more complicated. The arc length (l_{arc}) spanned by an angle θ is:

$$l_{\text{arc}} = r\theta$$

Thus generically, the wire length (l) as a function of the radius (r) is:

$$l = \int_0^{2\pi N} r(\theta)\,\mathrm{d}\theta$$

For a simple circular loop,

$$r(\theta) = r \quad \forall\,\theta$$

However, for a spiral we can define $r(\theta)$ as a function of the pitch (p):

$$r(\theta) = r_{\max} - \frac{p\theta}{2\pi}$$

Thus, the length of wire in a spiral is:

$$l_{\text{spiral}} = \int_0^{2\pi N} r(\theta)\,\mathrm{d}\theta = \int_0^{2\pi N} \left(r_{\max} - \frac{p\theta}{2\pi}\right)\,\mathrm{d}\theta$$

$$l_{\text{spiral}} = \left[r_{\max}\theta - \frac{p}{4\pi}\theta^2\right]_0^{2\pi N}$$

$$l_{\text{spiral}} = N\pi(2r_{\max} + Np)$$

Thus, the DC resistance of a spiral PCB antenna is:

$$R_{\text{dc,spiral,pcb}} = \frac{N\pi(2r_{\max} + Np)}{\sigma T z_{\text{wire}}}$$

While, the DC resistance of a wire-wound spiral is:

$$R_{\text{dc,spiral,wire}} = \frac{N(2r_{\max} + Np)}{\sigma r_{\text{wire}}^2}$$

And finally, the DC resistance of a wire-wound coil is:

$$R_{\text{dc,coil,wire}} = \frac{2rN}{\sigma r_{\text{wire}}^2}$$

AC Resistance The DC resistance of a wire, which can be computed using the above equations is only accurate for describing a wire's resistance at very low frequencies. At higher, RF frequencies there are two charge distribution effects which will increase the apparent resistance of the wire. The first is the skin effect, which states that charge flows primarily at the surface of the conductor at high frequencies. The second effect on wire coils is the proximity effect, in which charge is attracted and repelled by the charge flowing through the adjacent turns of the coil. This causes the charge density to be greatest between the coils, and much less dense elsewhere. In this paragraph, we will derive equations to handle both of these effects on the AC loss resistance of a loop antenna. The skin effect is well understood, and it is typically modeled by computing a skin depth, in which it is assumed that all of the charge flows. Although this is not truly accurate, it serves as a reasonable approximation. The skin depth (δ) is generally computed as a function of the operating frequency (f) and the material properties:

$$\delta = \frac{1}{\sqrt{\pi f \sigma \mu}}$$

We can therefore substitute the conductivity (σ) and magnetic permeability (μ) of copper to obtain an equation which is only a function of the operating frequency (f):

$$\delta_{\text{Cu}} = \frac{1}{\sqrt{\pi f \sigma_{\text{Cu}} \mu_{\text{Cu}}}}$$

The skin effect essentially limits current flow to the surface or skin of the conductor with a depth of δ. This will essentially reduce the cross-sectional area that we computed for the DC resistance by removing a hole in the middle. For round wires, the cross-sectional area

(A_c) becomes:

$$A_{c,\text{wire}} = \pi \left(r_{\text{wire}}^2 - (r_{\text{wire}} - \delta_{\text{Cu}})^2 \right)$$

For PCB traces, the cross-sectional area (A_c) becomes:

$$A_{c,\text{pcb}} = T z_{\text{wire}} - (z_{\text{wire}} - 2\delta)(T - 2\delta) = 2\delta(z_{\text{wire}} + T - 2\delta)$$

These expressions for the cross-sectional area can therefore be substituted into the equation for resistance to obtain the AC resistance when taking the skin effect into account:

$$R = \frac{l}{\sigma A_c}$$

Therefore, the AC resistance (considering only the skin effect) of a spiral PCB antenna is:

$$R_{\text{ac,skin,spiral,pcb}} = \frac{N\pi(2r_{\max} + Np)}{2\sigma\delta(z_{\text{wire}} + T - 2\delta)}$$

While, the AC resistance (considering only the skin effect) of a wire-wound spiral is:

$$R_{\text{ac,skin,spiral,wire}} = \frac{N(2r_{\max} + Np)}{\pi\sigma \left(r_{\text{wire}}^2 - (r_{\text{wire}} - \delta_{\text{Cu}})^2 \right)}$$

And finally, the AC resistance (considering only the skin effect) of a wire-wound coil is:

$$R_{\text{ac,skin,coil,wire}} = \frac{2rN}{\pi\sigma \left(r_{\text{wire}}^2 - (r_{\text{wire}} - \delta_{\text{Cu}})^2 \right)}$$

The above equations are accurate for spirals and coils with less than 1.5 turns (*i.e.*, $N < 1.5$); however, when there are multiple turns, they interact with each other via the proximity effect. The proximity effect is much difficult to model, but the most recent and accurate model for rectangular conductors (*i.e.*, PCB traces) is replicated below from [68]: First, we need to define the packing factor (η) which represents the fraction of the pitch (*i.e.*, sum of

the wire thickness and spacing) covered by the wire:

$$\eta = \frac{T}{T+s} \quad \text{for } 0 < \eta \leq 1$$

Next, we give a new definition of the skin depth as a function of the packing factor:

$$\delta(\eta) = \frac{1}{\sqrt{\pi f \sigma \mu \eta}}$$

We now use an approximation in which we model the rectangular wires with round wires with a diameter d such that the round wires have the same cross-sectional area (A_c) as the rectangular ones:

$$A_c = T * z_{\text{wire}}$$

$$d = 2\sqrt{\frac{A_c}{\pi}} = \frac{2}{\pi}\sqrt{\pi T z_{\text{wire}}}$$

We now define $\xi(\eta)$ in terms of the equivalent wire diameter (d) and the skin depth ($\delta(\eta)$):

$$\xi(\eta) = \frac{\sqrt{\pi}}{2}\frac{d}{\delta(\eta)}$$

Now, we can compute the AC resistance (R_{ac}) including both the skin effect and the proximity effect as a function of the DC resistance (R_{dc}) and $\xi(\eta)$:

$$R_{\text{ac}} = R_{\text{dc}}\left[\frac{\sinh\xi(\eta)+\sin\xi(\eta)}{\cosh\xi(\eta)-\cos\xi(\eta)} + \eta^2\frac{\sinh\xi(\eta)-\sin\xi(\eta)}{\cosh\xi(\eta)+\cos\xi(\eta)}\right]$$

This equation is designed to be used for rectangular conductors (*i.e.*, PCB traces), and another equation is to be used for round conductors [68]. However, I never implemented any model of the proximity effect for round conductors, and therefore simply rely on the AC resistance model which uses only the skin effect.

Inductance The inductance of a circular coil is simply defined in terms of the area of the loop (A_t), the number of turns (N), the height or thickness of the coil (h), and the magnetic permeability of the core (μ). The area the coil is simply:

$$A_t = \pi r^2$$

The height of the coil can be computed based on the pitch p (*i.e.*, the center-to-center distance between turns), the wire thickness (T), and the number of turns (N):

$$p = T + s$$

$$h = p(N-1) + w$$

Assuming an air-core, $\mu = \mu_0$, and the inductance of the coil (L_{coil}) can be computed as:

$$L_{\text{coil}} = \frac{\mu_0 A N^2}{h}$$

In a spiral, each turn has a different area and therefore a different partial inductance. Therefore, an approximation is used to compute the inductance of a planar spiral [145]. First, we compute the maximum and minimum turn diameters, d_{max} and d_{min}, respectively from the maximum and minimum turn radii, r_{max} and r_{min}:

$$d_{\text{max}} = 2r_{\text{max}} - \frac{p}{2} + T$$

$$r_{\text{min}} = r_{\text{max}} - Np$$

$$d_{\text{min}} = 2r_{\text{min}} + \frac{p}{2} - T$$

Now we compute the average diameter (d_{avg}), and the fill ratio (F) defined below:

$$d_{\text{avg}} = \frac{d_{\text{max}} + d_{\text{min}}}{2}$$

$$F = \frac{d_{\max} - d_{\min}}{d_{\max} + d_{\min}}$$

Finally, we can compute the inductance of an air-core planar spiral (L_{spiral}) as follows [145]:

$$L_{\text{spiral}} = \frac{\mu_0 N^2 d_{\text{avg}}}{1 + F}$$

Parasitic Capacitance Although the coil is primarily inductive, there is also a parasitic capacitance between the turns of the antenna. This effect is only present if there are 2 or more turns (*i.e.*, $N > 2$). To compute the parasitic capacitance, we need to compute the capacitance between each turn, and then take the series combination of all these capacitances [81]. The capacitance of each turn (C_t) can be estimated using the standard parallel plate capacitor theory in which the capacitance (C) is a function of the area of the plates (A) and the distance between the plates (d):

$$C = \frac{\varepsilon_0 A}{d}$$

In the case of a PCB antenna, the capacitance of a single turn is thus:

$$C_t = \frac{\varepsilon_0 z_{\text{wire}} l}{s}$$

Where z_{wire} is the height of the trace, s is the spacing between traces, and l is the path length of the trace. As derived earlier, the path length (l) of a spiral is:

$$l_{\text{spiral}} = \int_{2\pi n_1}^{2\pi n_2} r(\theta)\, \mathrm{d}\theta = \int_{2\pi n_1}^{2\pi n_2} \left(r_{\max} - \frac{p\theta}{2\pi} \right) \mathrm{d}\theta$$

$$l_{\text{spiral}} = \left[r_{\max}\theta - \frac{p}{4\pi}\theta^2 \right]_{2\pi n_1}^{2\pi n_2}$$

$$l_{\text{spiral}} = 2\pi r_{\max}(n_2 - n_1) + \pi p \left(n_1^2 - n_2^2 \right)$$

Where n_1 is the start position of the path, and n_2 is the end position of the path, defined such that $n = 0$ is the outermost point on the path (*i.e.*, the point $r_{\max}$ away from the center). Therefore, the capacitance of turn i is:

$$C_{t,i} = \frac{\varepsilon_0 z_{\text{wire}} \left(2\pi r_{\max}(n_i - n_{i-1}) + \pi p \left(n_{i-1}^2 - n_i^2\right)\right)}{s} \quad \forall\, i \in 1,2,\ldots,N-1$$

The total parasitic capacitance C_p is simply the series combination of each of the turn capacitances, $C_{t,i}$:

$$C_p = \left[\sum_{i=1}^{N-1} \frac{1}{C_{t,i}}\right]^{-1}$$

Finding Optimal Antenna Design

Using the antenna model described above, we can estimate all of the parameters of an antenna design without building the antenna. This is most useful for finding the optimal antenna design given a set of design constraints. To do this, I have developed an optimization program to return an optimal one-sided PCB antenna design given the following design constraints:

1. maximum antenna diameter
2. minimum diameter of component area (*i.e.*, the hole in the center of the antenna where the rest of the circuity goes)
3. PCB copper clad thickness (minimum and maximum)
4. source output impedance
5. operating frequency
6. tuning capacitance (minimum capacitance)
7. quality factor (minimum or maximum)
8. option to constrain to integer number of turns
9. trace thickness (minimum, maximum, and step size)
10. trace spacing (minimum, maximum, and step size)

Since the radiation resistance is proportional to the square of the effective area of the antenna, larger antennas perform significantly better. Therefore, the maximum antenna area is typically a design constraint determined by the size of the sensor node and the application. It is common to place the circuitry for the sensor node inside the antenna, so I also specify a keep-out area inside the antenna loop.

The next option is the thickness of the PCB copper clad, which is simply a decision related to cost. Thicker copper is typically more expensive, but also offers improved radiation efficiency by reducing the AC loss resistance. Due to the proximity effect, the charge is mostly at the edges between the coils of the antenna. As a result of this, the thickness of the copper clad has a very significant effect on the radiation efficiency. Roughly speaking, doubling the copper thickness will double the radiation efficiency!

The next four design constraints: source output impedance, operating frequency, minimum tuning capacitance, and quality factor all relate to the physical hardware around the antenna. The operating frequency is used in many of the antenna parameter calculations, but also to set a minimum constraint on the self-resonance frequency. The model assumes that a series capacitor will be used to tune the antenna into resonance. There are practical constraints on the size of the of the tuning capacitor, mostly due to part availability. This constraint allows the model to only produce antennas which are practical to tune. For certain applications, a designer may want to put constraints on the Q, or quality factor of the antenna. For example, it could be desirable to set a maximum so that the antenna is not too difficult to tune or so that it does not easily become detuned by nearby objects. Finally, the source output impedance is specified if it cannot be changed or an impedance matching network cannot be used, and you want to make sure that the total efficiency, including impedance matching losses are maximized.

There is an option to constrain the antenna to have an integer number of turns. This is a practical constraint, as in many designs the two feed points need to come from a single location on the board. Using an integer number of turns ensures that such an antenna will be designed. The last two design parameters are regarding the bounds on the trace thickness

and spacing. For manufacturabilty, there are minimums on the size of the PCB traces as well as the spacing between them. In addition, the user needs to be define the step size with which the optimization engine will search for an ideal antenna design.

The way the optimization code works is that it builds a list of possible antenna patterns that meet the specified design constrains, and then for each antenna pattern it computes the radiation efficiency. The antenna pattern with the greatest radiation efficiency that meets all of the design constraints is considered the optimal antenna. Because this relies on a brute-force optimization (*i.e.*, it tests every possible antenna pattern), the choice of the trace thickness and spacing step sizes are critical to runtime. It is best to do a course-grain search of the space, and then do fine-grained searches near a few of the local optima. However, through my testing, I did not see much improvement by doing more fine-grained searches. This is because the optimization functions are all quite smooth.

Note that the circular and elliptical antennas shown in Figure 5.19 were designed by running this optimization. The current antenna model only works for circular antennas, and as a result the elliptical antennas are simply the same area, number of turns, trace width, and trace spacing as the circular antenna. In Section 5.4.6 these antennas will be evaluated to see how much of a difference the shape makes.

5.4.3 Antenna Model Estimates

Using the antenna model described in Section 5.4.2, the antenna parameters for each of the antennas in Table 5.8 have been estimated. Table 5.9 shows the results of these estimates. Although these antennas can be modeled at any frequency, and estimates were made at all ISM bands of interest for SNUPI networks (*i.e.*, 13.56, 27.12, and 40.68 MHz). However, only the estimates at 13.56 MHz are shown in the table. In Section 5.4.6, a comparison is made between the estimated values from the model shown in Table 5.9, and the measured values shown in Table 5.10 and Table 5.11.

Note that the original 2010 SNUPI coil could not be modeled because it is a multi-turn wire-wound antenna with a square shape, which the current model does not support. Using

Antenna	R_{ant}	X_{ant}	Tuning Cap	Q	Self-Resonance	Rad. Efficiency
2010 SNUPI Coil	–	–	–	–	–	–
Red Loop†	0.028 Ω	7.4 Ω	1595 pF	266	–	−40.4 dB
White Loop$^{\dagger\diamond}$	0.028 Ω	7.4 Ω	1595 pF	266	–	−40.4 dB
Lg. Black Loop†	0.442 Ω	7.6 Ω	1544 pF	172	–	−42.4 dB
Sm. Black Loop†	0.035 Ω	6.0 Ω	1957 pF	170	–	−45.3 dB
3x1−75−2/0.5	0.254 Ω	57.3 Ω	205 pF	225	527 MHz	−42.4 dB
3x1−75−2/1	0.220 Ω	55.1 Ω	213 pF	251	760 MHz	−42.1 dB
3x1−75−2/2	0.183 Ω	51.0 Ω	230 pF	279	1122 MHz	−41.9 dB
3x1−75−1/0.5	0.472 Ω	63.5 Ω	185 pF	134	495 MHz	−44.3 dB
3x1−75−3/0.5	0.170 Ω	51.4 Ω	228 pF	302	562 MHz	−41.6 dB
6x1−75−2/0.5	0.453 Ω	180.8 Ω	65 pF	399	297 MHz	−40.8 dB
10x1−75−1/0.5	1.352 Ω	512.7 Ω	23 pF	379	175 MHz	−41.0 dB
Sm. Circle‡	0.879 Ω	462.4 Ω	25 pF	526	80 MHz	−42.0 dB
Med. Circle‡	0.430 Ω	297.0 Ω	40 pF	691	110 MHz	−39.8 dB
Lg. Circle‡	0.462 Ω	395.1 Ω	30 pF	855	78 MHz	−37.6 dB
Sm. Ellipse†‡*	0.940 Ω	472.2 Ω	25 pF	502	80 MHz	−42.1 dB
Med. Ellipse†‡*	0.469 Ω	307.8 Ω	38 pF	656	94 MHz	−39.9 dB
Lg. Ellipse†‡*	0.517 Ω	414.9 Ω	28 pF	803	77 MHz	−37.7 dB

Table 5.9: Antenna parameter estimates using the model for all tested antennas at 13.56 MHz.
†Circular antenna of equal area modeled.
‡Spiral modeled, however antenna is actually concentric circles/ellipses with 45° jogs.
*Actual antenna has non-uniform spacing between turns.
$^{\diamond}$Actual antenna has stranded wire (not modeled).

the model, the single turn loops do not have any parasitic capacitance, and therefore are not self-resonant. In addition, the current model only supports circular antennas. Since several of the antennas are ovals or ellipses (*i.e.*, the single-turn wire loops, and the PCB ellipses), these were modeled using a circle with equal area. Also note that the small, medium, and large PCB circles and ellipses were modeled as continuous spirals; however, the antenna is really made up of concentric circles or ellipses which are joined by 45° jogs, as shown in Figure 5.19. Since the use of concentric ellipses do not produce uniform spacing between the turns, the modeled capacitance is likely inaccurate. Finally, the white single-turn loop antenna is made from stranded wire; however, the model assumes solid-core wire.

5.4.4 Antenna Parameter Measurement

Many of the antenna parameters cannot be directly measured once the antenna is made. However, the antenna's impedance can be measured as can it's self-resonance frequency. From the impedance, many of the antenna selection properties can be computed, including the tuning capacitance, and the Q. Table 5.10 shows the measured impedance, and self-resonance as well as the tuning capacitance and Q computed from the impedance for each antenna in Table 5.8. All measurements were made using the S_{11} values from a calibrated Vector Network Analyzer, and care was taken to keep the antenna away from conductive materials, so as not to change the impedance. Although measurements were made at all ISM bands of interest for SNUPI networks (*i.e.*, 13.56, 27.12, and 40.68 MHz), only the measurements at 13.56 MHz are shown in the table. In Section 5.4.6, a comparison is made between measured values shown in Table 5.10, and values values from the model as shown in Table 5.9.

Note that the self-resonant frequency was not reported if greater than 50 MHz. For operation in any of the ISM bands of interest for SNUPI networks (*i.e.*, 13.56, 27.12, and 40.68 MHz), the exact value of the self-resonant frequency is not important if it is greater than 50 MHz.

Antenna	R_{ant}	X_{ant}	Tuning Cap	Q	Self-Resonance
2010 SNUPI Coil	1.10 Ω	217 Ω	54 pF	197	$>$ 50 MHz
Red Loop	0.13 Ω	15 Ω	767 pF	122	$>$ 50 MHz
White Loop	0.10 Ω	14 Ω	832 pF	138	$>$ 50 MHz
Lg. Black Loop	0.12 Ω	17 Ω	699 pF	135	$>$ 50 MHz
Sm. Black Loop	0.11 Ω	13 Ω	939 pF	114	$>$ 50 MHz
3x1-75-2/0.5	0.59 Ω	106 Ω	111 pF	180	$>$ 50 MHz
3x1-75-2/1	0.56 Ω	100 Ω	117 pF	179	$>$ 50 MHz
3x1-75-2/2	0.50 Ω	87 Ω	135 pF	174	$>$ 50 MHz
3x1-75-1/0.5	1.02 Ω	136 Ω	86 pF	133	$>$ 50 MHz
3x1-75-3/0.5	0.48 Ω	88 Ω	134 pF	183	$>$ 50 MHz
6x1-75-2/0.5	1.70 Ω	295 Ω	40 pF	174	38 MHz
10x1-75-1/0.5	10.00 Ω	1120 Ω	10 pF	112	24 MHz
Sm. Circle	11.50 Ω	610 Ω	19 pF	53	32 MHz
Med. Circle	6.00 Ω	362 Ω	32 pF	60	39 MHz
Lg. Circle	8.60 Ω	504 Ω	23 pF	59	33 MHz
Sm. Ellipse	15.00 Ω	655 Ω	18 pF	44	30 MHz
Med. Ellipse	7.50 Ω	384 Ω	31 pF	51	37 MHz
Lg. Ellipse	10.60 Ω	542 Ω	22 pF	51	31 MHz

Table 5.10: Antenna parameter measurements for all tested antennas at 13.56 MHz.

5.4.5 Radiation Efficiency Measurement

In order to experimentally measure the radiation efficiency, both the transmitted power and the input power into the antenna need to be carefully measured. However, both these measurements can be quite difficult to make in practice.

The radiated power was estimated using the same radiated field strength measurement described in Section 3.3.1. The antenna was connected to a transmitter which sent a continuous 13.56 MHz tone. A calibrated ETS-Lindgren Model 6502 active shielded loop antenna was used to receive the field generated by the antenna under test. The antenna under test was placed 3 m away on the same axis as the receiving antenna, as shown in Figure 5.21. The antenna under test was rotated 360°, and the maximum received power was recorded using a calibrated spectrum analyzer. Using the calibration constants described in Section 3.3.1, the maximum effective isotropic radiated power (EIRP) of the antenna was calculated. These measurements are shown in Table 5.11. Note that the EIRP reported is computed based on the maximum field strength produced by main lobe the antenna's field pattern. The true radiated power will be less, but the exact value is not known unless the gain of the antenna is known. Therefore, the efficiency reported in Table 5.11 is an overestimate, as the true radiated power is less.

In order to compute the radiation efficiency, the power input into the antenna needs to be measured as well as the radiated power. Instead of directly measuring the power input into the antenna, the entire system power is measured using a Keithley 236 source measure unit (SMU) as the power supply to the transmitter. The SMU measures the supply voltage in addition to the supply current in order to determine the power consumption of the circuitry. The measured supply power is listed in Table 5.11 for each antenna as well as the computed system efficiency. The system efficiency is computed by taking the ratio of the supply power and the EIRP. The system efficiency will be less than the actual radiation efficiency, because the system efficiency includes losses in the transmitter and the antenna matching network. Since no impedance matching was used, these unknown losses will be different among different antennas. However, care was taken to ensure that each antenna

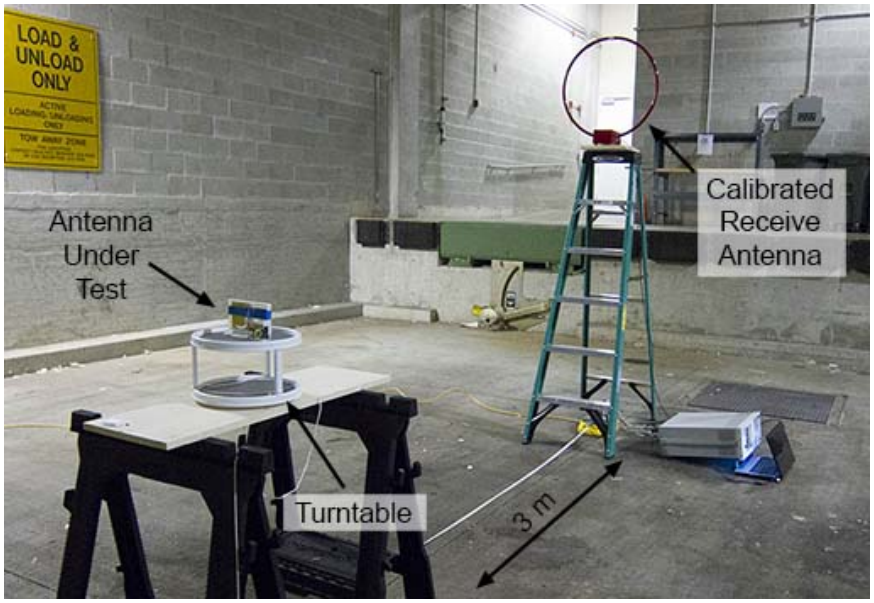

Figure 5.21: Picture of antenna radiated power measurement.

was tuned to resonance using on-board trim-capacitors.

Although measurements were made at all ISM bands of interest for SNUPI networks (*i.e.*, 13.56, 27.12, and 40.68 MHz), only the measurements at 13.56 MHz are shown in Table 5.11. In Section 5.4.6, a comparison is made between measured values shown in Table 5.11, and values values from the model as shown in Table 5.9.

5.4.6 Antenna Model Evaluation

Table 5.9 gives the estimated values from the model for all of the antennas tested, and Table 5.10 and Table 5.11 give the experimentally measured values from those same antennas. This section compares these results and draws some conclusions about antenna design and the validity of the model. Table 5.12 does a comparison of the estimated parameters from the model and the experimental results. For each parameter of interest, the ratio of model (M) vs. experimental (E) is shown in the table, and represented as either E/M or M/E. Note that some columns show the ratio of the model over the measurements (M/E), while other columns show the inverse (E/M).

Antenna	Supply Power	EIRP	Efficiency
2010 SNUPI Coil	22.21 dBm	−32.80 dBm	−55.02 dB
Red Loop	22.50 dBm	−38.74 dBm	−61.24 dB
White Loop	22.55 dBm	−39.37 dBm	−61.92 dB
Lg. Black Loop	22.43 dBm	−39.29 dBm	−61.72 dB
Sm. Black Loop	22.60 dBm	−43.27 dBm	−65.87 dB
3x1-75-2/0.5	22.23 dBm	−31.69 dBm	−53.92 dB
3x1-75-2/1	22.23 dBm	−31.56 dBm	−53.78 dB
3x1-75-2/2	22.21 dBm	−32.23 dBm	−54.45 dB
3x1-75-1/0.5	22.23 dBm	−30.36 dBm	−52.59 dB
3x1-75-3/0.5	22.24 dBm	−32.41 dBm	−54.65 dB
6x1-75-2/0.5	22.21 dBm	−27.10 dBm	−49.31 dB
10x1-75-1/0.5	21.54 dBm	−22.74 dBm	−44.28 dB
Sm. Circle	21.78 dBm	−29.84 dBm	−51.61 dB
Med. Circle	22.11 dBm	−29.75 dBm	−51.85 dB
Lg. Circle	22.01 dBm	−26.88 dBm	−48.90 dB
Sm. Ellipse	21.70 dBm	−29.60 dBm	−51.30 dB
Med. Ellipse	22.10 dBm	−29.42 dBm	−51.51 dB
Lg. Ellipse	21.92 dBm	−26.74 dBm	−48.65 dB

Table 5.11: Antenna efficiency measurements for all tested antennas at 13.56 MHz. The efficiency reported is not the antenna's radiation efficiency, but the entire system efficiency, including the efficiency of the transmitter and the antenna matching network. Also, note that the efficiency is an overestimate, as the true radiated power is less than the EIRP by a factor of the antenna's (unknown) gain.

Antenna	R_{ant} (E/M)	X_{ant} (E/M)	Tuning Cap (M/E)	Q (M/E)	Self-Resonance (M/E)	Efficiency (M/E)
2010 SNUPI Coil	–	–	–	–	–	–
Red Loop[†]	4.5	2.1	2.1	2.2	–	20.9 dB
White Loop[†◇]	3.7	1.9	1.9	1.9	–	21.6 dB
Lg. Black Loop[†]	0.3	2.2	2.2	1.3	–	19.3 dB
Sm. Black Loop[†]	3.1	2.1	2.1	1.5	–	20.6 dB
3x1-75-2/0.5	2.3	1.9	1.9	1.3	–	11.5 dB
3x1-75-2/1	2.5	1.8	1.8	1.4	–	11.7 dB
3x1-75-2/2	2.7	1.7	1.7	1.6	–	12.5 dB
3x1-75-1/0.5	2.2	2.1	2.1	1.0	–	8.3 dB
3x1-75-3/0.5	2.8	1.7	1.7	1.7	–	13.1 dB
6x1-75-2/0.5	3.8	1.6	1.6	2.3	7.8	8.5 dB
10x1-75-1/0.5	7.4	2.2	2.2	3.4	7.4	3.3 dB
Sm. Circle[‡]	13.1	1.3	1.3	9.9	2.5	9.6 dB
Med. Circle[‡]	14.0	1.2	1.2	11.5	2.8	12.0 dB
Lg. Circle[‡]	18.6	1.3	1.3	14.6	2.4	11.3 dB
Sm. Ellipse[†‡⋆]	16.0	1.4	1.4	11.5	2.7	9.2 dB
Med. Ellipse[†‡⋆]	16.0	1.2	1.2	12.8	2.5	11.6 dB
Lg. Ellipse[†‡⋆]	20.5	1.3	1.3	15.7	2.5	10.9 dB
Averages:	7.8	1.7	1.7	5.6	3.8	12.7 dB

Table 5.12: Comparisons between antenna model and experimental measurements for all tested antennas at 13.56 MHz. Each column represents a ratio of model (M) vs. experimental (E) values, represented as either E/M or M/E.

[†]Circular antenna of equal area modeled.

[‡]Spiral modeled, however antenna is actually concentric circles/ellipses with 45° jogs.

[⋆]Actual antenna has non-uniform spacing between turns.

[◇]Actual antenna has stranded wire (not modeled).

The first trend to notice is that the model consistently underestimates the impedance (*i.e.*, both the resistance and the reactance). The reactance is typically underestimated by a factor of about 2 ($\sigma = 1.7$), which results in an overestimate of the tuning capacitance by the same factor. This inaccuracy is manageable as it is quite consistent for almost all of the antennas tested. The resistance of the antennas are however much less consistent in their underestimate, resulting in an overestimate of the Q. This is a bit more troublesome, but the model still seems useful for many practical applications.

Although the model efficiency is overestimated by about 12 dB on average, this is expected. Remember that this is not a fair comparison, as the model estimates the efficiency of only the antenna's radiation, while the measured values also represent the inefficiency in the transmitter circuit and the antenna impedance mis-match. The difference in the efficiency does seem fairly consistent within antennas of similar types, and therefore is useful for comparing similar designs; although perhaps less useful for determining which type is best. This still makes the model extremely powerful.

The main strength of the model is not to determine the best type of antenna to use, but once an antenna type has been chosen, to design the best possible antenna given the constraints of the system. For this, it has performed quite well, and tends to design a near optimal antenna even if the estimated parameters are not 100% accurate.

It should also be noted that using the model to estimate the parameters of non-circular antennas using circles of the equivalent area seems to work equally well. This is because the shape of the antenna does not affect the efficiency much. Through experimentation, we have seen < 0.5 dB difference in the efficiency between circular and elliptical antennas with the same area.

The model can also be used to discover trends about how different aspects of the antenna design effect the radiation efficiency. As discussed earlier, using thicker traces (*i.e.*, heavier weight copper clad) has a nearly linear impact on the efficiency due to reduced R_{ac}. In addition, the area and number of turns have the greatest impact on the efficiency, so both should be increased to get more efficient antennas. However, increasing the number of

turns also increases the resistive losses, so a balance must be struck between many turns with small traces and few turns with large traces. The model is therefore very useful for finding this compromise. It should also be noted that the turns of the antenna should be placed as close to each other as possible. However, with many turns spaced very close together, the parasitic capacitance can get too large, and cause the self-resonance frequency to be too low.

5.4.7 Human Body Antennas

A persistent problem in most wearable wireless sensor systems is that the human body itself significantly attenuates the RF energy needed for communication. Wireless sensor networks typically operate in the globally available 2.4 GHz ISM band, as well as the regional 433 and 915 MHz bands. These spectra are attractive because they are unlicensed, have large bandwidths, and are widely available. However, the human body has been shown to cause significant attenuation of radio signals above 400 MHz [186]. To deal with this high path loss, I propose using a low operating frequency in which the body does not attenuate the RF energy as much. In fact, I hypothesize that at lower frequencies we can actually use the body as the transmitting antenna [73]. There is a large body of work which uses the body as a communication channel to communicate data across different parts of the body [46,85,254]. Since SNUPI also operates in the HF band, it may be possible to make a wearable implementation of SNUPI which uses the body as the transmitting antenna rather than simply using a loop antenna.

Body Antenna

We have designed an antenna similar to the loop antenna used in the original 2010 SNUPI node, but that is flat and can therefore be placed against the skin (*e.g.*, on the back of a watch). This antenna, shown in Figure 5.22, is made from 9 turns of 22 AWG wire, and has an inductance of 4.1 μH. The loop inductance can be tuned out at 27 MHz using an 8.4 pF series capacitor.

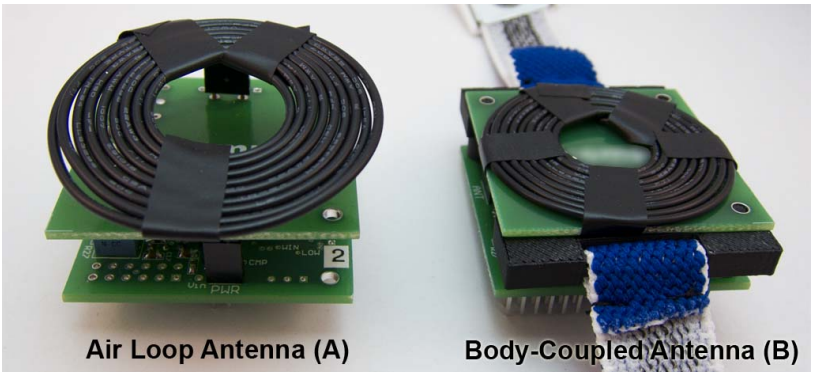

Figure 5.22: Air loop antenna (A) and body-coupled antenna (B) used in the whole-home C/N evaluation.

At higher frequencies, wearable sensor nodes typically suffer from significant attenuation from the human body [85]. To test the effect of the body for our 27 MHz inductive loop antenna, we conducted the following experiment in a 280 m^2 (*i.e.*, 3000 ft^2) home. We used two wireless sensor nodes shown in Figure 5.22, each using the same antenna design described above; however, one antenna was tuned to 27 MHz assuming there is no human body nearby, and the other was tuned assuming the coil was directly against the skin (*e.g.*, on the back of a watch). The tuning needs to be different because the human body capacitively loads the coil. Once tuned, the air loop antenna (A) has an impedance of 4.6 Ω, and the body coupled antenna (B) has an impedance of 190 Ω. To make the performance comparison between the two antennas fair, the transmit power of each node was tuned so that the total power consumption of each node while transmitting was 13.7 mW (*i.e.*, 4.57 mA at 3 V).

At 59 different locations around the home, we measured the carrier-to-noise ratio (C/N) from a spectrum analyzer coupled to the power line at the location indicated by the red triangle in Figure 5.23. At each location, a person wearing the body-coupled transmitter (B) on their wrist stood with his arms at his sides during the measurement. For the measurement using the air loop antenna (A), the node was placed on top of a non-conductive stand at the exact position, height, and orientation as the test with the body-coupled transmitter.

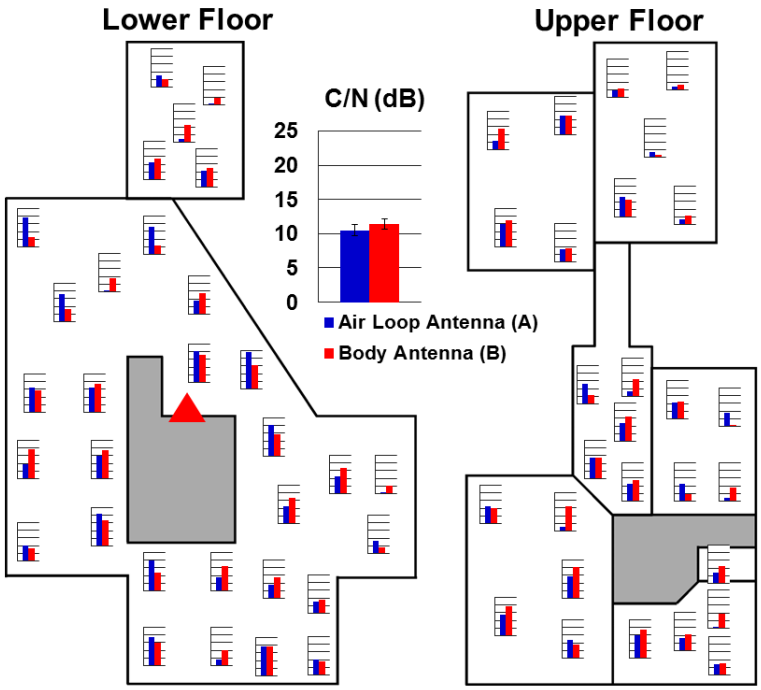

Figure 5.23: Body-coupled antenna performance results. Carrier of noise ratio (C/N) at 59 locations around a 280 m^2 home, using an air loop antenna (blue) and a body-coupled antenna (red). The large plot in the center serves as a legend for the small plots and shows the average C/N.

Figure 5.23 shows the C/N using both antenna configurations at all 59 locations around the home.

Evaluation Results

Figure 5.23 shows that the body antenna (B) actually performed better in 36 of the 59 locations, the air loop antenna was better in 21 locations, and they were equal in the remaining 2 locations. In addition, on average the body antenna (B) resulted in a 0.9 dB higher C/N. From this data it is clear that using the body as antenna does not result in significant performance degradation. In fact, the body antenna performed slightly better than standard air loop antenna.

There are a few contributing factors to success of the body-coupled antenna. Most

importantly, the body is much larger than the small air loop and therefore acts as a better low frequency antenna. In addition, the larger size of the body means that it is more likely to be closer to a power line than the small loop. For example, in locations where there are power lines running through the floor (*e.g.*, on the upper floor), the user's feet can efficiently couple to the wires. To demonstrate this effect, we placed a user at a location with a very poor C/N, even when an extension cord plugged into the wall was near his feet. Without changing the location of the transmitter or the extension cord, the user stepped on the extension cord resulting in a 10 dB increase in the carrier amplitude. A similar effect was observed at several locations when the user's body approached existing lamps.

5.5 Sensor Node Comparison

To evaluate the power advantages of SNUPI, it is essential to compare its power consumption to that of other similar sensor nodes. Since the application of the sensor node determines the required throughput of the data channel, we compare the *average power consumption* of each sensor node as a function of the *data throughput*. Our intended application area is whole-home sensing, and therefore we only evaluate nodes that work in this environment. In addition, this comparison is done independently of the sensor and the power supply (*i.e.*, only the power consumption of the microcontroller, ADC, and transmitter are considered), and all nodes are configured to run in transmit-only mode.

Comparing sensor nodes based on their published data is difficult and certainly not unbiased, and therefore we directly measure the parameters of interest in a lab environment from each of the sensor nodes. The parameters of interest include the power consumption and time period of each of the two phases of operation, the on-phase (P_{on}, t_{on}), and the sleep phase (P_{sleep}). During the on-phase, a single packet of data is transmitted, and contains n_{tx} bits of data, excluding the packet header and footer. Using these measured parameters, we calculate the average power consumption of each sensor node as a function of the data

throughput (R), expressed in bits per unit time:

$$P_{\text{avg}} = \frac{R\,t_{\text{on}}}{n_{\text{tx}}}\,P_{\text{on}} + \left(1 - \frac{R\,t_{\text{on}}}{n_{\text{tx}}}\right) P_{\text{sleep}} \quad \text{for } \frac{R\,t_{\text{on}}}{n_{\text{tx}}} \leq 1$$

When comparing the power consumption of several sensor nodes, it is important to attempt to normalize the performance of each node. For example, one node may consume significantly more power, yet it achieves a much greater range, and has a more reliable communications protocol. In this comparison we attempt to equalize the transmission range and the reliability of the nodes under consideration. We compare the average power consumption of SNUPI to several popular commercial wireless sensor nodes for whole-home sensing, the ZigBee-based SunSPOT[7] and the Crossbow[8] Motes. All nodes were configured to run in *transmit-only* mode in order to make a fair comparison to SNUPI, which does not have a receiver.

Table 5.13 shows the parameters measured from each of the sensor nodes, and Figure 5.24 plots the average power consumption of each sensor node as a function of the throughput using the equation above. In order to express these results in a more tangible way, we assume that all of the sensor nodes are powered from a 3 V, 225 mAh battery, and then we calculate the battery life of each node (see Figure 5.24) using a simplified model of battery life, where Q is the battery capacity and V is the battery voltage:

$$t_{\text{life}} = \frac{Q\,V}{P_{\text{avg}}}$$

It can be seen from the sensor node comparison shown in Figure 5.24 that at high throughput rates, the transmit power dominates, but for low throughput applications, it is the sleep power that determines the battery life of the node. From Figure 5.24, it is clear that the commercial nodes can operate at higher throughput rates than SNUPI, but for the

[7] SunSPOT, `http://www.sunspotworld.com`

[8] Crossbow Technology, Inc., `http://bullseye.xbow.com:81/index.aspx`

Sensor Node (Frequency)	P_{on} (mW)	P_{sleep} (µW)	t_{on} (ms)	n_{tx} (bits)
SNUPI (27 MHz)	1.00	1.5	6.6	16
SunSPOT (2.4 GHz)	153	108.9	36	816
Crossbow Mica (916 MHz)	58.8	18900	16	232
Crossbow Mica2 (433 MHz)	36.3	42	10.6	232
Crossbow MicaZ (2.4 GHz)	65.1	7.2	41	928

Table 5.13: Measured power consumption parameters for each sensor node under comparison.

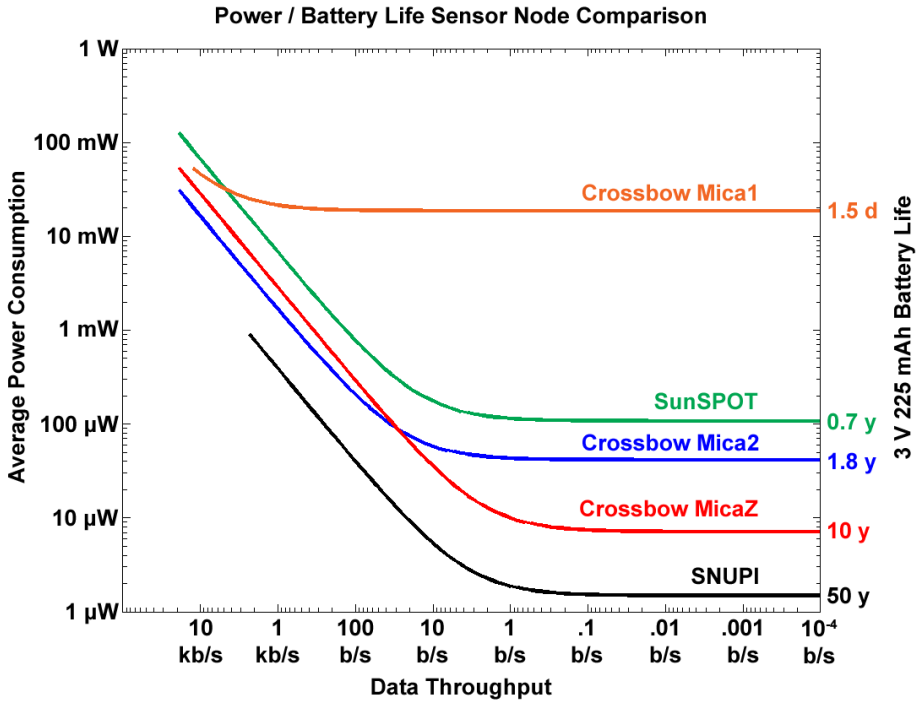

Figure 5.24: Average power consumption and battery life over throughput for five sensor nodes.

entire range of throughput for which SNUPI operates, it consumes significantly less power. The sleep power of SNUPI is so low that without a sensor attached, the calculated battery life of the node is 50 years, which is far longer than the shelf life of the battery. Of course, a sensor can dramatically reduce the battery life of the node, but the SNUPI node still has a battery life five times longer than the next best sensor node.

In addition, we compare SNUPI to Bumblebee, a state-of-the-art 433 MHz wireless sensor node in the research community, which to our knowledge is the lowest power wireless sensor node in existence [180]. Bumblebee was designed for a very short range ($<$ 15 m) neural tag, and is therefore not suitable for whole-home applications. In addition, Bumblebee does not incorporate a general purpose microcontroller, but rather has a fixed ADC integrated onto the radio chip. Bumblebee has a startup time very comparable to SNUPI, but since it does not have a full microcontroller, the total node transmission power is significantly lower: 500 μW, compared to 1000 μW for SNUPI. Since these nodes are very different, it makes more sense to compare only the transmitter of each device. Bumblebee has 433 MHz radio that can transmit 15 m and consumes 400 μW, while SNUPI has a 27 MHz radio that can transmit data to anywhere within a 280 square meter home using only 65 μW of power.

Chapter 6

SNUPI BASE STATION RECEIVER

This chapter describes the hardware and software design of the base station receiver used in a SNUPI network. The purpose of the receiver is to extract RF signals from the power lines, and decode the information present in each packet. Although there are many different architectures which achieve these goals, this section will describe one of the earliest implementations of the SNUPI receiver, on which all current implementations are based. This chapter will divide the receiver into three main sections. Section 6.1 discusses the analog circuitry which interfaces the RF receiver to the power line infrastructure. Section 6.2 describes the analog hardware of the RF receiver which demodulates the signal at the carrier frequency. Finally, Section 6.3 describes the software that is responsible for detecting and decoding the packets.

6.1 Power Line Interface

In the SNUPI receiver, the antenna is the power line, and therefore a circuit is needed to impedance match the power line to the receiver front end at the carrier frequency and to significantly attenuate the 60 Hz[1] signal used to carry power to appliances on the power line. In order to impedance match to the power lines, their impedance needs to be known. Early experiments show a wide variety of different impedances on the power line from one place to another. In addition, the impedance of the power line changes dynamically as the loads are switched on and off, and as the power line infrastructure changes due to whole circuits being switched in and out.

As as result of the variability in the power line impedance, current SNUPI implementa-

[1] 50 Hz is some countries. For brevity, I will refer to this is as just 60 Hz

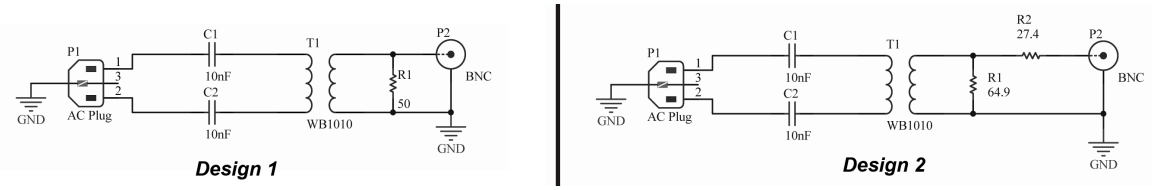

Figure 6.1: Power line interface schematics, showing Design 1 on the left, and Design 2 on the right. Design 2 has more loss at 27 MHz, but can handle more variability in the power line impedance, and gives a better impedance match to the 50 Ω receiver front-end.

tions do not attempt to match to the power line, they simply try to reduce the effect of these variations. Future designs may include an active impedance matching network which continually probes the power line impedance and adapts the matching network when the power line impedance changes. Although such systems are currently in use for some power line communications devices, no SNUPI implementation has attempted this type of network so far.

Figure 6.1 shows the schematic for two basic power line interfaces used in early SNUPI implementations. Figure 6.2 shows the frequency response of Design 2. It can be seen from Figure 6.2 that this design attenuates the 60 Hz signal by at least 60 dB, while allowing the RF signal at 27 MHz to pass though with less than 12 dB of attenuation. The frequency response for Design 1 is similar to that of Design 2. However, Design 2 has more loss at 27 MHz, but can handle more variability in the power line impedance, and gives a better impedance match to the 50 Ω receiver front-end. For these reasons, all future SNUPI implementations use a design similar to that of Design 2.

Note that both of the designs shown in Figure 6.1 connect to the power line through the line-to-neutral (LN) connection. However, the PLCW channel analysis described in Chapter 3 recommended using the line-to-ground (LG) connection when it is available. As a result of this, most of the more recent SNUPI implementations have used the LG connection. It was also seen in the channel analysis that the optimal power line connection varies between different homes, and even depends on which appliances are on. As a result,

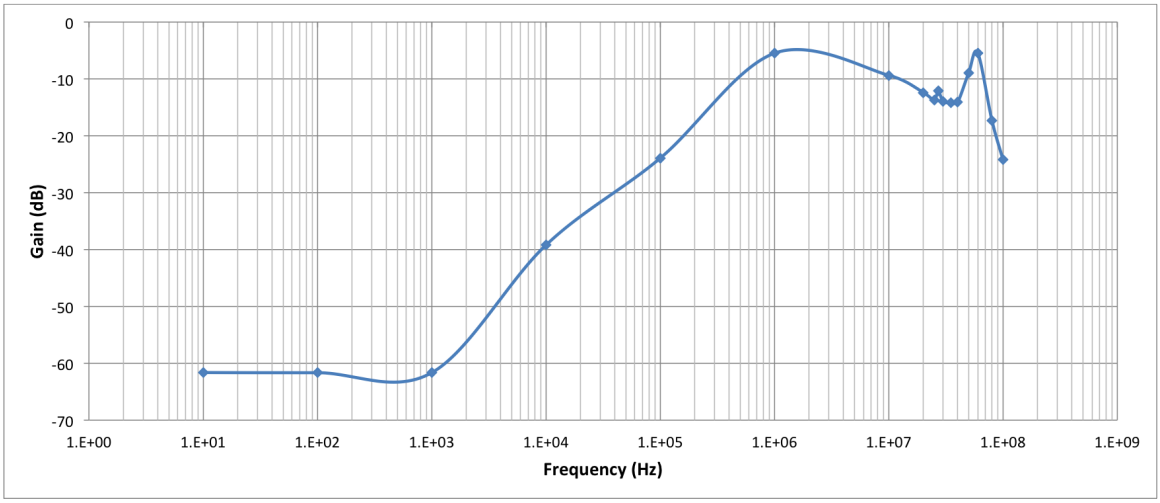

Figure 6.2: Frequency response of Design 2 of the power line interface.

it would be optimal to use all power line connections in a diversity combining receiver. Although this requires completely replicating the entire receiver pipeline three times, it may suffices to simply add two or more of the power line connections, or take some other linear combination of them. Alternatively, the different power line connections could be multiplexed in order to effectively monitor all three, but each at a lower rate. The exploration of these additional steps to improve the power line connection remains future work.

6.2 RF Demodulation

This section describes the analog hardware needed to demodulate the RF signal. A single-conversion super-heterodyne receiver architecture is used, as shown in the block diagram in Figure 6.3. This section described the earliest implementation of the SNUPI receiver, which is built around the Melexis[2] TH7122 analog integrated circuit, which is designed to be an FSK/FM/ASK transceiver for 27-930 MHz. Although many of the descriptions in this text are specific to the TH7122, other implementations are possible. In fact, many of the newer SNUPI receivers are built around the NXP[3] SA607 analog front-end IC.

[2] `http://www.melexis.com`

[3] `http://www.nxp.com/`

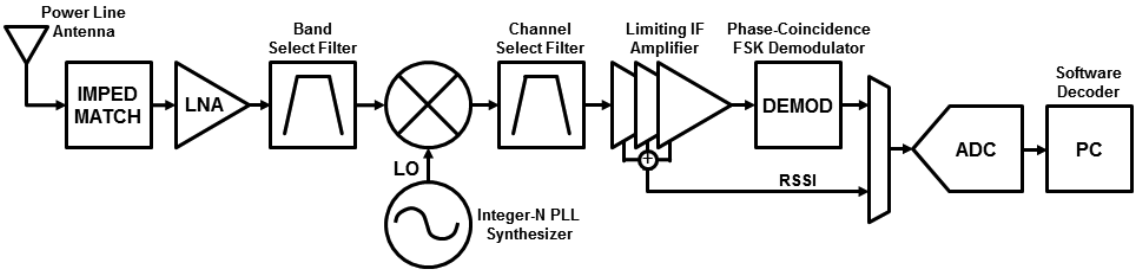

Figure 6.3: Block diagram of the SNUPI receiver.

Many of the analog blocks shown in Figure 6.3 are contained inside the TH7122 analog IC. A low-noise amplifier (LNA) inside the TH7122 is used to provide some initial gain and a band select filter further filters the 27 MHz carrier. The carrier is then mixed with the local oscillator (LO) to down-convert the carrier to the intermediate frequency (IF) of 10.7 MHz, which is sharply filtered with the channel select bandpass filter. The LO is generated and controlled using a phase-locked loop (PLL) frequency synthesizer inside the TH7122. A limiting IF amplifier (IFA) inside the TH7122 provides about 80 dB of gain and produces the received signal strength indicator (RSSI) signal. Finally, a phase-coincidence FSK demodulator inside the TH7122 is used to demodulate the data and produce the data output signal. An analog-to-digital converter (ADC) is used to digitize both the RSSI and data signals so that the remainder of the processing can be done in software. Figure 6.4 shows the full schematic of the printed circuit board, containing most of the receiver hardware. The following sections describe selected sub-blocks of this design.

Local Oscillator (LO) The local oscillator is an integer-N PLL frequency synthesizer which is part of the TH7122, as shown in Figure 6.5. The reference oscillator for the PLL uses an external $3,579,545$ Hz crystal resonator, and both R and N are controlled by software and programmed into the TH7122 using the serial communication interface. The PLL on the TH7122 is designed to produce a VCO frequency between 300 and 930 Hz. The external loop filter was modified to contain an additional varactor diode in order for

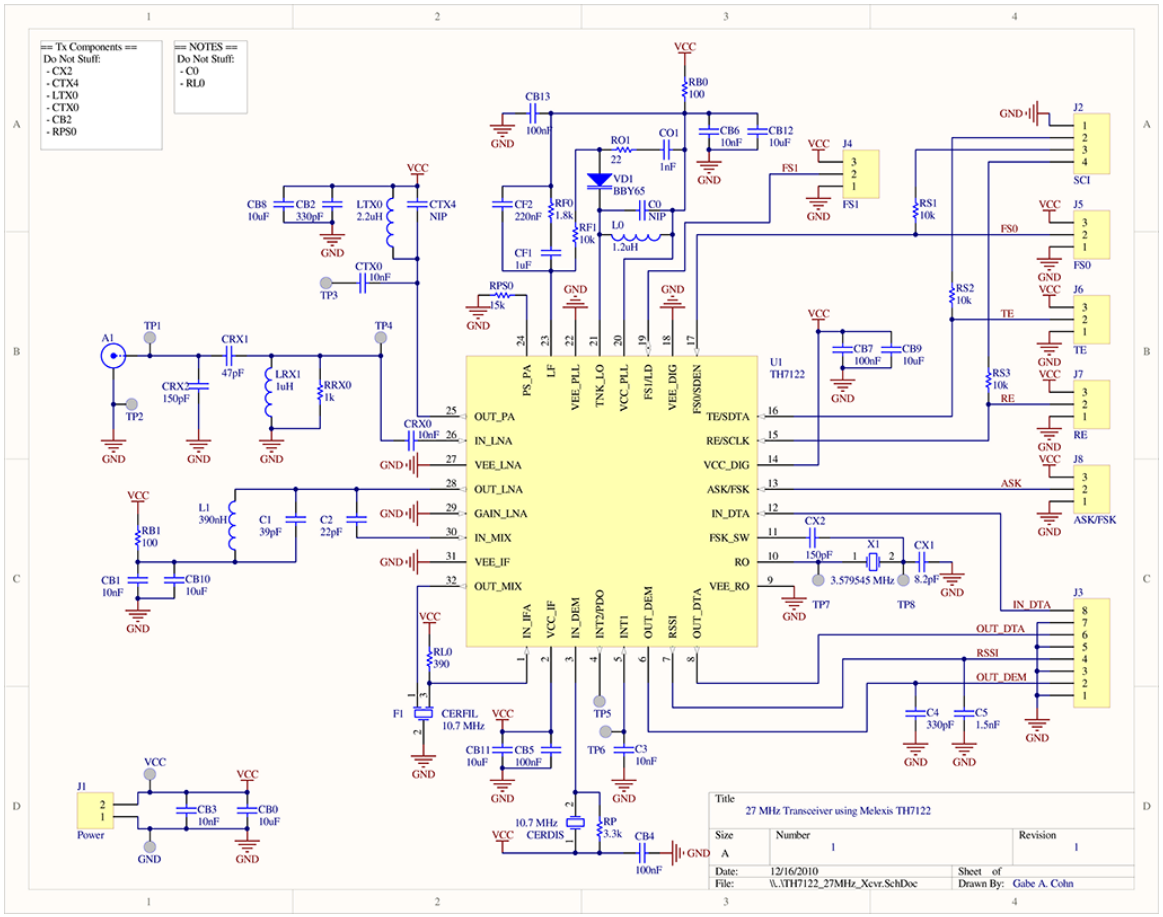

Figure 6.4: Schematic of TH7122 SNUPI receiver PCB.

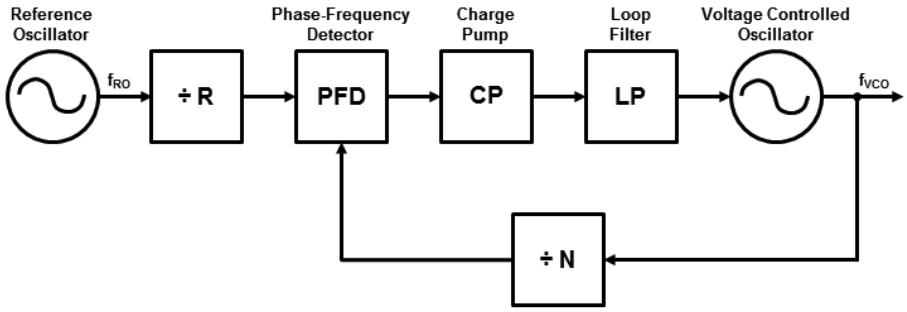

Figure 6.5: Block diagram of PLL frequency synthesizer used for the LO

the VCO frequency to operate around 27 MHz, as suggested in the TH7122 datasheet. Full schematics of the loop filter modification can be seen in Figure 6.4.

Note that the LO does not need to be generated with a PLL as shown in this implementation. In fact, later designs have reduced demodulation errors and jitter by using a custom cut crystal oscillator at the ideal LO frequency rather than using a PLL.

Band Select Filter The band select filter is a Murata CERAFIL ceramic filter with a center frequency of 10.7 MHz, and a bandwidth of 20 kHz. The bandwidth of this filter and all of the other filters in the receiver are very important. The filter needs to be just wide enough to let the RF communications signals pass through, but should not be any wider than necessary. Using a wider filter will let more noise into the receiver and thus make it much more difficult to decode weak signals.

Demodulator The phase-coincidence FSK demodulator is contained inside the TH7122, and uses an external 10.7 MHz Murata CERAFIL ceramic discriminator with a bandwidth of 155 kHz. In addition, there is a data slicer internal to the TH7122 with an RC time constant of 300 µs. Ideally, the discriminator should have a narrower bandwidth, although at the time, none were readily available. The bandwidth of the ceramic discriminator determines how much of a voltage deviation is produced by changes in frequency. In order to overcome noise, we want as much voltage deviation as possible, but also need to the

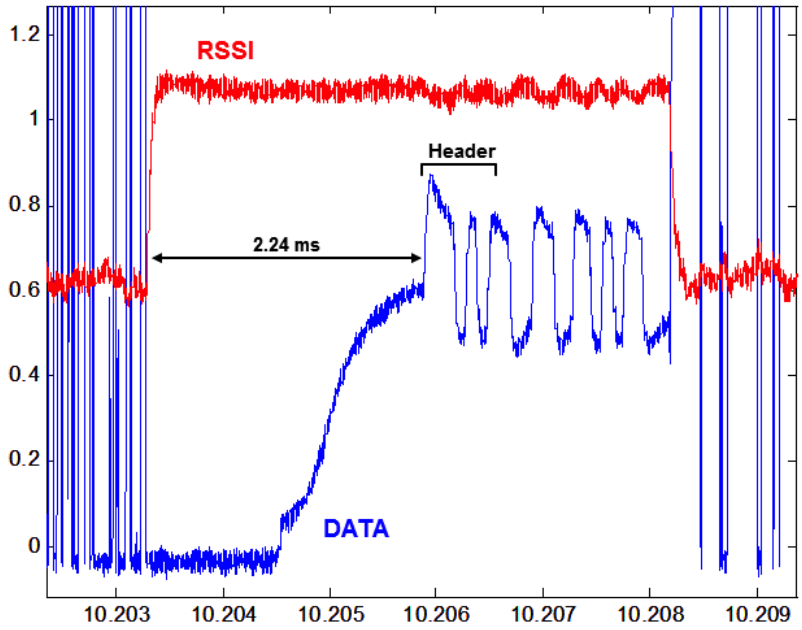

Figure 6.6: Example RSSI (red) and data (blue) signals at output of receiver hardware.

discriminator to cover the entire frequency band of interest.

Analog-to-Digital Converter (ADC) The two outputs of the receiver hardware are the received signal strength indicator (RSSI) and the demodulated data, which can be seen in Figure 6.6. Both of these signals need to be digitized so that they can be processed by the software decoder. In the current implementation, a National Instruments NI-DAQ USB-6259 is used to sample the RSSI and data signals at 240 kS/s, and digitize them with 16-bit resolution. The digital values are then sent to a host PC via USB for the remainder of the processing. Later, more integrated implementations of the SNUPI receiver use an on-board ADC and on-board embedded processor to implement the entire receiver pipeline.

Limitations Figure 6.6 shows an example of the digitized RSSI (red) and data (blue) signals. It should first be noted that the data is not sent until 2.24 ms after the transmitter turns on. This is the time it takes for the receiver hardware to be ready to demodulate the

data. This is an important limitation of the receiver and thus requires the transmitter to turn on 2.24 ms before it begins modulating the data. From the point of view of the software decoder, this is a nice feature in that it makes it easier to distinguish a valid packet from noise (*i.e.*, every valid packet must have a 2.24 ms rest period after the RSSI increases due to a transmission and before the data modulation begins). This limitation of the earliest implementation of the SNUPI receiver was later removed when transitioning away from the TH7122 to the SA607.

Another important limitation of the receiver hardware is the time constant of the data slicer in the demodulator. A steady-state data value tends to drift toward the mean over time, as seen in Figure 6.6. The time constant of the data slicer determines how quickly this drift occurs. In order for such a demodulator to work properly, the modulated data must therefore be run-length limited. That is, it cannot stay in either the high or low state for too long at any given time. In other words, there must be a maximum pulse width in the data signal. There are a set of run-length limited (RLL) codes discussed in Section 4.6.5 which are therefore ideally suited for this type of receiver. This limitation of the earliest implementation of the SNUPI receiver was later removed by using a DC-coupled gain stage after the discriminator, rather then the AC-coupled data slicer that is inside the TH7122.

6.3 Packet Decoder

The packet decoder is completely implemented in software and needs to take the sampled RSSI and FM (*i.e.*, demodulated data stream) voltage waveforms and extract and decode the packet data. The packet decoder consists of three main steps: (1) the packet parser which extracts valid packets from the live stream of data, (2) the decoder which strips off the channel coding to reveal the uncoded data, and (3) the packet checker which checks the validity of the packet and separates out each element from the packet structure. A block diagram of the packet decoder is shown in Figure 6.7. The initial implementation of the software decoder is written in C++/CLI (managed C++) and runs in real-time on a host PC. More recent implementations have been written in C++ in order to run in real-time on an

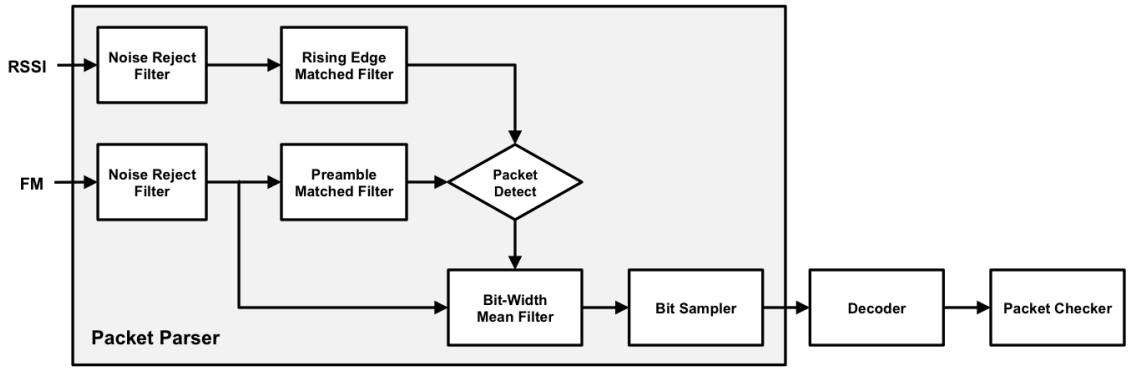

Figure 6.7: Block diagram for packet decoder.

embedded ARM processor.

6.3.1 Packet Parser

The packet parser is tasked with extracting potential packets out of the raw data stream. To do this, the parser can use two different data streams, the RSSI and the FM data. In the most general case, it should use some combination of the two, although many existing SNUPI implementations use one or the other. First, an FIR filter is used to remove noise from both data streams.

On the RSSI data stream, we are looking for a positive step change in the RSSI indicating that a packet has been received. The RSSI is the measure of the total received power seen by the receiver. Since the packet data will necessarily be above the noise, we expect to see a step change in the RSSI. Therefore we use a matched filter to search for the step change. The output of this matched filter represents the probability that the step change has been detected, and is passed to the packet detection decision block.

On the FM data stream, we are looking for a pattern that looks like a packet. Luckily, we used a specific preamble in the packet to help identify it, as described in Section 4.5. The most common preamble to use is a Barker code which has favorable autocorrelation properties and therefore stands out above the noise when using a matched filter to find it.

Again a matched filter is used to try to find the packet preamble, but care must be taken to normalize the amplitude and offset of the incoming waveform. The output of the matched filter is passed to the packet detection decision block.

The packet detection decision block needs to determine when a potential packet has been found using a threshold set on some linear combination of the RSSI edge detection and the FM preamble detection. Relying solely on the RSSI detection tends to miss packets with low SNR, while relying solely on the preamble detection tends to falsely trigger on random noise. It's important to realize that this is a hard decision, and therefore it is better to err on the side of false positives. If valid packets are missed, there is no way to recover the data, but if noise is misclassified as a packet, it will be caught by the decoder or packet checker later in the pipeline.

Once a packet is detected, we can start to extract the individual bits from the packet. Doing so requires the packet parser to be synchronized to the bitrate of the encoded packet. Luckily the packet's preamble, which is often a Barker code, allows us to do this easily and was already done as part of the packet detection algorithm. Once the synchronization is known, a mean filter (*i.e.*, running average) is used to compute the average over the width of one bit. The output of this filter is then sampled to get the value of each bit. Once a value has been determined for each bit, this data is sent to the decoder in order to remove the channel coding and extract the underlying data.

6.3.2 Decoder

The decoding process is greatly dependent on the type of source and channel coding used to encode the data. In order to illustrate an example decoder, this section will refer to the combined RLL/ECC code known as "French90-1c" that is described in Section 4.6.5. Since this is a convolutional code, it can be decoded using a Viterbi decoder. The state machine for the French90-1c code is shown in Figure 6.8.

The most straight forward approach would be to directly use the sequence of bits out of the packet parser along with a Viterbi decoder to extract the original data given the state

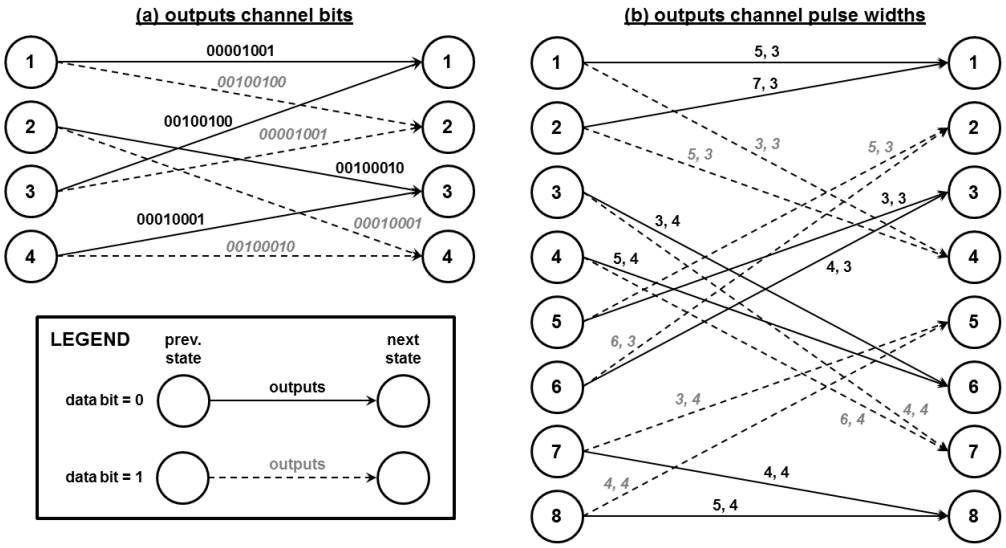

Figure 6.8: French90-1c combined ECC/RLL code state diagrams: (a) outputs channel bits, (b) outputs channel pulse widths.

machine for the French90-1c code, shown in Figure 6.8(*a*).

However, this approach is not always available. Although the packet parser described in the previous section outputs values for each bit, this is not the only type of packet parser. Some packet parsers, including the first one implemented for the SNUPI receiver detect the transitions between the bits rather than the values of each individual bit. Therefore, the output of the packer parser is a list of pulse widths. Although these can be simply converted to bits and then fed into the Viterbi decoder, this can lead to many errors. Small errors in the size of the pulses will produce missing or extra bits in the sequence of RLL bits and therefore errors would propagate creating far too many errors for the error-correction code to handle.

An alternative approach which is much more robust to small errors in the size of the pulses is to map the French90-1c state machine into the space of pulse widths rather than bits. The transformed state diagram is shown in Figure 6.8(*b*). Using this state machine, a Viterbi decoder is used which computes the error metric as the absolute difference between

the expected pulse width (*i.e.*, from the state machine) and the observed pulse width. The output of the Viterbi decoder represents the original packet data, which is finally passed to the packet checker.

Note that either a soft decision or hard decision decoder can be used, although performance should increase with the use of a soft decision decoder. In addition, some decoders, like the Viterbi decoder, output a metric which represents the likelihood that resultant bit sequence represents a real packet. This metric can therefore be used with a threshold to eliminate bad packets before encountering the packet checker.

6.3.3 Packet Checker

The packet checker will ensure that the data returned from the decoder has no errors. Often error checking codes like parity bits or CRCs are embedded into the packet structure to accommodate this check, as discussed in Section 4.5. In addition, the packet checker could check the node ID, packet ID, or data values for validity. If no errors are found, then the packet checker, which is aware of the packet structure, will break the packet down into each of its individual parts and return them to the application code.

Chapter 7

CONCLUSION

In this dissertation, I describe a novel indoor wireless sensor network called SNUPI (Sensor Network Utilizing Powerline Infrastructure) that overcomes the traditional power/ range trade-off posed by wireless sensor networks by leveraging the existing power line infrastructure in buildings. In this new wireless sensor network, a single base station receiver is connected directly to the power line (*i.e.*, plugged into an outlet). Each node in the sensor network transmits wireless signals that couple to nearby power lines, thus creating signals that travel through the power line infrastructure to the base station receiver. In this way, the sensor nodes can transmit at much lower power because the signals do not need to propagate over-the-air for the entire path to the base station receiver; they only need to propagate to the nearest power line. In this dissertation, I have built upon earlier work, using the power line as a receiving antenna for communications, to thoroughly investigate the details required to make such a wireless sensor network both practical and robust. I have explored this new power line coupled communications channel and discussed the designs and application-specific optimizations of each element of the sensor network, including the communications protocol, the sensor node, and the base station receiver.

SNUPI networks have a number of strengths over traditional wireless sensor networks. Much of the focus in this dissertation has been on the power consumption and lifetime advantages, but SNUPI's benefits are not limited to these aspects. SNUPI networks are ideally suited for many indoor applications in which it is preferable to place the sensor at the periphery (*e.g.*, behind the walls or under appliances). It is in these locations that traditional over-the-air wireless sensor networks tend to fail. They work best when there is a line of sight between the transmitter and the receiver, and therefore the center of a large

room is ideal. With a SNUPI network, the ideal locations are inverted. At the center of a large open room, SNUPI nodes are far from the power lines, and therefore in a position of very weak signal. However, when placed at the periphery, either along or inside the walls, or behind or under the appliances, SNUPI nodes thrive. These locations position the nodes very close the power line or very close to appliances which are themselves part of the electrical infrastructure. This is another important reason why SNUPI nodes are ideally suited for many indoor sensing applications.

Despite the many advantages of SNUPI networks in indoor environments, they still have a number of limitations, and will therefore never replace traditional over-the-air wireless sensor networks. It seems almost obvious, but is worth clarifying that SNUPI networks only work when in close proximity to the power line infrastructure. This eliminates the vast majority of applications in which wireless sensor networks have already been widely deployed in outdoor settings (see Section 2.1.1 for details). However, SNUPI networks also have severe limitations in indoor environments. Since the SNUPI network is currently unidirectional, the reliability and robustness of the network is extremely limited. Since there is no way for a node to be sure that its data has made it to the base station receiver, there are no guarantees on the reliability of the network. The use of multiple retransmissions (see Section 4.2) and forward error correction (see Section 4.6) can increase the probability that data arrives. In fact through statistical proof any arbitrary level of reliability can be obtained; however, it can never be certain that the data will arrive. Furthermore, the use of the HF band for the power line coupled wireless channel dramatically limits the bandwidth available for communications. As a result of this narrow bandwidth, SNUPI networks cannot take advantage of sub-channels in order to reduce the probability of collisions. SNUPI networks will also never be able to stream live audio or video media, as the HF band simply does not have enough bandwidth. For these reasons, SNUPI network will never replace traditional wireless sensor networks.

Despite the limitations of SNUPI networks, they have a number of advantages which allow them broaden the number of practical applications of wireless sensor networks. Al-

though sensor networks have been around for decades, they still have not made a stronghold in our homes. The smart home, home automation, and internet of things movements have helped to develop application domains in which household wireless sensor networks can dominate, yet none of these movements have yet gone mainstream. I'm very proud that my startup company, SNUPI Technologies, is currently working to develop smart home products which use the SNUPI wireless sensor network, but time will tell if the technology will have commercial success.

7.1 Summary of Contributions

In addition to the introduction of the power line coupled wireless channel and the SNUPI network in general, there are number of specific contributions that have been made throughout this dissertation. Some of these contributions apply only to the power line coupled wireless channel or the SNUPI network specifically, while others are more broadly applicable. This section will mirror the structure of the dissertation and point out some of the specific research contributions made in each chapter.

Studying the properties of the power line coupled wireless channel rank as some of the most important contributions toward the design of any SNUPI sensor network. My extensive 15 home background noise data collection, consisting of over 130 days and over 2 TB of background noise data will continue to provide useful insights into the power line coupled wireless channel for years to come. Although the analysis of this data presented in Section 3.2.2 is very preliminary, the dataset can be used to achieve a better understanding of the strengths and weaknesses of the channel. In addition, this dataset can be used for determining an algebraic noise model for the channel, and for determining the best type of forward error correction to use for SNUPI networks. In addition, another data collection effort yielded path loss measurements in 5–19 different locations in each of 15 homes. These path loss measurements help us answer questions about how to efficiently construct SNUPI networks, and how the network performance varies with location.

Thus far, my contributions on the communications channel have mostly been focused

on developing a bandwidth model which can used for optimizing the bitrate and frequency deviation for optimal performance. Additionally, much work has gone into the development of forward error correction schemes which are ideally suited to SNUPI networks, given the noise and interference present in the power line coupled wireless channel. Although much of the work on the forward error correction is in its early stages, I have set the groundwork for this type of research.

The vast majority of my contributions have been made in the development and investigation of the design of the SNUPI sensor node. This includes a battery discharge experiment lasting over 4 months, and scheduled to continue long after this dissertation is submitted. The battery testing and analysis, although still rather preliminary has contributions that go well beyond the design of SNUPI sensor nodes. A thorough understanding of the behavior and lifetime of batteries under very bursty loads is crucial in a number of wireless devices and other battery powered electronics. As leakage currents are reduced in modern electronics, the battery life of such devices will continue to grow, and it will become increasingly important to understand how the batteries will behave throughout their lifetime in such devices.

I have also made a number of contributions in the design and implementation of the wireless transmitter used on SNUPI sensor nodes. This includes both analog and digital implementations of the transmitter as well as an implementation which uses the MSP430 microcontroller as the primary component in the design of the transmitter. Although these designs were made specifically for the SNUPI sensor node, many of the ultra-low-power techniques can be applied to other transmitters, and the case of the MSP430-based design, they can even be applied to entirely different applications.

I also made a number of contributions in the design and modeling of electrically small wire loop antennas. Both my antenna model and my antenna measurements can be applied to other applications of these types of loop antennas.

7.2 Future Work

Although this dissertation encapsulates nearly 5 years of research and development on SNUPI wireless sensor network, there still much work to be done. This section will mirror the organization of the dissertation and highlight some of the main areas for future work in each chapter.

Chapter 3 describes an extensive data collection of the background noise on the power lines in 15 homes. Although Section 3.2.2 begins to analyze this data, there is a significant amount of further analysis to be done. First, the analysis discussed in Section 3.2.2 looks only at a 10 kHz bandwidth of noise; however, data was collected at 50 Hz, 100 Hz, 500 Hz, 1 kHz, 5 kHz, 10 kHz, and 50 kHz bandwidths. Each of these bandwidths should be individually analyzed and then compared. In addition to helping us gain an understanding of the channel properties of each bandwidth, this will help us answer questions about how much bandwidth the power line coupled wireless channel can support. It would also be very useful to try to develop an algebraic model of the noise on the power lines in order to better design communications protocols that are optimized to this channel. Such a model will need to incorporate the interference present on the power lines, and the bursty nature of the noise. To do this will require looking carefully at the raw data samples which were collected in addition to the noise statistics. It would also be very useful to look for patterns in the noise to see if there are any correlations with the time of day, or types of loads that are on in the home.

Much of the insight gained from a channel noise model can be directly applied to the communications protocol discussed in Chapter 4. As mentioned above, an algebraic noise model would allow for mathematical optimization of many aspects of the communications protocol. Most importantly, such a model would greatly aid in the development of the forward error correction. Although Section 4.6 discusses several different options for channel codes which can be applied to the SNUPI networks, all these different codes are simply compared based on the assumption of an AWGN channel. Instead, we need to test each of these codes on the PLCW channel. Since there are far too many codes to implement

and test in real SNUPI networks, the codes should be compared via Monte Carlo simulations. The raw noise samples that were collected in a number of homes can be used to run these simulations, although an algebraic model would still be greatly preferred. Once high performing channel codes have been identified through simulation, I'd like to implement them in real SNUPI networks to evaluate their real-world performance. Finally, I like to perform a comparison between the simulation results and implementation results in order to gain a better understanding of the strengths and weaknesses of the simulations. I would also like to look more deeply into using rateless codes like RCPC codes in order to help SNUPI networks recover from the loss of whole packets due to long noise bursts. I would also like to explore hybrid coding schemes which use both convolutional and block codes, as is commonly done in other communications networks.

As was mentioned several times in this dissertation, the biggest limitation of SNUPI networks is the reliability and robustness issues related to the unidirectional nature of the network. In the future, I'd like to explore ways in which the robustness can be improved by leveraging unique properties of the power line coupled wireless channel. One such property is the fact that the power lines also carry high voltage 60 Hz signals[1]. It has been shown that this 60 Hz wave radiates off of the power line, and can be sensed throughout the space inside of the building [42,43]. My assumption is that the strength of the 60 Hz radiated line noise is correlated with the proximity to the power line, and thus may be correlated to the path loss in a power line coupled wireless channel. In other words, if a SNUPI node can sense a very strong 60 Hz signal, then it may assume that there is low path loss between the node and the base station receiver. For fixed nodes (*i.e.*, nodes which are placed in a single location during the installation of the network and then don't move), I can use the amplitude of the 60 Hz line noise to provide feedback to the installer, perhaps in the form of LEDs. This feedback can help the installer place the node in a location in which is has a good quality link with the base station receiver. Remember that there is no receiver on the

[1] 50 Hz is some countries. For brevity, I will refer to this is as just 60 Hz

nodes, so there is no other way for them to know whether or not they are in a good location. For mobile nodes (*i.e.*, nodes which will move around within, and perhaps outside of the network area), I can use the amplitude of the 60 Hz line noise to allow them to send their data only when they have a sufficiently high quality link. It would be a waste of energy for the node to transmit data if it is unlikely for the data to be received by the base station. However, using the strength of the 60 Hz line noise, the node can determine when it is in a good location, and then only send its data at that time. This is particularly important because the node will never know if its data arrives successfully at the base station, and thus it would be nearly impossible to implement a SNUPI network of mobile nodes unless the nodes have some confidence that their data will be received.

An alternate way to provide some robustness to the SNUPI network is to provide a simple receiver to the SNUPI nodes, even if it is only for getting an acknowledgment that data was received by the base station. A simple way of doing this may be to modulate the power line load in order to send very low data rate acknowledgments to the SNUPI nodes. It is known that appliance noise on the power lines radiates and can be easily received anywhere inside the building [42,43]. I have found that the triacs found in dimmer switches emit large amounts of radiated noise and can be easily received over the air. Furthermore, the dimmer level, which corresponds to the phase angle at which the triac is activated can also be easily seen in the radiated noise. Since a sensor node can easily detect the presence and phase of the triac from the radiated noise, data can be encoded using a triac load on the power line. In the simplest form, the triac (or other load) can be turned on and off at some rate ≤ 60 times per second. This would allow communication at an extremely low rate of 60 b/s. Although this is horribly slow, it would enable acknowledgments to be sent back to the SNUPI nodes. By varying the phase in the 60 Hz wave at which the triac is triggered, data can be sent at a higher rate. In this case, we can send 60 *messages* per second in which each message can be an analog value (*i.e.*, the phase) or a digital value, simply by discretizing the phase. The ability to send acknowledgments to the SNUPI nodes opens the door on the number and types of applications which could benefit from SNUPI. The

ability to send acknowledgments allows ARQ-based error correction to be implemented in addition to a wide variety of MAC protocols.

Enhanced MAC protocols can also be enabled using time synchronization. The idea is to use the radiated 60 Hz line noise to time synchronize the nodes, as proposed by Rowe *et al.* [184]. The ability to have time synchronization between the SNUPI nodes allows them to communicate over the channel in a way in which they can minimize collisions using an enhanced MAC protocol, such as time division multiple access (TDMA) or Slotted-ALOHA. TDMA could be implemented in the form of a fixed time slot for each node, or a slotted-ALOHA MAC in which several nodes share each slot. It would also be very useful to use both the line synchronization and the data acknowledgment (described above) to implement more complicated MAC protocols on the SNUPI network.

The reliability and number of potential applications for SNUPI networks would increase dramatically if a full receiver could be implemented on the SNUPI nodes. For several years this idea has been criticized because doing so would require significant complexity and draw a significant amount of power, thus making the power benefits of SNUPI disappear. One of the reasons for this argument is that according to the FCC regulations, the base station has the same transmit power limit as the node's transmitter, and this means that the receiver on the nodes needs to be just as good as the one on the base station. It has been assumed that this is not feasible because the base station receiver is very complex and power hungry (since the network was designed to minimize power and complexity at the transmitter). Recently however, I am beginning to think that putting a low power receiver on the SNUPI nodes may in fact be feasible. First, although the base station would have the same power limit as the nodes, the current 13.56 MHz implementations of the node are transmitting at about 50 dB below the FCC limit. Furthermore, recent receiver performance analysis has shown that in many locations, the performance is limited by the noise and interference on the power lines, and the not the path loss. The receiver on the nodes would not see this background noise and thus could be able to receive lower power signals. I think this is a very promising direction for SNUPI that is certainly worth pursuing.

In addition to adding new capabilities to the SNUPI nodes, there are a number of improvements that can be made to the current designs. Although most SNUPI implementations use the discrete digital IC implementation of the transmitter described in Section 5.3.2, this design is not very efficient, and therefore the power consumption could be reduced dramatically by using a custom analog design. For example, the use of a class C or class E amplifier for the power amplifier would likely improve the efficiency greatly over the 7400s currently being used.

A significant amount of additional testing is needed to draw more conclusions from the battery analysis. Most important, we need to quantify the effect that the transmit time, duty cycle, and transmit current have on the available capacity of the battery, as this quantity is crucial to the current battery life model. It will also be very important to look at the effect of temperature variation on the battery model, as the current experimentation has completely ignored all temperature effects. Furthermore, the current testing and modeling is based on a constant resistance model of the load, while many wireless systems have a constant current load. I'd also like to look beyond the use of 3 V battery cells. For example, I think it would be very useful to build a SNUPI sensor node which can run off a single 1.5 V cell. In addition, I'd like to build a hybrid regulated/unregulated power supply as described in Section 5.1.4. I also think that SNUPI is an idea sensor network to leverage harvested energy either in combination with or in place of the battery, and would like to explore these possibilities further.

I would also like to continue to refine the antenna model described in Section 5.4. I'd like to perform better comparisons between the modeled antennas the actual antennas in order to identify the strengths and weaknesses of the model. I think it is also very important to study the mapping between different antenna parameters and the antenna's ability to couple its signals onto the power line. Although all antenna development to date has focused on inductively coupled antennas, I'd also like to explore electric field or capacitively coupled antennas. I suspect that capacitively coupled antennas will work particularly well for coupling signals onto the human body, as described in Section 5.4.7.

Finally, there are number of improvements that can be made to the base station receiver. Future designs of the receiver can include an active impedance matching network which continually probes the power line impedance and adapts the matching network when the power line impedance changes. I think it would also be very useful to measure the impedance on the power line over a long duration in a number of homes in order to understand the challenges and/or importance of such a design. I also think it would be useful to use all of the power line connections in a diversity combining receiver rather than using just a single connection. Although this requires completely replicating the entire receiver pipeline three times, it may suffice to simply add two or more of the power line connections, or take some other linear combination of them. Alternatively, the different power line connections could be multiplexed in order to effectively monitor all three, but each at a lower rate.

In addition to the research and development of each component of the SNUPI network, there are number of questions that arise from the fact that the exact mechanism with which the electromagnetic signals are coupled onto the power line is not well understood. For example, where does this coupling take place? Can the coupling occur anywhere along on the power lines, or only at appliances, switches, outlets, and junction boxes, where the wires separate away from each other? In addition, what is the effect of other grounded objects in the home? I have observed strong coupling to the power line infrastructure in locations with no nearby power lines, but instead copper plumbing (*e.g.*, in parts of a bathroom). It is hypothesized that this coupling onto the power line is occurring through the copper pipes, which themselves are grounded just like the power line infrastructure; however, the exact mechanism is still unknown. Similarly, we can postulate whether SNUPI networks will work better or worse in buildings with shielding wires (*e.g.*, most commercial buildings). If the coupling generally occurs along the length of a wire run, then shielded wire should dramatically degrade the performance of a SNUPI network; however, if it generally couples from the appliances, switches, outlets, and junction boxes, then no degradation will be seen. Furthermore, if the shielding, which itself is grounded, can also act as part of the receiving

antenna, then the system may even work better in these environments, just as has been observed for the plumbing infrastructure. Questions about the type of home construction also arise, since the coupling mechanism is not fully understood. Are there certain types of homes for which SNUPI doesn't work at all? The 15 homes in which the power line coupled wireless channel has been tested and analyzed were all from the same region (*i.e.*, the greater Seattle area) and therefore did not represent a wide variety of different home types. Through the commercialization effort, hundreds of homes around the U.S. have now been tested, but many more are needed in order to answer many of these fundamental questions.

There are also a number of practical concerns to be worked out when working with SNUPI networks. The first of which is the wireless range: how close to the power lines does the transmitter need to be in order to couple its signals onto the infrastructure? This is a very difficult question to answer in general; as it depends greatly on nature of the infrastructure at each location in addition to the placement and orientation of the transmitter. The best way to approach answering this question would be to break the path loss into two components: the loss in coupling to the power line and the attenuation along the power line path. Doing this experiment would not be too difficult, as the total path loss can be measured using the procedure described in Section 3.3. In addition, the attenuation inside of the power lines can be measured by injecting a signal of known amplitude at the position on the power line closest to the transmitter, and then measuring the received signal at the receiver. By subtracting (*i.e.*, in dB) the power line attenuation from the total path loss, the result will be the coupling loss. However, in order to get a generalizable result, these measurements would need to be done at many different locations around each of many homes, with different positions and orientations of the transmitter at each location.

In addition to range, sensor network practitioners often care about the density of nodes, or the number of sensor nodes that the network can support. Since the current implementations of SNUPI are unidirectional and share a single channel using the ALOHA protocol, the maximum number of SNUPI nodes is a function of their range, duty cycle, and the

amount of data loss that can be tolerated for a given application. The probability of packet loss can be computed as a function of the number of nodes within range and the duty cycle using the equation presented in Section 4.2. However, the key phrase here is the "number of nodes *within range*." This notion of range is different from the wireless range discussed in the previous paragraph. Here, I am referring to the range of a node's data either over the air or through the power line. For example, if several different SNUPI networks share a single power line infrastructure (*e.g.*, in an apartment building with a different network in each apartment), then collisions can still occur between these different networks since they share the same medium (*i.e.*, the power line). This begs the question: how far do SNUPI signals propagate within the power lines? Could networks in nearby, yet distinct buildings observe collisions from networks in other buildings? Although more study is needed to fully answer these questions, it is believed from initial testing and understanding of the systems in question that the SNUPI signals do not propagate through electrical meters or pole-top transformers. Therefore, individual buildings will not observe collisions from other SNUPI networks via conduction over the power lines; however, multi-tenant buildings (*e.g.*, apartments) will have these types of issues, and therefore fewer nodes can be supported per "sub-network." However, individual buildings can still observe collisions via over-the-air coupling between the wires in each building, or from a sensor node placed near the boundary between the two buildings. More experimentation is needed in order to fully answer many of these practical issues related to range and density.

In addition to further exploring the practical issues of SNUPI networks in general, future work should also focus on the applications of SNUPI networks. Although this dissertation focuses almost entirely on in-home SNUPI networks, they can also be applied to commercial and industrial applications. In these scenarios, the power line coupled wireless channel will likely look very different in terms of noise and interference, and different challenges will likely be faced. I hope that this dissertation stands a guide to anyone interested in designing future SNUPI networks in new applications domains in order to further push forward the boundary of technology and innovation.

BIBLIOGRAPHY

[1] Abowd, G. D., and Sterbenz, J. P. Final Report on the Inter-Agency Workshop on Research Issues for Smart Environments. *IEEE Personal Communications 7*, 5 (2000), 36–40.

[2] Abramson, N. The ALOHA System: Another Alternative for Computer Communications. In *Proceedings of the Fall Joint Computer Conference*, AFIPS '70 (Fall), ACM (New York, NY, USA, 1970), 281–285.

[3] Agre, J., and Clare, L. An Integrated Sensing Networks. *Computer 33*, 5 (2000), 106–108.

[4] Agre, J. R., Clare, L. P., Pottie, G. J., and Romanov, N. P. Development Platform for Self-Organizing Wireless Sensor Networks. In *AeroSense '99* (1999), 257–268.

[5] Ailamaki, A., Faloutos, C., Fischbeck, P. S., Small, M. J., and VanBriesen, J. An environmental sensor network to determine drinking water quality and security. *ACM SIGMOD Record 32*, 4 (Dec. 2003), 47–52.

[6] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. Wireless sensor networks: a survey. *Computer Networks 38*, 4 (2002), 393–422.

[7] ALERT Systems Organization. `http://www.alertsystems.org`.

[8] Alippi, C., and Galperti, C. An Adaptive System for Optimal Solar Energy Harvesting in Wireless Sensor Network Nodes. *IEEE Transactions on Circuits and Systems 55*, 6 (July 2008), 1742–1750.

[9] Andrews, J. P. Remote continuous physiological monitoring in the home. *Journal of Telemedicine and Telecare 2*, 2 (1996), 107–113.

[10] Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., Vora, A., and Miyashita, M. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks 46*, 5 (2004), 605–634.

[11] Assimakopoulos, C., and Pavlidou, N. An Enhanced Powerline Channel Noise Model. *WSEAS Transactions on Power Systems 1*, 1 (2006), 239–245.

[12] Aziz, O., Atallah, L., Lo, B., ElHelw, M., Wang, L., Yang, G., and Darzi, A. A Pervasive Body Sensor Network for Measuring Postoperative Recovery at Home. *Surgical Innovation 14*, 2 (2007), 83–90.

[13] Baggio, A. Wireless sensor networks in precision agriculture. In *ACM Workshop on Real-World Wireless Sensor Networks*, REALWSN 2005 (2005).

[14] Baker, C. R., Armijo, K., Belka, S., Benhabib, M., Bhargava, V., Burkhart, N., Minassians, A. D., Dervisoglu, G., Gutnik, L., Haick, M. B., Ho, C., Koplow, M., Mangold, J., Robinson, S., Rosa, M., Schwartz, M., Sims, C., Stoffregen, H., Waterbury, A., Leland, E. S., Pering, T., and Wright, P. K. Wireless Sensor Networks for Home Health Care. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)* (2007), 832–837.

[15] Balanis, C. A. *Antenna Theory: Analysis and Design*. John Wiley & Sons, New York, 1982.

[16] Bauer, M., Jendoubi, L., and Siemoneit, O. Smart Factory – Mobile Computing in Production Environments. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[17] Bauer, P., Sichitiu, M., Istepanian, R., and Premaratne, K. The Mobile Patient: Wireless Distributed Sensor Networks for Patient Monitoring and Care. In *2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine* (2000), 17–21.

[18] Beckwith, R., Teibel, D., and Bowen, P. Report from the field: results from an agricultural wireless sensor network. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, 2004* (Nov. 2004), 471–478.

[19] Beigl, M., Krohn, A., Riedel, T., Zimmer, T., Decker, C., and Isomura, M. The uPart Experience: Building a wireless sensor network. In *5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, ACM (New York, NY, USA, 2006), 366–373.

[20] Beutel, J., Kasten, O., Mattern, F., Römer, K., Siegemund, F., and Thiele, L. Prototyping Wireless Sensor Network Applications with BTnodes. In *Proceedings*

of the First European Workshop on Wireless Sensor Networks, EWSN 2004, Springer-Verlag (Berlin, Heidelberg, 2004), 323–338.

[21] Beutel, J., Kasten, O., and Ringwald, M. Poster Abstract: BTnodes A Distributed Platform for Sensor Nodes. In *1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, ACM (New York, NY, USA, 2003), 292–293.

[22] Bhushan, M., and Rengaswamy, R. Comprehensive Design of a Sensor Network for Chemical Plants Based on Various Diagnosability and Reliability. *Industrial & Engineering Chemistry Research 41*, 7 (2002), 1826–1839.

[23] Bhuvaneswari, P., Balakumar, R., Vaidehi, V., and Balamuralidhar, P. Solar Energy Harvesting for Wireless Sensor Networks. In *Proceedings of the First International Conference on Computational Intelligence, Communication Systems and Networks, 2009*, CICSYN '09 (July 2009), 57–61.

[24] Biagioni, E. S., and Bridges, K. The Application of Remote Sensor Technology To Assist the Recovery of Rare and Endangered Species. *International Journal of High Performance Computing Applications 16*, 3 (2002), 315–324.

[25] Bonnet, P., Gehrke, J., and Seshadri, P. Querying the Physical World. *IEEE Personal Computing 7*, 5 (2000), 10–15.

[26] Buettner, M., Greenstein, B., Sample, A., Smith, J. R., Wetherall, D., et al. Revisiting smart dust with RFID sensor networks. In *Proceedings of the 7th ACM Workshop on Hot Topics in Networks (HotNets-VII)* (2008).

[27] Buettner, M., Prasad, R., Sample, A., Yeager, D., Greenstein, B., Smith, J. R., and Wetherall, D. RFID Sensor Networks with the Intel WISP. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, ACM (New York, NY, USA, 2008), 393–394.

[28] Bulusu, N., Estrin, D., Girod, L., and Heidemann, J. Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems. In *6th International Symposium on Communication Theory and Applications (ISCTA '01)* (2001).

[29] Burns, A., Greene, B., McGrath, M., O'Shea, T., Kuris, B., Ayer, S., Stroiescu, F., and Cionca, V. SHIMMER™ - A Wireless Sensor Platform for Noninvasive Biomedical Research. *IEEE Sensors Journal 10*, 9 (Sept. 2010), 1527–1534.

[30] Burrell, J., Brooke, T., and Beckwith, R. Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Computing 3*, 1 (January–March 2004), 38–45.

[31] Butler, Z., Corke, P., Peterson, R., and Daniela, R. Networked Cows: Virtual fences for Controlling Cows. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[32] Cai, J., Ee, D., Pham, B., Roe, P., and Zhang, J. Sensor Network for the Monitoring of Ecosystem: Bird Species Recognition. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007*, ISSNIP 2007 (Dec. 2007), 293–298.

[33] Cardell-Oliver, R., Kranz, M., Smettem, K., and Mayer, K. A Reactive Soil Moisture Sensor Network: Design and Field Evaluation. *International Journal of Distributed Sensor Networks 1*, 2 (2005), 149–162.

[34] Celler, B., Hesketh, T., Earnshaw, W., and Ilsar, E. An instrumentation system for the remote monitoring of changes in functional health status of the elderly at home. In *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1994* (1994), 908–909.

[35] Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., Zhao, J., and Rey, M. Habitat Monitoring: Application Driver for Wireless Communications Technology. *ACM SIGCOMM Computer Communications Review 31*, 2 (2001), 20–41.

[36] Chen, A., Muntz, R. R., Yuen, S., Locher, I., Park, S. I., and Srivastava, M. B. A Support Infrastructure for the Smart Kindergarten. *IEEE Pervasive Computing 1*, 2 (2002), 49–57.

[37] Chen, S.-L., Lee, H.-Y., Chen, C.-A., Lin, C.-C., and Luo, C.-H. A Wireless Body Sensor Network System for Healthcare Monitoring Application. In *Proceedings of IEEE Biomedical Circuits and Systems Conference*, BIOCAS 2007 (Nov. 2007), 243–246.

[38] Chiasserini, C., and Rao, R. A model for battery pulsed discharge with recovery effect. In *IEEE Wireless Communications and Networking Conference 1999 (WCNC 1999)* (1999), 636–639 vol.2.

[39] Chiasserini, C. F., and Rao, R. R. Pulsed Battery Discharge in Communication Devices. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, ACM (New York, NY, USA, 1999), 88–95.

[40] Chintalapudi, K., Fu, T., Paek, J., Kothari, N., Rangwala, S., Caffrey, J., Govindan, R., Johnson, E., and Masri, S. Monitoring civil structures with a wireless sensor network. *IEEE Internet Computing 10*, 2 (March-April 2006), 26–34.

[41] Cohn, G., Gupta, S., Froehlich, J., Larson, E., and Patel, S. N. GasSense: Appliance-Level, Single-Point Sensing of Gas Activity in the Home. In *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive '10, Springer-Verlag (Berlin, Heidelberg, 2010), 265–282.

[42] Cohn, G., Morris, D., Patel, S., and Tan, D. Humantenna: Using the Body as an Antenna for Real-Time Whole-Body Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 1901–1910.

[43] Cohn, G., Morris, D., Patel, S. N., and Tan, D. S. Your Noise is My Command: Sensing Gestures Using the Body as an Antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (New York, NY, USA, 2011), 791–800.

[44] Cohn, G., Stuntebeck, E., Pandey, J., Otis, B., Abowd, G. D., and Patel, S. N. SNUPI: Sensor Nodes Utilizing Powerline Infrastructure. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010)*, ACM (Copenhagen, Denmark, 2010), 159–168.

[45] Corke, P., Valencia, P., Sikka, P., Wark, T., and Overs, L. Long-duration solar-powered wireless sensor networks. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, ACM (New York, NY, USA, 2007), 33–37.

[46] Cotton, S. L., D'Errico, R., and Oestges, C. A Review of Radio Channel Models for Body Centric Communications. *Radio Science* (2014), n/a–n/a.

[47] Crossbow Technoloy, Inc. *Imote2 Datasheet*. 4145 North First Street, San Jose, CA 95134-2109.

[48] Crossbow Technoloy, Inc. *IRIS Datasheet*. 4145 North First Street, San Jose, CA 95134-2109.

[49] Crossbow Technoloy, Inc. *MICA2 Datasheet*. 41 Daggett Drive, San Jose, CA 95134-2109.

[50] Crossbow Technoloy, Inc. *MICA2DOT Datasheet*. 41 Daggett Drive, San Jose, CA 95134-2109.

[51] Crossbow Technoloy, Inc. *MICAz Datasheet*. 4145 North First Street, San Jose, CA 95134-2109.

[52] Dementyev, A., Hodges, S., Taylor, S., and Smith, J. Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario. In *2013 IEEE International Wireless Symposium (IWS)* (April 2013), 1–4.

[53] Diamond, S. M., and Ceruti, M. G. Application of Wireless Sensor Network to Military Information Integration. In *Proceedings of the 5th IEEE International Conference on Industrial Informatics, 2007* (june 2007), 317–322.

[54] Digi International. *XBee®/XBee-PRO® RF Modules Datasheet*, Sept. 2009.

[55] Dinh, T. L., Hu, W., Sikka, P., Corke, P., Overs, L., and Brosnan, S. Design and Deployment of a Remote Robust Sensor Network: Experiences from an Outdoor Water Quality Monitoring Network. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks, 2007*, LCN 2007 (Oct. 2007), 799–806.

[56] Doukas, C., and Maglogiannis, I. Advanced patient or elder fall detection based on movement and sound data. In *Proceedings of the Second International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth 2008, IEEE (Jan. 2008), 103–107.

[57] Dulman, S., and Havinga, P. Operating System Fundamentals for the EYES Distributed Sensor Network. In *3D PROGRESS Workshop on Embedded Systems* (2002), 34–37.

[58] Dutta, P., Grimmer, M., Arora, A., Bibyk, S., and Culler, D. Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events. In *4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, IEEE Press (Piscataway, NJ, 2005).

[59] EnOcean GmbH. *Transceiver Module TCM 120 User Manual V1.53*, 1.53 ed., Aug. 2008.

[60] EnOcean GmbH. *Scavenger Transceiver Module STM 300 / STM 300C / STM 300U User Manual V1.33*, 1.33 ed., Nov. 2012.

[61] EnOcean GmbH. *Transceiver Module TCM 300 / TCM 320 User Manual 9 (v.1.29)*, 1.29 ed., Oct. 2012.

[62] Erol-Kantarci, M., and Mouftah, H. Wireless Sensor Networks for Cost-Efficient Residential Energy Management in the Smart Grid. *IEEE Transactions on Smart Grid 2*, 2 (June 2011), 314–325.

[63] Essa, I. A. Ubiquitous Sensing for Smart and Aware Environments. *IEEE Personal Communications 7*, 5 (2000), 47–49.

[64] Estrin, D., Girod, L., Pottie, G., and Srivastava, M. Instrumenting the World with Wireless Sensor Networks. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)* (2001), 2033–2036.

[65] Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. Next Century Challenges: Scalable Coordination in Sensor Networks. In *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, ACM (New York, NY, USA, 1999), 263–270.

[66] European Conference of Postal and Telecommunications Administrations (CEPT), Electronic Communications Committee (ECC). *ERC Recommendatation 70-03: Relating to the use of Short Range Devices (SRD)*, 2012.

[67] Federal Communications Commission (FCC). *Rules and Regulations*, 2012.

[68] Ferreira, J. Appropriate modelling of conductive losses in the design of magnetic components. In *21st Annual IEEE Power Electronics Specialists Conference, 1990 (PESC '90)* (1990), 780–785.

[69] Fleming, R. Line Cord Antenna, 1965. United States Patent #3324473.

[70] French, C. Distance Preserving Run-Length Limited Codes. *IEEE Transactions on Magnetics 25*, 5 (Sept. 1989), 4093–4095.

[71] French, C., and Lin, Y. Performance Comparison of Combined ECC/RLL Codes. In *Proceedings of the IEEE International Conference on Communications, 1990*, ICC '90 (Apr. 1990), 1717–1722.

[72] Froehlich, J., Larson, E., Campbell, T., Haggerty, C., Fogarty, J., and Patel, S. N. HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, Ubicomp '09, ACM (New York, NY, USA, 2009), 235–244.

[73] Gabriel, S., R W Lau, R. W., and Gabriel, C. The dielectric properties of biological tissues: II. Measurements in the frequency range 10 Hz to 20 GHz. *Physics in Medicine and Biology 41*, 11 (1996), 2251.

[74] Gauger, M., Minder, D., Marrón, P. J., Wacker, A., and Lachenmann, A. Prototyping sensor-actuator networks for home automation. In *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, REALWSN '08, ACM (New York, NY, USA, 2008), 56–60.

[75] Gaynor, M., Moulton, S., Welsh, M., Lacombe, E., Rowan, A., and Wynne, J. Integrating wireless sensor networks with the grid. *IEEE Internet Computing 8*, 4 (July–Aug 2004), 32–39.

[76] Gilbert, J., and Balouchi, F. Comparison of energy harvesting systems for wireless sensor networks. *International Journal of Automation and Computing 5* (2008), 334–347.

[77] Gill, K., Yang, S.-H., Yao, F., and Lu, X. A ZigBee-Based Home Automation System. *IEEE Transactions on Consumer Electronics 55*, 2 (May 2009), 422–430.

[78] Glatz, P., Hörmann, L., Steger, C., and Weiss, R. Designing Perpetual Energy Harvesting Systems Explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring. *Electronic Journal of Structural Engineering, Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application* (2010), 55–66.

[79] Goel, S., Imielinski, T., Ozbay, K., and Nath, B. Poster Abstract: Sensors on wheels – Towards a Zero-Infrastructure Solution for Intelligent Transportation. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, ACM (New York, NY, USA, 2003), 338–339.

[80] Gollakota, S., Reynolds, M., Smith, J., and Wetherall, D. The Emergence of RF-Powered Computing. *Computer 47*, 1 (Jan 2014), 32–39.

[81] Grandi, G., Kazimierczuk, M., Massarini, A., and Reggiani, U. Stray capacitances of single-layer air-core inductors for high-frequency applications. In *Conference Record of the 1996 IEEE Industry Applications Conference (IAS '96)*, vol. 3 (Oct 1996), 1384–1388 vol.3.

[82] Gungor, V., Lu, B., and Hancke, G. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *IEEE Transactions on Industrial Electronics 57*, 10 (Oct. 2010), 3557–3564.

[83] Gupta, S., Reynolds, M. S., and Patel, S. N. ElectriSense: Single-Point Sensing using EMI for Electrical Event Detection and Classification in the Home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, ACM (New York, NY, USA, 2010), 139–148.

[84] Hagenauer, J. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions on Communications 36*, 4 (Apr 1988), 389–400.

[85] Hall, P. S., and Hao, Y. Antennas and propagation for body centric communications. In *2006 European Conference on Antennas and Propagation (EuCAP 2006)* (Nov 2006), 1–7.

[86] Han, D.-M., and Lim, J.-H. Design and Implementation of Smart Home Energy Management Systems Based on ZigBee. *IEEE Transactions on Consumer Electronics 56*, 3 (Aug. 2010), 1417–1425.

[87] He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J. A., Abdelzaher, T. F., Hui, J., and Krogh, B. VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions Sensor Networks (TOSN) 2*, 1 (Feb. 2006), 1–38.

[88] Hill, J., Horton, M., Kling, R., and Krishnamurthy, L. The Platforms Enabling Wireless Sensor Networks. *Communications of the ACM 47*, 6 (2004), 41–46.

[89] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. System Architecture Directions for Networked Sensors. *SIGPLAN Not. 35*, 11 (Dec. 2000), 93–104.

[90] Hill, J. L. *System Architecture for Wireless Sensor Networks by*. Phd, University of California Berkeley, 2003.

[91] Hill, J. L., and Culler, D. E. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro 22*, 6 (2002), 12–24.

[92] Hoang, D., Tan, Y., Chng, H., and Panda, S. Thermal energy harvesting from human warmth for wireless body area network in medical healthcare system. In *Proceedings of the International Conference on Power Electronics and Drive Systems, 2009.*, PEDS 2009 (Nov. 2009), 1277–1282.

[93] Hori, T., Nishida, Y., Aizawa, H., Murakami, S., and Mizoguchi, H. Sensor network for supporting elderly care home. In *Proceedings of IEEE Sensors, 2004* (Oct. 2004), 575–578.

[94] Huang, G. T. Casting the Wireless Sensor Net, 2003.

[95] Huang, H., Xiao, S., Meng, X., and Xiong, Y. A remote home security system based on wireless sensor network and gsm technology. In *2010 Second International Conference on Networks Security Wireless Communications and Trusted Computing*, NSWCTC '10 (Apr. 2010), 535–538.

[96] Huang, J.-H., Amjad, S., and Mishra, S. CenWits: A Sensor-Based Loosely Coupled Search and Rescue System Using Witnesses. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, ACM (New York, NY, USA, 2005), 180–191.

[97] Huang, M.-C., Huang, J.-C., You, J.-C., and Jong, G.-J. The Wireless Sensor Network for Home-Care System Using ZigBee. In *Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007*, IIHMSP 2007 (Nov. 2007), 643–646.

[98] Huo, H., Xu, Y., Yan, H., Mubeen, S., and Zhang, H. An elderly health care system using wireless sensor networks at home. In *Proceedings of the Third International Conference on Sensor Technologies and Applications, 2009*, SENSORCOMM '09 (June 2009), 158–163.

[99] Hymer, D., Moran, M., and Keefer, T. Soil Water Evaluation Using a Hydrologic Model and Calibrated Sensor Network. *Soil Science Society of America Journal 64*, 1 (2000), 319–326.

[100] Immink, K. Runlength-Limited Sequences. *Proceedings of the IEEE 78*, 11 (Nov. 1990), 1745–1759.

[101] Inoue, M., Higuma, T., Ito, Y., Kushiro, N., and Kubota, H. Network architecture for home energy management system. *IEEE Transactions on Consumer Electronics 49*, 3 (Aug. 2003), 606–613.

[102] International Telecommunication Union (ITU). *Radio Regulations*, 2012.

[103] Jacob, S. Crossbow Introduces IRIS Wireless Product Line. `http://www.businesswire.com/news/home/20070425005470/en/Crossbow-Introduces-IRIS-Wireless-Product-Line`, Apr. 2007.

[104] Johnstone, I., Nicholson, J., Shehzad, B., and Slipp, J. Experiences from a Wireless Sensor Network Deployment in a Petroleum Environment. In *Proceedings of the*

2007 International Conference on Wireless Communications and Mobile Computing (IWCMC '07), ACM (New York, NY, USA, 2007), 382–387.

[105] Kahn, J., Katz, R., and Pister, K. Next Century Challenges: Mobile Networking for Smart Dust. In *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, ACM (New York, NY, USA, 1999), 271–278.

[106] Kansal, A., and Srivastava, M. An environmental energy harvesting framework for sensor networks. In *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003*, ISLPED '03 (Aug. 2003), 481–486.

[107] Karl, H., and Willig, A. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Hoboken, NJ, 2005.

[108] Kasten, O., and Langheinrich, M. First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network. In *Workshop on Ubiquitous Computing and Communications (PACT 2001)* (2001).

[109] Khedo, K. K., Perseedoss, R., and Mungur, A. A Wireless Sensor Network Air Pollution Monitoring System. *International Journal of Wireless & Mobile Networks 2* (2010), 31–45.

[110] Kim, K., Jun, J., Kim, S., and Sung, B. Medical Asset Tracking Application with Wireless Sensor Networks. In *Proceedings of the Second International Conference on Sensor Technologies and Applications, 2008*, SENSORCOMM '08 (Aug. 2008), 531–536.

[111] Kim, Y.-G., Kim, H.-K., Lee, S.-G., and Lee, K.-D. Ubiquitous Home Security Robot Based on Sensor Network. In *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, IAT '06, IEEE Computer Society (Washington, DC, USA, 2006), 700–704.

[112] Kling, R., Adler, R., Huang, J., Hummel, V., and Nachman, L. Intel Mote-based sensor networks. *Structural Control and Health Monitoring 12*, 3-4 (July 2005), 469–479.

[113] Ko, J., Lu, C., Srivastava, M., Stankovic, J., Terzis, A., and Welsh, M. Wireless Sensor Networks for Healthcare. *Proceedings of the IEEE 98*, 11 (Nov. 2010), 1947–1960.

[114] Korber, H.-J., Wattar, H., and Scholl, G. Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications. *IEEE Transactions on Industrial Informatics* (May 2007), 111–119.

[115] Krishnamurthy, L., Adler, R., Buonadonna, P., Chhabra, J., Flanigan, M., Kushalnagar, N., Nachman, L., and Yarvis, M. Design and Deployment of Industrial Sensor Networks: Experiences from a Semiconductor Plant and the North Sea. In *Proceedings of the 3rd International Conference of Embedded Networked Sensor Systems (SenSys '05)*, ACM (New York, NY, USA, 2005), 64–75.

[116] Langendoen, K., Baggio, A., and Visser, O. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium, 2006*, IPDPS 2006 (Apr. 2006).

[117] Lee, D. Energy Harvesting Chip and the Chip Based Power Supply Development for a Wireless Sensor Network. *Sensors 8*, 12 (2008), 7690–7714.

[118] Lee, J.-H., and Hashimoto, H. Controlling mobile robots in distributed intelligent sensor network. *IEEE Transactions on Industrial Electronics 50*, 5 (Oct. 2003), 890–902.

[119] Lee, K. Y., and Choi, J. W. Remote-controlled home automation system via bluetooth home network. In *SICE 2003 Annual Conference* (Aug. 2003), 2824–2829.

[120] Lee, S. Crossbow Announces Crossbow Imote2, the Most Powerful Mote Platform for Wireless Sensor Network Research. `http://www.businesswire.com/news/home/20070123005191/en/Crossbow-Announces-Crossbow-Imote2-Powerful-Mote-Platform`, Jan. 2007.

[121] Lee, Y.-D., and Chung, W.-Y. Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring. *Sensors and Actuators B: Chemical 140*, 2 (2009), 390–395.

[122] León, R., Vittal, V., and Manimaran, G. Application of Sensor Network for Secure Electric Energy Infrastructure. *IEEE Transactions on Power Delivery 22*, 2 (Apr. 2007), 1021–1028.

[123] Leonov, V., Fiorini, P., Sedky, S., Torfs, T., and Van Hoof, C. Thermoelectric MEMS generators as a power supply for a body area network. In *Proceedings of*

the 13th International Conference on Solid-State Sensors, Actuators and Microsystems, 2005, Transducers '05 (June 2005), 291–294.

[124] Leonov, V., Torfs, T., Fiorini, P., and Van Hoof, C. Thermoelectric Converters of Human Warmth for Self-Powered Wireless Sensor Nodes. *IEEE Sensors Journal 7*, 5 (May 2007), 650–657.

[125] Levis, P., Madden, J., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D. TinyOS: An Operating System for Sensor Networks. In *Ambient Intelligence*, W. Weber, J. Rabaey, and E. H. Aarts, Eds. Springer-Verlag, Berlin, Heidelberg, 2005, ch. 9, 115–148.

[126] Li, P., Wen, Y., Liu, P., Li, X., and Jia, C. A magnetoelectric energy harvester and management circuit for wireless sensor network. *Sensors and Actuators A: Physical 157*, 1 (2010), 100–106.

[127] Lin, M., Wu, Y., and Wassell, I. Wireless sensor network: Water distribution monitoring system. In *Radio and Wireless Symposium, 2008 IEEE* (jan. 2008), 775 –778.

[128] Lin, Y., and Wolf, J. Combined ECC/RLL Codes. *IEEE Transactions on Magnetics 24*, 6 (Nov. 1988), 2527–2529.

[129] Liu, H., Meng, Z., and Cui, S. A wireless sensor network prototype for environmental monitoring in greenhouses. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on* (Sept. 2007), 2344–2347.

[130] Liu, V., Parks, A., Talla, V., Gollakota, S., Wetherall, D., and Smith, J. R. Ambient Backscatter: Wireless Communication out of Thin Air. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, ACM (New York, NY, USA, 2013), 39–50.

[131] Lo, B., Thiemjarus, S., King, R., and Yang, G. Body Sensor Network — A Wireless Sensor Platform for Pervasive Healthcare Monitoring. In *The 3rd International Conference on Pervasive Computing*, Pervasive '03 (2005).

[132] Lombriser, C., Bharatula, N. B., Roggen, D., and Tröster, G. On-body activity recognition in a dynamic sensor network. In *Proceedings of the ICST 2nd International Conference on Body Area Networks*, BodyNets '07, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (Brussels, Belgium, 2007), 1–17.

[133] Lorincz, K., Chen, B., Challen, G., Chowdhury, A., Patel, S., Bonato, P., and Welsh, M. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, ACM (New York, NY, USA, 2009), 183–196.

[134] Lorincz, K., Malan, D., Fulford-Jones, T., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., and Moulton, S. Sensor Networks for Emergency Response: Challenges and Opportunities. *IEEE Pervasive Computing 3*, 4 (Oct. 2004), 16–23.

[135] LPB Communications. *Technical Note #1 – Carrier Current Broadcasting Theory*.

[136] Lundquist, J. D., Cayan, D. R., and Dettinger, M. D. Meteorology and Hydrology in Yosemite National Park: A Sensor Network Application. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks*, IPSN 2003, Springer-Verlag (Berlin, Heidelberg, 2003), 518–528.

[137] Malan, D., Fulford-Jones, T., Welch, M., and Moulton, S. CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[138] Matese, A., Gennaro, S. D., Zaldei, A., Genesio, L., and Vaccari, F. A wireless sensor network for precision viticulture: The NAV system. *Computers and Electronics in Agriculture 69*, 1 (2009), 51–58.

[139] Mathúna, C. O., O'Donnell, T., Martinez-Catala, R. V., Rohan, J., and O'Flynn, B. Energy scavenging for long-term deployable wireless sensor networks. *Talanta 75*, 3 (2008), 613–623.

[140] McCulloch, J., McCarthy, P., Guru, S. M., Peng, W., Hugo, D., and Terhorst, A. Wireless sensor network deployment for water use efficiency in irrigation. In *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, REALWSN '08, ACM (New York, NY, USA, 2008), 46–50.

[141] McKelvin, Jr., M. L., Williams, M. L., and Berry, N. M. Integrated radio frequency identification and wireless sensor network architecture for automated inventory management and tracking applications. In *Proceedings of the 2005 Conference on Diversity in Computing*, TAPIA '05, ACM (New York, NY, USA, 2005), 44–47.

[142] Min, R., Bhardwaj, M., Cho, S.-H., Ickes, N., Shih, E., Sinha, A., Wang, A., and Chandrakasan, A. Energy-Centric Enabling Technologies for Wireless Sensor Networks. *IEEE Wireless Communications 9*, 4 (2002), 28–39.

[143] Min, R., and Chandrakasan, A. A Framework for Energy-Scalable Communication in High-Density Wireless Networks. In *2002 International Symposium on Low Power Electronics and Design (ISLPED '02)*, ACM (New York, NY, USA, 2002), 36–41.

[144] Mlynek, P., Koutny, M., and Misurec, J. Power line modelling for creating PLC communication system. *International Journal of Communications 4*, 1 (2010).

[145] Mohan, S., del Mar Hershenson, M., Boyd, S., and Lee, T. Simple accurate expressions for planar spiral inductances. *IEEE Journal of Solid-State Circuits 34*, 10 (Oct 1999), 1419–1424.

[146] Moon, T. *Error correction coding : mathematical methods and algorithms*. Wiley-Interscience, Hoboken, NJ, 2005.

[147] Musăloiu-E, R., Terzis, A., Szlavecz, K., Szalay, A., Cogan, J., and Gray, J. Life under your feet: A wireless soil ecology sensor network. In *Proceedings of the 3rd Workshop on Embedded Networked Sensors*, EmNets 2006 (2006).

[148] Nachman, L., Kling, R., Adler, R., Huang, J., and Hummel, V. The Intel Mote Platform: A Bluetooth-Based Sensor Network for Industrial Monitoring. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, IEEE (2005), 437–442.

[149] Nam, Y. H., Halm, Z., Chee, Y. J., and Park, K. S. Development of remote diagnosis system integrating digital telemetry for medicine. In *20th Annual International Conference on the IEEE Engineering in Medicine and Biology Society, 1998* (1998), 1170–1173.

[150] Nishimoto, H., Kawahara, Y., and Asami, T. Prototype implementation of ambient RF energy harvesting wireless sensor networks. In *IEEE Sensors 2010* (Nov 2010), 1282–1287.

[151] Noury, N., Herve, T., Rialle, V., Virone, G., Mercier, E., Morey, G., Moro, A., and Porcheron, T. Monitoring behavior in home using a smart fall sensor and position sensors. In *1st Annual International Conference on Microtechonologies in Medicine and Biology, 2000* (2000), 607–610.

[152] O'Flyrm, B., Martinez, R., Cleary, J., Slater, C., Regan, F., Diamond, D., and Murphy, H. SmartCoast: A Wireless Sensor Network for Water Quality Monitoring. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks, 2007*, LCN 2007 (Oct. 2007), 815–816.

[153] Ogawa, M., Tamura, T., and Togawa, T. Fully automated biosignal acquisition in daily routine through 1 month. In *20th Annual International Conference on the IEEE Engineering in Medicine and Biology Society, 1998* (1998), 1947–1950.

[154] Osterlind, F., Pramsten, E., Roberthson, D., Eriksson, J., Finne, N., and Voigt, T. Integrating building automation systems and wireless sensor networks. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, 2007*, ETFA '07 (Sept. 2007), 1376–1379.

[155] Otto, C., Milenkovic, A., Sanders, C., and Jovanov, E. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia 1*, 4 (2006), 307–326.

[156] Panchard, J., Rao, S., Prabhakar, T., Hubaux, J., and Jamadagni, H. COMMONSense Net: A wireless sensor network for resource-poor agriculture in the semiarid areas of developing countries. *Information Technologies and International Development 4*, 1 (2007), 51–67.

[157] Paradiso, J., and Starner, T. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing 4*, 1 (January–March 2005), 18–27.

[158] Paradiso, J. A. Systems for human-powered mobile computing. In *Proceedings of the 43rd Annual Design Automation Conference*, DAC '06, ACM (New York, NY, USA, 2006), 645–650.

[159] Paradiso, J. A., and Feldmeier, M. A Compact, Wireless, Self-Powered Pushbutton Controller. In *Proceedings of the International Conference on Ubiquitous Computing*, Ubicomp 2001, Springer-Verlag (Berlin, Heidelberg, 2001), 299–304.

[160] Park, C., Lahiri, K., and Raghunathan, A. Battery discharge characteristics of wireless sensor nodes: an experimental analysis. In *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks 2005 (SECON 2005)* (Sept 2005), 430–440.

[161] Park, C., Liu, J., and Chou, P. H. Eco: an Ultra-Compact Low-Power Wireless Sensor Node for Real-Time Motion Monitoring. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)* (2005), 398–403.

[162] Parks, A., Sample, A., Zhao, Y., and Smith, J. A wireless sensing platform utilizing ambient RF energy. In *2013 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS)* (Jan 2013), 154–156.

[163] Patel, S., Reynolds, M., and Abowd, G. Detecting Human Movement by Differential Air Pressure Sensing in HVAC System Ductwork: An Exploration in Infrastructure Mediated Sensing. In *Proceedings of the 6th International Conference on Pervasive Computing*, Pervasive '08, Springer-Verlag (Berlin, Heidelberg, 2008), 1–18.

[164] Patel, S. N. *Infrastructure Mediated Sensing*. PhD thesis, Georgia Institute of Technology, 2008.

[165] Patel, S. N., Robertson, T., Kientz, J. A., Reynolds, M. S., and Abowd, G. D. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, Ubicomp '07, Springer-Verlag (Berlin, Heidelberg, 2007), 271–288.

[166] Patel, S. N., Stuntebeck, E. P., and Robertson, T. PL-Tags: Detecting Batteryless Tags through the Power Lines in a Building. In *Proceedings of the 7th International Conference on Pervasive Computing*, Pervasive '09, Springer-Verlag (Berlin, Heidelberg, 2009), 256–273.

[167] Patel, S. N., Truong, K. N., and Abowd, G. D. PowerLine Positioning: A Practical Sub-Room-Level Indoor Location System for Domestic Use. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, Ubicomp '06, Springer-Verlag (Berlin, Heidelberg, 2006), 441–458.

[168] Petriu, E. M., Georganas, N. D., Petriu, D. C., Makrakis, D., and Groza, Voicu, Z. Sensor-Based Information Applicances. *IEEE Instrumentation & Measurement Magazine 3*, 4 (2000), 31–35.

[169] Philipose, M., Smith, J., Jiang, B., Mamishev, A., Roy, S., and Sundara-Rajan, K. Battery-free wireless identification and sensing. *IEEE Pervasive Computing 4*, 1 (Jan 2005), 37–45.

[170] Polastre, J., Szewczyk, R., and Culler, D. Telos: Enabling Ultra-Low Power Wireless Research. In *Fourth International Symposium on Information Processing in Sensor Networks, 2005 (IPSN '05)* (2005), 364–369.

[171] Porret, A.-S., Melly, T., Enz, C. C., and Vittoz, E. A. A Low-Power Low-Voltage Transceiver Architecture Suitable for Wireless Distributed Sensors Network. In *IEEE International Symposium on Circuits and Systems (ISCAS 2000)* (2000), 56–59.

[172] Pottie, G., and Kaiser, W. Wireless Integrated Network Sensors. *Communications of the ACM 43*, 5 (2000), 51–58.

[173] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. The Cricket location-support system. In *6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, ACM (New York, NY, USA, 2000), 32–43.

[174] Rabaey, J., Ammer, J., da Silva, J.L., J., and Patel, D. PicoRadio : Ad-hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes. In *IEEE Computer Society Workshop on VLSI 2000* (2000), 9–12.

[175] Rabaey, J., Ammer, J., Otis, B., Burghardt, F., Chee, Y. H., Pletcher, N., Sheets, M., and Qin, H. Ultra-low-power design. *IEEE Circuits and Devices Magazine 22*, 4 (2006), 23–29.

[176] Rabaey, J., Ammer, M., da Silva, Jr., J., and Roundy, S. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *Computer 33*, 7 (2000), 42–48.

[177] Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., and Srivastava, M. Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, IEEE Press (Piscataway, NJ, USA, 2005).

[178] Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. B. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine 19*, 2 (2002), 40–50.

[179] Rahimi, M., Shah, H., Sukhatme, G., Heideman, J., and Estrin, D. Studying the feasibility of energy harvesting in a mobile sensor network. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2003*, ICRA '03 (Sept. 2003), 19–24.

[180] Rai, S., Holleman, J., Pandey, J., Zhang, F., and Otis, B. A 500 μW Neural Tag with 2 μV_{rms} AFE and Frequency-Multiplying MICS/ISM FSK Transmitter. In *Proceedings of the IEEE International Solid-State Circuits Conference* (2009).

[181] Riem-Vis, R. Cold Chain Management using an Ultra Low Power Wireless Sensor Network. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[182] Roundy, S., Steingart, D., Frechette, L., Wright, P., and Rabaey, J. Power Sources for Wireless Sensor Networks. In *Proceedings in the First European Workshop on Wireless Sensor Networks*, EWSN 2004, Springer-Verlag (Berlin, Heidelberg, 2004), 1–17.

[183] Roundy, S., Wright, P. K., and Rabaey, J. M. *Energy Scavenging for Wireless Sensor Networks: With Special Focus on Vibrations*. Springer-Verlag, Berlin, Heidelberg, 2004.

[184] Rowe, A., Gupta, V., and Rajkumar, R. R. Low-Power Clock Synchronization using Electromagnetic Energy Radiating from AC Power Lines. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, ACM (New York, NY, USA, 2009), 211–224.

[185] Roy, S., Jandhyala, V., Smith, J., Wetherall, D., Otis, B., Chakraborty, R., Buettner, M., Yeager, D., Ko, Y.-C., and Sample, A. RFID: From Supply Chains to Sensor Nets. *Proceedings of the IEEE 98*, 9 (Sept 2010), 1583–1592.

[186] Ryckaert, J., De Doncker, P., Meys, R., de Le Hoye, A., and Donnay, S. Channel model for wireless communication around human body. *Electronics Letters 40*, 9 (April 2004), 543–544.

[187] Sample, A., Yeager, D., Powledge, P., Mamishev, A., and Smith, J. Design of an RFID-Based Battery-Free Programmable Sensing Platform. *IEEE Transactions on Instrumentation and Measurement 57*, 11 (Nov 2008), 2608–2615.

[188] Sample, A., Yeager, D., Powledge, P., and Smith, J. Design of a Passively-Powered, Programmable Sensing Platform for UHF RFID Systems. In *Proceedings of the IEEE International Conference on RFID, 2007.* (Mar. 2007), 149–156.

[189] Savvides, A., and Srivastava, M. B. A Distributed Computation Platform for Wireless Embedded Sensing. In *IEEE International Conference on Computer Design: VLSI in Computers and Processors 2002 (ICCD '02)*, IEEE Comput. Soc (2002), 220–225.

[190] Sazonov, E., Janoyan, K., and Jha, R. Wireless intelligent sensor network for autonomous structural health monitoring. *Smart Structures/NDE 2004* (2004).

[191] Schiller, J., Liers, A., Ritter, H., Winter, R., and Voigt, T. ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing. In *38th Hawaii International Conference on System Sciences (HICSS '05)* (2005).

[192] Schneider, A. ELECTRONICA 2004: "Enabled by EnOcean" - User programmable transceiver module enables highly innovative sensor networks. `http://www.wellington-partners.com/wp/downloads/pfn/enocean/2004/041105_eno.pdf`, Nov. 2004.

[193] Schor, L., Sommer, P., and Wattenhofer, R. Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, ACM (New York, NY, USA, 2009), 31–36.

[194] Schwiebert, L., Gupta, S. K., and Weinmann, J. Research Challenges in Wireless Networks of Biomedical Sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, ACM (New York, NY, USA, 2001), 151–165.

[195] Seah, W., Eu, Z. A., and Tan, H.-P. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges. In *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009*, Wireless VITAE 2009 (May 2009), 1–5.

[196] Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Kang, W., Stankovic, J., Young, D., and Porter, J. LUSTER: wireless sensor network for environmental research. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, ACM (New York, NY, USA, 2007), 103–116.

[197] Shan, Q., Liu, Y., Prosser, G., and Brown, D. Wireless intelligent sensor networks for refrigerated vehicle. In *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004* (2004), 525–528.

[198] Shih, E., Cho, S.-H., Ickes, N., Min, R., Sinha, A., Wang, A., and Chandrakasan, A. Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, ACM Press (New York, NY, USA, 2001), 272–287.

[199] Sibbald, B. Use computerized systems to cut adverse drug events: report. *Canadian Medical Association Journal 164*, 13 (2001).

[200] Simjee, F., and Chou, P. Efficient Charging of Supercapacitors for Extended Lifetime of Wireless Sensor Nodes. *IEEE Transactions on Power Electronics 23*, 3 (May 2008), 1526–1536.

[201] Simon, G., Maróti, M., Lédeczi, A., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., and Frampton, K. Sensor network-based countersniper system. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, ACM (New York, NY, USA, 2004), 1–12.

[202] Sivaharan, T., Blair, G., Friday, A., Wu, M., Duran-Limon, H., Okanda, P., and Sørensen, C.-F. Cooperating Sentient Vehicles for Next Generation Automobiles. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[203] Smith, J. R., Fishkin, K. P., Jiang, B., Mamishev, A., Philipose, M., Rea, A., Roy, S., and Sundara-Rajan, K. RFID-Based Techniques for Human Activity Recognition. *Communications of the ACM 48*, 9 (Sept. 2005).

[204] Smith, J. R., Sample, A. P., Powledge, P. S., Roy, S., and Mamishev, A. A Wirelessly-Powered Platform for Sensing and Computation. In *Proceedings of the 8th ACM International Conference on Ubiquitous Computing*, UbiComp '06, Springer-Verlag (Berlin, Heidelberg, 2006), 495–506.

[205] Snoonian, D. Smart Buildings. *IEEE Spectrum 40*, 8 (2003), 18–23.

[206] Sohraby, K., Minoli, D., and Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, Hoboken, NJ, 2007.

[207] Srivastava, M., Muntz, R., and Potkonjak, M. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, ACM (New York, NY, USA, 2001), 132–138.

[208] Stankovic, J., Cao, Q., Doan, T., Fang, L., He, Z., Kiran, R., Lin, S., Son, S., Stoleru, R., and Wood, A. Wireless sensor networks for in-home healthcare: Potential and challenges. In *High Confidence Medical Device Software and Systems (HCMDSS) Workshop* (2005), 2–3.

[209] Steere, D. C., Baptista, A., McNamee, D., Pu, C., and Walpole, J. Research challenges in environmental observation and forecasting systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, ACM (New York, NY, USA, 2000), 292–299.

[210] Stuntebeck, E. P. *An Analysis of the Domestic Power Line Infrastructure to Support Indoor Real-Time Localization*. PhD thesis, Georgia Institute of Technology, 2010.

[211] Stuntebeck, E. P., Patel, S. N., Robertson, T., Reynolds, M. S., and Abowd, G. D. Wideband powerline positioning for indoor localization. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, ACM (New York, NY, USA, 2008), 94–103.

[212] Sun Labs. *Sun™SPOT Main Board Technical Datasheet, Rev 8.0*, 2.0.1 ed. Oracle America, Inc, 16 Network Circle, Menlo Park, CA 94025, 10 2010.

[213] Sun Labs. *Sun™SPOT Programmer's Manual, Rev 8.0*, 2.0 ed., 11 2010.

[214] Sung, M., and Pentland, A. S. LiveNet: Health and Lifestyle Networking Through Distributed Mobile Devices. In *Proceedings of the Workshop on Applications of Mobile Embedded Systems at Mobisys 2004*, WAMES 2004 (2004).

[215] Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., and Estrin, D. Habitat monitoring with sensor networks. *Communications of the ACM 47*, 6 (June 2004), 34–40.

[216] Tabar, A. M., Keshavarz, A., and Aghajan, H. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, ACM (New York, NY, USA, 2006), 145–154.

[217] Talla, V., Buettner, M., Wetherall, D., and Smith, J. Hybrid analog-digital backscatter platform for high data rate, battery-free sensing. In *2013 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)* (Jan 2013), 1–3.

[218] Talla, V., and Smith, J. Hybrid analog-digital backscatter: A new approach for battery-free sensing. In *2013 IEEE International Conference on RFID (RFID)* (April 2013), 74–81.

[219] Tapia, E. M., Intille, S. S., and Larson, K. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Proceedings of the Seconds International Conference on Pervasive Computing*, Pervasive 2004, Springer-Verlag (Berlin, Heidelberg, 2004), 158–175.

[220] Tapia, E. M., Intille, S. S., Lopez, L., and Larson, K. The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection. In *Proceedings of the 4th International Conference on Pervasive Computing*, Pervasive 2006, Springer-Verlag (Berlin, Heidelberg, 2006), 117–134.

[221] Thomas, S., Harrison, R., Leonardo, A., and Reynolds, M. A battery-free multi-channel digital neural/EMG telemetry system for flying insects. In *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (Nov 2011), 229–232.

[222] Thomas, S., and Reynolds, M. A 96 Mbit/sec, 15.5 pJ/bit 16-QAM modulator for UHF backscatter communication. In *2012 IEEE International Conference on RFID (RFID)* (April 2012), 185–190.

[223] Thompson, M. Line-Cord Antenna, 1947. United States Patent #2581983.

[224] Togami, T., Yamamoto, K., Hashimoto, A., Watanabe, N., Takata, K., Nagai, H., and Kameoka, T. A wireless sensor network in a vineyard for smart viticultural management. In *Proceedings of the SICE Annual Conference*, SICE 2011 (Sept. 2011), 2450–2454.

[225] Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., and Hong, W. A Macroscope in the Redwoods. In *Proceedings of the 3rd International Conference of Embedded Networked Sensor Systems (SenSys '05)*, ACM (New York, NY, USA, 2005), 51–63.

[226] Tsujita, W., Yoshino, A., Ishida, H., and Moriizumi, T. Gas sensor network for air-pollution monitoring. *Sensors and Actuators B: Chemical 110*, 2 (2005), 304–311.

[227] Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., and Corke, P. Data Collection, Storage, and Retrieval with an Underwater Sensor Network. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, ACM (New York, NY, USA, 2005), 154–165.

[228] Virone, G., Wood, A., Selavo, L., Cao, Q., Fang, L., Doan, T., He, Z., and Stankovic, J. An advanced wireless sensor network for health monitoring. In *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)* (2006), 2–4.

[229] Virone, G., Wood, A., Selavo, L., Cao, Q., Fang, L., Doan, T., He, Z., Stoleru, R., Lin, S., and Stankovic, J. An Assisted Living Oriented Information System Based on a Residential Wireless Sensor Network. In *Proceedings of the 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, D2H2 2006 (Apr. 2006), 95–100.

[230] Volgyesi, P., Balogh, G., Nadas, A., Nash, C. B., and Ledeczi, A. Shooter localization and weapon classification with soldier-wearable networked sensors. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, ACM (New York, NY, USA, 2007), 113–126.

[231] Vullers, R., Schaijk, R., Visser, H., Penders, J., and Hoof, C. Energy Harvesting for Autonomous Wireless Sensor Networks. *IEEE Solid-State Circuits Magazine 2*, 2 (Spring 2010), 29–38.

[232] Wang, H., Elson, J., Girod, L., Estrin, D., and Yao, K. Target classification and localization in habitat monitoring. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '03 (Apr. 2003).

[233] Wang, L., Yang, Y., Noh, D. K., Le, H., Liu, J., Abdelzaher, T., and Ward, M. AdaptSens: An Adaptive Data Collection and Storage Service for Solar-Powered Sensor Networks. In *Proceedings of the 30th IEEE Real-Time Systems Symposium, 2009*, RTSS 2009 (Dec. 2009), 303–312.

[234] Wang, N., Zhang, N., and Wang, M. Wireless sensors in agriculture and food industry—recent development and future perspective. *Computers and Electronics in Agriculture 50*, 1 (2006), 1–14.

[235] Wang, W., Wang, N., Jafer, E., Hayes, M., O'Flynn, B., and O'Mathuna, C. Autonomous wireless sensor network based building energy and environment monitoring system design. In *Proceedings of the 2010 International Conference on Environmental Science and Information Application Technology*, ESIAT 2010 (July 2010), 367–372.

[236] Want, R. Enabling ubiquitous sensing with RFID. *Computer 37*, 4 (April 2004), 84–86.

[237] Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D., and Bishop-Hurley, G. Transforming Agriculture through Pervasive Wireless Sensor Networks. *IEEE Pervasive Computing 6*, 2 (April–June 2007), 50–57.

[238] Warneke, B., Last, M., Liebowitz, B., and Pister, K. S. Smart Dust: Communicating with a Cubic- Millimeter Computer. *Computer 34*, 1 (2001), 44–51.

[239] Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., and Welsh, M. Monitoring volcanic eruptions with a wireless sensor network. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005* (2005), 108–120.

[240] Werner-Allen, G., Lorincz, K., Welsh, M., Marcillo, O., Johnson, J., Ruiz, M., and Lees, J. Deploying a Wireless Sensor Network on an Active Volcano. *IEEE Internet Computing 10*, 2 (2006), 18–25.

[241] Wheeler, A. Commercial Applications of Wireless Sensor Networks Using ZigBee. *IEEE Communications Magazine 45*, 4 (Apr. 2007), 70–77.

[242] EnOcean – Dolphin Platform. `http://www.madebydelta.com/imported/images/DELTA_Web/documents/TC/Wireless/Articles/EnOcean_Dolphin_Platform.pdf`, 2009.

[243] Wu, W.-J., Chen, Y.-F., Chen, Y.-Y., Wang, C.-S., and Chen, Y.-H. Smart Wireless Sensor Network Powered by Random Ambient Vibrations. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2006*, SMC '06 (Oct. 2006), 2701–2708.

[244] Xiong, F. *Digital Modulation Techniques*. Artech House, Boston, 2000.

[245] Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R., and Estrin, D. A Wireless Sensor Network for Structural Monitoring. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, ACM (New York, NY, USA, 2004), 13–24.

[246] Xue, X., Gonzalez-Argueta, L., and Sundararajan, V. Energy Scavenging for Wireless Sensor Networks. In *2007 ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications*, IDETC 2007 (2007).

[247] Yang, X., Ong, K. G., Dreschel, W. R., Zeng, K., Mungle, C. S., and Grimes, C. A. Design of a Wireless Sensor Network for Long-term, In-Situ Monitoring of an Aqueous Environment. *Sensors 2*, 11 (2002), 455–472.

[248] Yap, K.-K., Srinivasan, V., and Motani, M. MAX: Human-Centric Search of the Physical World. In *Proceedings of the 3rd International Conference of Embedded Networked Sensor Systems (SenSys '05)*, ACM (New York, NY, USA, 2005), 166–179.

[249] Yeager, D., Powledge, P., Prasad, R., Wetherall, D., and Smith, J. Wirelessly-Charged UHF Tags for Sensor Data Collection. In *2008 IEEE International Conference on RFID* (April 2008), 320–327.

[250] Yeager, D. J., Sample, A. P., Smith, J. R., and Smith, J. R. WISP: A passively powered UHF RFID tag with sensing and computation. *RFID Handbook: Applications, Technology, Security, and Privacy* (2008), 261–278.

[251] Yick, J., Mukherjee, B., and Ghosal, D. Wireless sensor network survey. *Computer Networks 52*, 12 (Aug. 2008), 2292–2330.

[252] Zensys. *ZM3102N Z-Wave Module Datasheet*, 6 ed., Oct. 2007.

[253] Zhang, P., Sadler, C. M., Lyon, S. A., and Martonosi, M. Hardware Design Experiences in ZebraNet. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, ACM (New York, NY, USA, 2004), 227–238.

[254] Zimmerman, T. Personal Area Networks: Near-field intrabody communication. *IBM Systems Journal 35*, 3.4 (1996), 609–617.